

МАКСИМ КУЗНЕЦОВ, ИГОРЬ СИМДЯНОВ

программирование: ступени успешной карьеры

ЧЕМУ И КАК УЧИТЬСЯ

СТРАТЕГИЯ ПОИСКА РАБОТЫ

КАК «РАСКРУТИТЬ» СЕБЯ

ХОРОШИЙ ТОН В ПРОГРАММИРОВАНИИ
И ИНТЕРФЕЙСЕ

ВАЖНОСТЬ ПРОЕКТИРОВАНИЯ

ПРИМЕРЫ ДОГОВОРОВ

АВТОРСКОЕ ПРАВО В ИТ

О.К.

**Максим Кузнецов
Игорь Симдянов**

программирование: ступени успешной карьеры

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.06
ББК 32.973.26
К89

Кузнецов М. В., Симдянов И. В.

К89 Программирование: ступени успешной карьеры. —
СПб.: БХВ-Петербург, 2006. — 320 с.: ил.

ISBN 5-94157-802-4

Написанная опытными профессионалами и известными авторами книга посвящена различным аспектам успешного создания карьеры в программировании и области IT-технологий. Рассмотрены как технические аспекты, такие как хороший тон в программировании, правила дизайна интерфейсов, приемы проектирования, так и вопросы, напрямую с программированием, казалось бы, не связанные: методология обучения программированию, стратегия поиска работы, специфика работы в команде и т. д. Большое внимание уделено практике применения закона об авторском праве и смежных правах в IT-области, а также грамотному подходу к составлению договоров.

Для программистов и IT-специалистов

УДК 681.3.06
ББК 32.973.26

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Игорь Цырульников</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 30.03.06.

Формат 70×100^{1/16}. Печать офсетная. Усл. печ. л. 25,8.

Тираж 2000 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04
от 11.11.2004 г. выдано Федеральной службой по надзору
в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"

199034, Санкт-Петербург, 9 линия, 12

ISBN 5-94157-802-4

© Кузнецов М. В., Симдянов И. В., 2006
© Оформление, издательство "БХВ-Петербург", 2006

Оглавление

Введение	1
Для кого и о чем эта книга.....	1
Описание глав	1
Благодарности	3
Глава 1. Программирование: где, чему и как учиться	5
Вопросы и ответы	7
Глава 2. Как найти работу	19
Базовая техника устройства на работу.....	19
Настрой на поиск работы	22
Цель.....	27
Поиск информации	30
Сбор информации.....	32
Техника устройства на работу в нужную организацию.....	36
Составление резюме	38
Собеседование	43
Что делать, когда увольняют	46
Глава 3. Работа в команде	49
Введение в социальную психологию	49
Какие бывают группы	53
Учебно-карьеристские группы	54
Культурно-развлекательные группы	54
Алкогольно-сексуальная группа.....	56
О референтных группах.....	57
Групповые процессы	58
Об антилидерстве	59
Как найти свое место в группе	62
Глава 4. Взаимодействие с заказчиком	65
Встречают по одежке.....	66
О внешнем имидже.....	66
Встреча начинается задолго до встречи	68
Не думайте плохо о заказчике.....	68
Заказчик дурак?.....	70

Почему они такие?	70
Рожденный ползать — уйди с взлетной полосы!	71
Невербальное общение	71
Что в имени тебе моем.....	73
Основной закон психологии общения	75
В любом из нас спит гений. И с каждым днем все крепче	78
Личная встреча — лишний шаг к успеху	78
Не возражайте в лоб.....	78
ЯЗВа	79
АнтиЯЗВа.....	79
Закон об объеме оперативной памяти.....	80
Закон края (закон Эббингауза)	80
Закон контрастов	80
Не уходите от скользких вопросов	81
Несколько правил ведения дискуссии	82
Никогда не перебивайте собеседника	82
Не рассуждайте о том, чего сами не знаете	83
Не спорьте с очевидным	83
Да поможет нам НЛП	84
Мы снова говорим на разных языках.....	87
Подстройка и ведение	88
Влияние установок	90
Убеждение с игрой на некоторых слабостях	93
Неуверенность в себе.....	93
Медлительность.....	93
Тщеславность.....	93
Эффект ореола или эффект обобщения.....	94
Эффект близости.....	96
Важность первого впечатления	96
Точность — вежливость королей	96
Давайте делать паузы в словах	97
Вместо резюме или минус эмоции	98

Глава 5. Проектирование 101

Требования	101
Архитектура проекта.....	104
Спецификации	106
Планирование времени	107
Сложность.....	109
Масштабирование	112
Осознанное и случайное проектирование	115
Не давать пользователю почувствовать себя дураком	115
Осторожней с функциональностью!	116

Глава 6. Если вы — руководитель	117
Две стратегии руководства	117
Способы управления	120
Чем отличается руководство компанией от руководства командой	122
Но о тактике тоже нельзя забывать	123
Не складывайте все яйца в одну корзину	123
Главный психолог коллектива	127
Люди живут для себя	127
Престижность	127
Как превратить группу в коллектив	128
Поручайте своим сотрудникам только те дела, которые они могут выполнить	130
О критике или метод кнута и пряника	131
 Глава 7. Стратегический маркетинг — программисту, или как найти свою нишу на рынке software.....	 133
MySQL.....	136
Microsoft	136
Sun	138
Adobe	138
WinAmp	139
Opera	139
Yandex.....	140
Электронные платежные системы	140
Вместо заключения.....	140
 Глава 8. Проектирование пользовательского интерфейса	 143
Компьютер — это средство, а не цель.....	144
Виды интерфейсов.....	146
Командная строка	149
Графический интерфейс пользователя	155
Обратная связь	162
Эмуляция реальных предметов	165
Многословность программ	167
Психологический аспект.....	171
Распространение программы	173
Хороший разработчик — хороший пользователь.....	175
 Глава 9. Правила хорошего тона в программировании	 179
Объявления переменных и функций	179
Имена переменных и функций	182
Расстановка фигурных скобок и отступы	188
Пробелы вокруг символов	197
Комментарии.....	198

Объектно-ориентированное программирование	201
Когда используется объектно-ориентированное программирование	202
Интерфейс класса	207
Глава 10. Вместо заключения.....	213
Управление временем.....	213
Мечтать не вредно?.....	215
О добре, которое нередко бывает с кулаками	219
Скажи мне кто твой друг, и я скажу, кем ты станешь	221
Упражнение: куда уходит время?	224
Упражнение на развитие временной интуиции.....	225
Несколько советов по организации времени	226
Осторожно — хронофаги!	227
О пользе разумного эгоизма.....	229
Самое главное условие достижения успеха	231
Вдох глубокий, руки шире!.....	233
Немного о виктимологии.....	237
ПРИЛОЖЕНИЯ	241
Приложение 1. Договорные отношения.....	243
Договор на создание сайта.....	244
Техническое задание.....	252
Вопросы и ответы	252
Приложение 2. Авторское право	261
Закон "Об авторском праве и смежных правах".....	262
Содержание закона	262
Тексты статей	265
Вопросы и ответы	310

Введение

Для кого и о чем эта книга

Эта книга о том, что необходимо предпринять, чтобы сделать успешную карьеру в области информационных технологий, и отличается от предыдущих книг авторов тем, что она — не о кодировании, а о тех вопросах, на которые многие программисты, и, особенно начинающие, не обращают должного внимания: как устроиться на работу, как вести переговоры с клиентами, как грамотно работать в команде и т. д. Статистика показывает, что большинство программистских карьер рухнуло не из-за того, что программист не умел писать код (наоборот, многие делали это блестяще), а по той причине, что авторы программ не уделяли внимания "второстепенным" вопросам, считая их не относящейся к делу "внешней атрибутикой": умению вести переговоры с клиентом, рекламе своего программного продукта, изучению положения на рынке программного обеспечения и т. д. Между тем, при нынешней большой конкуренции на рынке программных продуктов не обращать внимания на подобные "мелочи" — значит, разрушить свою программистскую карьеру в самом начале. Тому, как не допустить подобного исхода, и посвящена эта книга. В этой книге нет никаких теоретических рассуждений — все советы и рекомендации сугубо практические, основанные на опыте авторов, регулярно проводящих очные и заочные консультации по рассматриваемым в книге вопросам.

Несмотря на то, что в книге много разделов, к программированию как таковому не относящихся, все-таки эта книга для программистов. И та часть книги, в которой в той или иной мере сделан упор на психологию, тоже для них. Это как с английским языком: язык-то один, а специфика преподавания для тех или иных специальностей разная. И то, что касается психологии, авторы попытались изложить языком, понятным именно программистам.

Описание глав

По сути, эта книга — анализ причин, почему те, кто потенциально мог бы стать хорошими программистами или IT-специалистами, ими не стали. И главы, из которых состоит эта книга, появлялись по принципу именно этого анализа. И именно по этому принципу книга виртуально разделена на две части "психологическую" и "техническую". Психологические главы —

это все, что касается взаимодействия с внешним миром: переговоров с заказчиками, работы в команде или руководства оной, продвижения собственного продукта и т. д. Техническая часть — все, что касается работы над программным продуктом. Интерфейс, проектирование, правила хорошего тона программирования. Причины, по которым некоторые программисты терпят неудачу, могут лежать как в технической сфере, так и во внешней. У одного вроде бы все хорошо, но продукт не идет, потому что интерфейс откровенно недружественный — появилась глава об интерфейсе. У другого все было хорошо до тех пор, пока не стал разрабатывать большой проект, в котором запутался, и завершить который не смог. Почему так бывает и как этого избежать — в главах о проектировании и хорошем тоне программирования. Теперь о внешней стороне. Многие успешные в техническом смысле программисты совершенно не могут продать свой продукт, потому что не владеют навыками общения с клиентом. Для них — глава о взаимодействии с заказчиком с многочисленными статьями-вставками из различных областей психологии, по сути представляющая собой небольшое пособие по ведению переговоров. В основном я приводил те приемы, которые сам использую в повседневной практике ведения переговоров. Кто-то, хорошо программируя, не сумел сработаться с командой, когда пришла пора. О том, как сделать процесс вхождения в новую команду менее болезненным — в главе о том, как работать в команде. А другой дорос до руководства командой и успешно ее развалил, в результате чего чуть не спился. О том, как не разваливать команды, а наоборот спланировать их — в главах "Работа в команде" и "Если вы — руководитель". Много вопросов у нас на форуме касается "институтской" проблемы — чему, где и как учиться программисту. Обо всем этом — в первой главе. В этой же главе мы говорим о том, что проблема, когда программировать очень хочется, а профильного образования нет — проблема надуманная. Нет такой проблемы. А сколько прекрасных программистов, что называется "от Бога", в силу тех или иных причин получив "не то" образование, загубили себе жизнь, послушав высокоумных дураков, говорящих о том, что без диплома они не программисты. Дорогой читатель! Если вы один из тех, кому так говорят, и вы уже почти в это поверили — поверьте теперь и в то, что это не так. Пока просто поверьте. А прочитав первую главу, увидите более чем убедительные реальные доказательства этого положения. Что диплом, это еще очень и очень мало. Хорошо, когда есть, ничего страшного, если нет. Не дипломами войны выигрываются, а умением. И если вы реально можете что-то хорошо делать (в нашем случае программировать), то ни один здравомыслящий руководитель ваш профильный диплом не спросит. А если очень хорошо умеете что-то делать, то, возможно, простят и отсутствие вообще какого бы то ни было диплома, а не только профильного.

Теперь о самой последней главе, которую можно было бы назвать "Фундаментальная", — от слова "фундамент", а не от нескромности автора. Просто

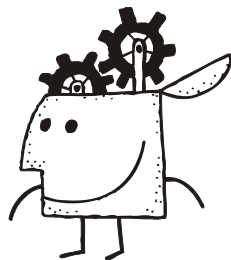
очень часто бывает, что говоришь с человеком, который пришел посоветоваться насчет проблемы, и видишь, что технически он еще, может, и грамотнее тебя будет, и все книжки по психологии на предмет того, как вести переговоры перечитал, но не идет дело, и все тут. Чудесный программный продукт, а продать никому не удастся. Первая мысль — неумение вести переговоры. И верно. Не умеет. Но о том, как их вести все прекрасно знает. А голос блеющий, руки потеющие... В общем, "все не так, ребята". А всему виной та самая оценочная зависимость, зависимость от чужого мнения, неверие в себя... Почти все через это прошли, но далеко не все успешно... О ней и о некоторых вариантах избавления — в "подглаве" прикладная виктимология. А у другого иная проблема. О которой он, правда, пока не догадывается. Сидит целыми днями и программирует, программирует без отдыха, без прогулок, без спорта. И вроде все хорошо, только как работник умственного труда он уже обречен. Потому что через пару-тройку лет такой жизни начнет производить умственный шлак вместо гениальных строк кода... Если вообще что-то будет в состоянии производить. Примеров — масса, исключений нет. ...Так появилась подглава о вреде нездоровья. А у третьего вообще ни на что не хватает времени. В принципе. А хочется так многого... Но ничего не успевает. О том, почему так, в разделе "Управление временем". Если у вас со временем, здоровьем и удачливостью все нормально, можете не читать эту главу. А вот если чувствуете, что что-то не так, задержитесь на ней подольше и постарайтесь решить вопросы, в ней освещаемые. С помощью этой книги, или других. Неважно. Важно то, что, не построив фундамент, постройку дома не начинают. И, к примеру, если вы не уважаете своих клиентов, а терпеть их не можете, то читать главу о том, как вести переговоры, бесполезно. Вы их все равно провалите, даже если будете безупречно соблюдать все рекомендации. И причина ваших неудач — гораздо глубже, чем неисполнение той или иной техники.

Все, что написано в этой книге, нами много раз апробировано и испытано. Это относится и к технической и к психологической стороне. О том, чего пока сами не испробовали и не знаем, мы не писали. Очень часто, особенно в главах, так или иначе касающихся психологии, повествование идет от первого лица. Поэтому, наверное, будет уместно сказать, что в основном "психологические" главы написаны М. Кузнецовым, а "технические" — И. Симдяновым.

Благодарности

Мы признательны сотрудникам издательства "БХВ-Петербург", благодаря которым эта книга увидела свет, а также всем читателям наших предыдущих книг.

Глава 1



Программирование: где, чему и как учиться

Для начала абстрагируемся от программирования и поговорим об образовании и обучении вообще. И в первой части этой главы развеем один очень распространенный миф, который не одному человеку помешал стать тем, кем он хотел бы быть. А во второй мы приведем наши ответы на вопросы читателей, которые они нам задавали по поводу обучения программированию (как в рамках вузовской программы, так и самостоятельного) и попутно развеем еще несколько мифов.

Итак, миф, который гласит, что, не имея диплома о высшем профильном профессиональном образовании, найти себе работу программиста (дизайнера, системного администратора и т. д.) и заниматься программированием невозможно.

Проще говоря, если вы не закончили какой-нибудь ВМК или мехмат, то о карьере программиста можете забыть. Это — не так. И я говорю об этом, имея достаточные на то основания. Да, такая теория существует, она очень живуча и не одному человеку жизнь испортила. Да, те, кто такое образование имеет, нередко делают все для того, чтобы человек "не их касты" не проник "в святая святых".

Примечание

Сейчас такого, конечно, меньше. Но учитывайте, что "высшеобразовательный снобизм" очень свойствен людям, получившим образование в 70-х и 80-х годах.

Теперь давайте эту теорию разрушим.

Примечание о методологии

Кстати, методология говорит, что любую теорию можно опровергнуть только двумя способами. Первый — это доказать ее внутреннюю противоречивость. (На этом способе основан известный метод доказательства теорем "от противного"). А второй — показать, что основные положения теории противоречат реальным фактам.

Здесь мы пойдем по второму пути и приведем несколько значимых примеров, которые эту теорию абсолютно опровергают. Кстати, это справедливо по отношению не только к программированию, но и ко многим другим отраслям деятельности, в которых, казалось бы, без профильного образования вообще никуда. И тем не менее...

К примеру, одним из основоположников гипнотерапии в современной психотерапии считается Эмиль Куэ, корифеем от психотерапии и медицины никогда не бывший, а бывший всего лишь очень скромным аптекарем.

Всемирно известный писатель-зоолог Д. Даррелл никогда не учился на зоолога. Более того, он вообще нигде не учился, и считал, что ему невероятно повезло в том, что он не получил высшего образования, тем более профильного. И во многих интервью говорил о том, что если бы он выучился на зоолога, то никогда бы не стал "всемирно известным писателем-натуралистом и основателем и директором собственного уникального зоопарка".

Всемирно известный ученый-физик академик (!) Я. Б. Зельдович тоже вообще не имел высшего образования. Он лишь два года проучился на заочном отделении физико-математического факультета Ленинградского университета, который не окончил. Но он очень много занимался самообразованием и добился того, что в 1934 г. был принят в аспирантуру ИХФ, в 1936 г. защитил кандидатскую диссертацию, а в 1939 г. диссертацию на степень доктора физико-математических наук. А потом еще и академиком стал.

А очень известный в радиотехнике метод "прыжков по частотам" придуман, между прочим, не титулованным ученым-радиотехником, а ...американской киноактрисой Х. Ламарр. Которая, кроме того, что в эпоху Второй мировой войны была известной киноактрисой, еще очень интересовалась радиоэлектроникой. Сугубо для себя, казалось бы. И вот, когда она по приглашению каких-то там чинов из ихнего министерства обороны присутствовала на учениях, на которых демонстрировались торпедные атаки, ей пришла в голову мысль, что для надежной защиты от помех систем радиоуправления торпедами нужно, чтобы источник и приемник сигнала синхронно меняли частоту. Вот именно этот метод и получил название "прыжки по частотам".

Примечание

Так часто бывает, когда значительный вклад в развитие какой-либо научной области делают люди, от науки очень далекие. И Х. Ламмар как киноактрису мало кто помнит, а вот как первооткрывателя метода "прыжков по частотам" знают многие. Кстати, эта идея лежит сейчас в основе помехозащищенности и секретности передачи информации в цифровых системах сотовой связи, таких как GSM, DECT и т. д.

А если вернуться к программированию, то лучше всего вспомнить ненавидимого многими Билла Гейтса, у которого, как всем известно, есть только

незаконченное профильное образование. И что-то он не сильно комплексует по поводу отсутствия оного.

А вот, к примеру, совершенно недавняя ситуация с моим участием. Пришли к нам устраиваться на работу два Web-дизайнера. Один не имеет никаких дипломов (вернее имеет — об окончании ПТУ), а у второго диплом о профильном высшем образовании (специальность "Компьютерная графика"). Но первый рисует гораздо лучше второго (вернее второй не умеет рисовать вообще). И если я, как руководитель, не являюсь идиотом, конечно, я возьму (и взял) на работу первого, какие бы чудесные "корочки" не были у второго.

Думаю, примеров достаточно. Хотя приводить их можно бесконечно, — их очень много. Кстати, три четверти самых богатых людей на земле вообще не имеют никакого образования. И не очень по этому поводу расстраиваются.

Конечно, я самый первый буду за то, чтобы у вас был диплом о высшем образовании, и желательно профильном. Я не о том говорю, что диплом не нужен. Я о том, что если уж по каким-то причинам его нет, то крест на себе ставить не надо. И о том, что если он есть, это еще совершенно ничего не гарантирует. Потому что людей, которые не могут устроиться после окончания "программистских факультетов" по специальности, тоже очень много.

Примечание

Примеров таких действительно очень много. К сожалению, многие, получившие профильное образование, считают, что если они успешно прослушали все курсы, — то все уже сделано, и можно дальше ни о чем не беспокоиться. Это не так. Да, диплом, в частности, и очень хорошее университетское образование вообще — это хорошая база. И только. А все остальное нужно сделать самому.

Вопросы и ответы

В этом разделе приведены реальные вопросы наших читателей, касающиеся обучения программированию, которые они нам задавали на форуме и по почте, и наши ответы на них.

Скажите, с чего вы начали? Трудно было сначала? Я прочитал три электронных учебника, просмотрел кучу примеров, но ничего не понял — это значит мне не дано понять язык РНР?

И. Симдянов:

РНР не совсем удобный язык для изучения. Дело в том, что большинство РНР-программистов не изучало язык с нуля, а изучало его, уже зная либо С, либо Perl.

Поскольку он просто логически следует из Perl и широкое распространение получил именно из-за того, что программистам было легко переходить с Perl

на РНР. Кроме того, оба этих языка являются С-подобными, т. е. имеют синтаксис и конструкции языка С, являющегося одним из самых сложных языков программирования. Кроме того, для его серьезного понимания желательно знание очень многих вещей (HTML, SQL, UNIX, прикладные протоколы, хотя бы HTTP, стили программирования на С, Perl, Java).

Это не значит, что РНР сложно изучить: любой язык можно изучить, проработав хорошее руководство страниц на 900. Но так как РНР находится в начале своего развития, здорово эволюционирует от версии к версии, и еще не устоялся, хороших руководств нет. Обычно, перед изучением РНР советуют сначала изучить либо С, либо Perl, так как они уже устоялись и по ним не сложно найти хорошую книгу.

Изучение языков программирования очень индивидуально, кто-то делает это наскоком, кому-то требуется время. Но самое главное — при чтении руководств и учебников необходимо возвращаться и прорабатывать главу несколько раз, до тех пор, пока полностью ее не поймете. Так как следующие главы в основном опираются на предыдущие. При изучении С я даже откладывал руководство на полгода и потом возвращался к нему — дело после этого пошло гораздо веселее. Мне сложно давать советы по изучению РНР в качестве первого языка программирования, так для меня он был восьмым.

М. Кузнецов:

1. Не обижайтесь, но прочтение трех электронных книг и просмотр кучи примеров — это очень мало. Книги надо долго прорабатывать, а примеры надо разрабатывать самому. Пусть для начала и очень простые. Но — свои.
2. Начните с изучения "азов", на которых строится любой язык: разберитесь с логическими операторами, порешайте алгоритмические задачки, узнайте, что такое область видимости переменных, изучите работу с массивами, файлами, строками и т. д. Действительно, РНР для этого не самый подходящий язык для изучения основ программирования вообще.
3. По поводу "не дано" — неправда. Миллионы людей смогли, а вы не сможете? Дано, но только надо много и долго работать.

Все говорят, что программирование и математика это почти одно и то же. Ни разу не видел аналогии. Скажите, пожалуйста, кто, что думает по этой теме?

М. Кузнецов:

Нет, это далеко не одно и то же. Просто иногда так считается из-за того, что программированию у нас традиционно обучают на математических факультетах вузов, и у тех программистов, которые оканчивают эти факультеты, в дипломах, как правило, написана специальность "математик". Дело в том, что на заре становления информатики в России ею занимались преимущественно математики. В отличие, скажем, от той же Америки, где спе-

специалисты по информатике инженерные науки изучают в большей степени, нежели математические. Судить, какой подход лучше, я не возьмусь, плюсы и минусы есть и там и там. Американский подход (инженерный) более ориентирован на рынок, так как предполагает быстрое решение вопроса путем применения стандартных типовых решений. Российский подход более ориентирован на фундаментальное осмысление проблемы. Другое дело, что, наверное, не очень хорошо, что у нас в вузах пока еще отдается предпочтение только одному "математическому" подходу. Было бы неплохо, если бы на ряде факультетов у студентов была возможность выбирать, так сказать, между стилями обучения. Сейчас ситуация в этой области начинает исправляться, но пока не так, как хотелось бы. Ведь в программировании есть разные задачи: есть и такие, где знание математики нужно. Это, как правило, задачи, связанные с разработкой наиболее экономичных алгоритмов. В фирме Intel научные сотрудники этим занимаются. Графы там всякие строят, ищут кратчайшие маршруты обхода этих графов и т. д. А есть и такие задачи, и их большинство, где нужен подход скорее инженерный. Я не помню, чтобы мы с того момента, как ушли из университета и стали работать в SoftTime, сталкивались в программировании с какой-то серьезной математической проблемой. "Экономичность" алгоритма в большинстве случаев меня и заказчика тоже мало волнует. Заказчику главное, чтобы он мог быстрее начать работать, и чтобы программа работала. А если я ему скажу, чтобы он подождал пару месяцев, потому что мне надо придумать более красивый алгоритм, так он меня пошлет далеко, и всё. Даже если мне надо, к примеру, знать, какой алгоритм чего-либо наиболее быстрый, то я просто открою книгу и посмотрю. А математическое доказательство этого меня совершенно не волнует.

...Если вы спрашиваете, можно ли стать программистом, не зная математики, то ответ однозначный — можно. Повторюсь — я уже не раз говорил, что знаю неслабых программистов, у которых специальность не то, что не математическая, но даже не естественнонаучная.

Если интересно, то по большому счету программисту не помешают следующие разделы математики:

- ☐ дискретная математика (дискретный анализ, комбинаторика, математическая логика, теория графов), да и вообще вся алгебра;
- ☐ методы математического моделирования;
- ☐ теория приближенных вычислений (в вузовских курсах иногда обзывается "вычислительной математикой" или "численными методами");
- ☐ в некоторых случаях — теория вероятностей и теория игр.

И. Симдянов:

Небольшое уточнение. Те, кто говорит, что программирование и математика это одно и то же, исходят не только из этих позиций. Дело в том, что программирование в том виде, в котором оно существует сейчас, не всегда было таким — на заре программирования программисты были сильными математиками, которые для написания более экономичного алгоритма доказывали теоремы, а алгоритму вычисления синуса угла посвящались диссертации. И по сей день тем кто занимается системным программированием, поддержанием научного ПО, библиотек алгоритмов, графических, криптографический библиотек — знание математического аппарата необходимо. Но в 99,5% случаев программист имеет дело с прикладной программой, насыщенной арифметическими действиями (причем, которые за него производит машина). Современный программист является пользователем — только на более глубоком уровне — он использует наработки предыдущих поколений: вряд ли кому-нибудь в голову придет менять алгоритм вычисления синуса — есть задачи и поинтереснее, в том числе и с применением стандартной функции синуса. Программирование — это инженерная специальность, у нас действительно программирование изучают на математических факультетах, на западе для этого имеется отдельная специальность — Computer Science: там изучают принципы операционных систем, протоколы и т. п.

На самом деле программисту не мешают деньги, на которые можно купить библиотеки, реализующие все это "добро". Математический код отлаживается очень тяжело и время, потраченное на его разработку, часто стоит много дороже, чем готовые библиотеки. Впрочем, это относится не только к математическому коду, но и вообще к программным библиотекам.

Сам увлекаюсь комбинаторикой, в будущем собираюсь стать программистом. Мне интересно, насколько сильно нужна комбинаторика и высшая математика программисту?

Это сильно зависит от области специализации. Системному программисту не мешает, также будет полезна прикладному программисту, если он имеет дело с инженерной предметной областью. Жизненно необходимо для создания движков игр и научного ПО.

Программист-прикладник, как правило, имеет дело со стандартным кодом. Это раньше по алгоритму вычисления синуса в калькуляторах диссертации защищали, а сейчас имеются промышленные библиотеки, с которыми и имеет дело прикладной программист. Реализация собственных хитрых алгоритмов вряд ли потребуются — они реализованы в STL и других стандартных модулях — вот их изучить придется... Математика вам потребуется, только если она будет вашей предметной областью — иначе будет дешевле использовать чужой код, так как вряд ли вы пойдете для решения какой-либо задачи на создание собственного языка программирования или библиотеки —

коллеги ничего не поймут и предпочтут выкинуть ваш код и написать так, чтобы всем все было понятно.

Нужно ли для глубокого усвоения языков программирования высокого уровня, в частности, PHP, разбираться в высшей математике? И если да, то каких ее областей следует коснуться? Какие математические знания необходимы для Web-программиста (по образованию я — лингвист)?

Нет. Линейная алгебра бывает очень полезна в программировании, но не в Web-программировании. Для глубокого усвоения лучше изучить несколько языков программирования, включая Perl и C++, но это необязательно. Для того чтобы сносно программировать, достаточно школьного курса математики плюс матричное исчисление (все-таки его неплохо бы знать).

При изучении (повышении уровня) о предварительном изучении математики лучше не думать (там можно надолго застрять), а лучше сосредоточиться на основах операционных систем (Windows, UNIX), в основном это дает власть над кодом, а не математика.

> по образованию я — лингвист

Кстати, Ларри Уолл, создатель языка Perl (являющегося веховым, как C++, Java), тоже является лингвистом. Создавая свой язык, он старался, чтобы тот походил на естественный язык, каждый оператор действовал в своем контексте, т. е. если что-то в списочном контексте — он ведет себя как список, если в скалярном, как скаляр и т. п. PHP во многом наследует Perl, хотя он не такой целостный (его делали программисты, и они очень торопились). Если встретите его книгу "Программирование на Perl" издательства O'Reilly, 2005 — обязательно берите, в ней, конечно, более 1000 страниц, но она вам позволит более глубоко понять как Perl, так и PHP, и, возможно, вы уже не захотите программировать больше ни на каком языке — при изучении Perl это рядовое явление. Правда, книга не из легких, но написана с юмором, продавшись через нее, вам PHP будет "по колено".

...У нас в стране не было специальности Computer Science, где как раз больше внимания уделяется операционным системам, протоколам связи и т. п. У нас преобладали либо факультеты для инженеров, которые в производственных условиях работали (в политехах; они как раз ближе к Computer Science), и факультеты ВМК (вычислительной математики и кибернетики). Первые работали на производстве и еще 20 лет назад программирование и математика были неразделимы. Это сейчас, когда создан огромный базис, все просто, а раньше защищали кандидатские диссертации по алгоритмам вычисления синуса в калькуляторе. Вторые (ВМК) создавали научные основы производства и занимались собственно кибернетикой — моделированием систем, что требует серьезной математики, поэтому на данных факультетах ей в основном и занимаются. Создание операционных систем и ПО для 3D-моделирования тоже подразумевает неслабое знание математики.

Поэтому традиционно на вступительных экзаменах сдают математику, хотя отличные программисты нередко получают из троечников. Но большинство факультетов являются все-таки математическими и учат там не программированию (это средство), а математике.

Куда пойти учиться? Случилось так, что скоро получу диплом о высшем образовании, правда в химии, но что-то меня она уже не прельщает... Теперь хочется учиться по другой специальности, в области Web-программирования. Есть ли в Москве такой ВУЗ?

В Москве вузов очень много. Если касаясь программирования — ВМиК МГУ, к примеру. Но еще раз тянуть такую ляжку... Стоит ли? Классическое образование, особенно в таких динамичных штуках, как IT, всегда отстает от того, что реально творится и делается. Хорошие вторые корочки, вы, безусловно, получите, а вот реальные знания, которые можно реально применять в повседневной работе, вряд ли. Я здесь IT-область, скорее, имею в виду, — в химии, к примеру, с этим немного попроще. По поводу Web-программирования... Конкретно ему, почти 100%, вас нигде не научат. В лучшем случае вам предложат двухсеместровый курс плюс дисциплина специализации за все время обучения. И в довесок кучу мусора, которую вы либо уже знаете, либо она вам нужна как одеяла папуасам. Конечно, при многих факультетах (и в Москве этих мест масса) есть платные факультативные курсы, на которых в течение, скажем, 4—5—9 месяцев говорят о Web-программировании, Web-дизайне и т. д. По окончании — сертификат. Это уже, конечно, не второе высшее. Но и тут есть немало "но". Если эти курсы ведут штатные преподаватели университета, ходить вряд ли стоит (это если речь идет о знаниях, а не о корочках). А вот если занятия ведет практикующий человек, каждодневно сталкивающийся в своей работе с "реальной жизнью", — это посетить не мешает.

В нашей стране сейчас по статистике более 80% выпускников работают не по специальности (цифра, скорее, занижена, некоторые приводят данные, что более 90%). Профессия по диплому это ведь как имя, которое дается нам родителями, и про которое никто не спрашивает, нравится оно нам или нет. Тут дело не всегда в родителях, а в том, что в 16 лет, когда поступаешь в вуз, сложно осознать свое истинное призвание. Это приходит попозже... Если вы беспокоитесь о том, что с вашим химическим образованием вас не возьмут работать IT-специалистом, то этого делать не стоит. Сейчас, как мы уже не раз говорили, чаще всего берут по реальным умениям. Знаю даже историков по базовому образованию, которые сейчас не без успеха программируют в серьезных конторах серьезные вещи.

Мне 18 лет... И если честно, то я ничего не знаю почти, и хотя "комп" с 10 лет, а толку мало. И я по этому поводу очень расстраиваюсь. Конечно, есть какие-то знания, но мне кажется, этого мало, очень мало. Так сказать, сам себя критикую. Мне интересно, конечно, если это не секрет, а во сколько лет вы

начали и сколько лет учили тот или иной язык. И есть ли у меня перспектива в моих познаниях? Извините за пессимистичное настроение...

1. Во-первых, не расстраивайтесь, когда чего-то не понимаете. Точнее говоря, расстраивайтесь конструктивно. Такое настроение бывает у каждого, кто что-то пытается делать. Кажется, что вот, вроде перелез через стену, все понял, и... новая стена, на которую снова надо взбираться. И это — хорошо, и огорчаться этому не стоит. Хотя, признаюсь, это первое чувство, которое возникает у меня самого, когда что-то не получается. Правда, сейчас я научился быстро это перешагивать.
2. >Так сказать, сам себя критикую.

Может я немного не в тему, но, так сказать, на будущее... Вы знаете, самое трудное, наверное, будет еще дальше... Когда критиковать себя будете уже не вы, а совсем другие люди. И критиковать очень часто — совсем не конструктивно. И люди эти будут не просто с улицы, а те, которым вы верите или верили... Вот это действительно страшно. Потому что с собой еще можно договориться, а когда человек, которому ты веришь, говорит, к примеру, что тебе не стоит заниматься программированием (и т. д.), потому что у тебя мозги под это не заточены — вот этому противостоять очень сложно. Мне так говорили, и перешагнуть через это было очень сложно. В первую очередь, потому, что я привык верить тем людям, которые это говорили, чуть было не поверил и тогда. Теперь я понимаю, почему так говорят, и знаю, как этому противостоять. (По этому поводу есть очень хорошая пословица: "Я все сделаю для того, чтобы ты был счастливым при условии, что ты и дальше будешь оставаться несчастным"). А по "первости" можно очень просто, как Высоцкий пел, "попасть в чужую колею, глубокую" и перестать верить в себя. (Очень, кстати, хорошая эта песня. Она так и называется — "Чужая колея". Студентам ее особенно стоит послушать). А Ф. Д. Рузвельт так и говорил, что "самое сложное — поверить в себя и поверить себе". Именно поверить, а не зазнаваться, не заболеть звездной болезнью... То есть — просто хорошо знать себе цену. Не выше, не ниже — а ровно столько, сколько нужно.

3. И. Симдянов очень хорошо написал по поводу приказов (*Видел массу примеров, когда люди развивались таким образом, что могли программировать, если им приказывали: создать то-то, сделать то-то. А потом, когда они тому, кто им приказывал, уже были не нужны, они садились и смотрели телевизор в ожидании приказа, который почему-то никто не давал...*). Действительно, многие (как правило, отличники) достигают очень немало, пока с них чего-то требуют, пока их хвалят за сделанную работу. А когда ничего не требуют, то все: человек хиреет, чахнет, ничего делать уже не может... Есть очень хороший парафраз пословицы "Скажи мне, кто твой друг, и я скажу, кто ты". И звучит он так "Скажи мне, как ты переносишь одиночество, и я скажу кто ты". Я это понимаю так, что надо

учиться общаться с самим собой, быть самому себе интересным собеседником. Применительно к нашему разговору это значит, что нужно самому себе ставить задачи, решать их, а не ждать пока это сделают другие. Ведь, заметьте, — многие люди, когда уходят на пенсию (спортсмены, военные особенно), очень часто после этого чахнут и вскоре умирают. Мне думается, из-за того, что они привыкли жить "под приказом", и другой жизни просто уже не мыслят. И с другой стороны: лично хорошо знаю одного военного, который ушел на пенсию в 45 лет в звании подполковника и не помер, не спился, как многие его коллеги, а организовал свое дело. Теперь ему 52 — очень крупный бизнесмен. Вот что значит умение хорошо ладить с самим собой.

4. Программировать начал с Паскаля. Потом — Fortran 77/90, C++, Java, чего-то-еще-уже-не-помню, и вот теперь PHP и все, что с ним связано. Всегда тянуло заниматься сетевым программированием, но удалось заняться плотно только сейчас, а до этого занимались разработкой всяких программ научно-технической направленности. Заниматься программированием, как и И. Симдянов, стал в 18 лет и в том же самом месте, что и он (мы вместе учились — и программированию в том числе). Так что он говорит правильно — у вас перед нами огромная фора. Главное — не сбиться с пути. Кстати, я в школе вообще не мог программировать и очень этого боялся, а все программы за меня писал мой хороший школьный товарищ Сашка Котляревский, который потом закончил факультет ВМК. А потом как-то так получилось, что увлекся программированием.
5. > И есть ли у меня перспектива в моих познаниях?

Резюмирую. Перспектива у вас настолько громадная, что мне даже сейчас сложно предположить, чего вы сможете достичь. Главное — не растерять запал. Я очень рад, когда читаю, что те, у кого компьютер появился намного раньше, чем у меня — занимаются делом. Честно говоря, я этого даже не ожидал, поскольку большинство моих сверстников, у которых компьютеры появились достаточно рано, увы, ничего стоящего из них не выжали. Поэтому не расстраивайтесь, не отчаивайтесь — разбирайтесь со всем этим безобразием, которое сейчас называют IT-индустрией, а мы чем сможем — поможем.

Вот хочу узнать, что нужно знать перед тем, как искать работу программиста. Мне еще только 15 лет, но я уже имею приличные знания PHP и MySQL. Планирую доучиться до 11 класса, а потом уже искать работу. Так вот, пока есть время набрать знания, что мне нужно, чтобы быть уверенным, что я не останусь без работы? Пожалуйста, расскажите, где вы учились, с какого возраста начали программировать и когда начали работать. Мне это очень важно.

Я начал программировать по вашим меркам уже глубоким стариком — лет в 18, наверное. Или в 17. Точно помню, что уже в вузе. А вы в 15 уже знаете

PHP и MySQL. Поэтому в вашем случае, скорее, надо постараться не перегореть. Потенциал у вас хороший — главное не растерять его по дороге.

Мы в годы обучения в вузе сетевым программированием заниматься совершенно не собирались, а занимались разработкой научно-технических программ, так скажем. А получилось, что стали известными в области сетевого программирования. Это я к тому, что дело, по большому счету, не в том, что вы сейчас выучите, а в том, в какой форме будете к часу X. Потому что выучить-то можно, а технологии могут смениться. И что — менять профессию? Такого быть не должно. Поэтому — главное форма. Чтобы вы четко про себя знали, что если надо — за 3 недели выучите любой язык и минимально (для того, чтобы начать работать) освоите любую технологию. А то, что вы знаете сейчас — не суть важно. И что я знаю — тоже. Вот завтра появится новая технология — и кто ее быстрее освоит, тот и будет на коне. В этом смысле я за универсализм. Нам вот, к примеру, совершенно не нужен человек, знающий только PHP. (Вообще ведь это достаточно простой язык). Язык — это только инструмент для решения задач. А они могут быть очень разные. Значит, и инструмент в запасе желательно иметь не один, или, хотя бы, знать какие бывают. Потому что сегодня надо, к примеру, делать интернет-магазин, завтра — скрестить его с бухгалтерией 1С, после завтра — писать ПО для учета заказов, с Интернетом никак не связанное, а через неделю все это бросить и пойти делать ПО, которое считывает сигналы, посылаемые на порт собственно созданной схемой. То есть, понимаете — задачи перед вами могут стоять очень разные.

Вот вы говорите, "чтобы быть уверенным, что я не останусь без работы". Для этого надо пахать часов по 12 в сутки и знать, чего хочешь. Тогда я буду уверен, что что бы не произошло — у вас все будет прекрасно. Оба условия обязательны (много работать и знать, чего хочешь). Причем второе — даже поважнее первого: дизель может работать хоть круглые сутки, но если вхолостую — пользы от него никакой.

Ну и еще. Так сказать, техническая рекомендация. Поучите C++. И объектно-ориентированный подход. Просто для себя. Не мешает — точно. А с вероятностью 80% — не один раз пригодится.

Здравствуйтесь! Я сейчас выбираю вуз, в который буду поступать, посоветуйте, как это лучше сделать.

>Здравствуйтесь. Я сейчас выбираю вуз, в который буду поступать.

Обычно сначала выбирают специальность. И речь нужно вести, скорее, не о вузе, а о факультете какого-то вуза, и даже о специальности в рамках данного факультета. Сейчас на каждом факультете почти везде есть 3–4 специальности. Поэтому для того, чтобы вам кто-то что-то подсказал, нелишне знать, какую специальность вы хотите получить, принципиально ли для вас, в каком городе находится вуз, где вы сами живете, — Москва и Ново-Уральск разные города.

По поводу специальности. Это серьезный вопрос. И здесь надо сначала определиться, зачем вы идете в вуз. Как правило, грубо говоря, существуют два варианта:

- просто получить корочки и "общефилософское" образование;
- получить специальность, которая реально будет на вас работать.

Оба варианта достойны, и я, если честно, не знаю какой лучше. Можно пойти по первому варианту, получая отсрочку от армии, неплохое образование и корочки, а в свободное время изучая нужную себе специальность, программирование, скажем.

А можно — по второму варианту. Но тут надо серьезно поработать в плане выяснения того, куда лучше податься. Чтобы не скатиться к первому варианту. И стоит провести предварительную разведывательно-аналитическую работу. Допустим, вы хотите стать программистом.

Выясните, многие ли из окончивших данный факультет работают по специальности. Ах, никто? Или 5% выпускников, да и то потому, что их по блату пристроили родители? Значит, это — не ваш факультет.

Придите на лекцию по программированию на том факультете, на который собрались поступать. Потому что, если вы придете на лекцию по программированию и услышите вначале что-то типа "Основываясь на 121-м следствии из 143-ей теоремы Свигнера—Бзингера, изучению которого мы посвятили 3 предыдущие лекции, докажем, что 5 нормальная форма существует", — то это не ваш вуз, если вы хотите стать программистом. Вот если математиком — то да.

Посмотрите, кто в вашем вузе читает программирование. Если 50—60-летний профессор, то вряд ли он скажет вам что-то стоящее в смысле программирования. В смысле математики — сколько угодно.

Узнайте, работают ли сами преподаватели по той специальности, которую они преподают. Тут дело вот в чем. Многие специальности, особенно прикладного плана, в вузах преподают непрофессионалы. Ведь как обычно бывает? Вызывает декан какого-нибудь профессора по математическому моделированию в деканат и говорит: "Слушай, Вася Петрович, тут такое дело — мы открываем новую специальность, в рамках которой будет читаться двухгодичный курс лекций по программированию систем дистанционного управления. Надо бы тебе его взять на себя. Да, все понимаю — не твоя специальность. Да, понимаю, конечно, лучше пригласить спецов из НПП «На взлет», но ты же понимаешь — это деньги. И лишняя ставка. А у нас их нет и в ближайшее время не будет, даже если половина профессорско-преподавательского состава «коньки отбросит»". И идет Вася Петрович читать лекции по не своей специальности. Конечно, он их выучит. Но тонко-

стей, изюминок и изящных и простых объяснений (то единственное, ради чего стоит посещать лекции — все остальное можно в учебниках прочитать) вы от него не услышите.

У вас не должно остаться сомнений относительно того, что вуз — дело серьезное. Это 5—6 лет жизни, причем в самом ее расцвете. И от вашего нынешнего выбора зависит очень много.

Московский государственный инженернофизический институт (МИФИ). Специальность: "Комплексное обеспечение информационной безопасности автоматизированных систем". Объясните проще, что будешь делать на работе (если работать по специальности, конечно), если у тебя будет такая специальность?

И. Симдянов:

Предположу, что человек с такой специальностью должен уметь проектировать и писать ПО для всяких хитрых железок типа детекторов присутствия, систем видеонаблюдения, поиска скрытых камер, сигнализации; создание защищенных локальных систем (как на физическом, так и на программном уровне), распознавание образов, "прослушка" обычных и сотовых телефонов, а также защита оных от прослушки и т. п. Ну, я, по крайней мере, от выпускника ожидал бы, чтобы он мог приступить к решению таких задач.

М. Кузнецов:

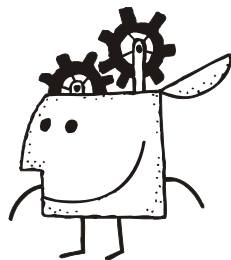
Сказать что-то конкретное по этому факультету я затрудняюсь, так как лично никого с него не знаю и ни с кем не встречался — ни с преподавателями, ни с выпускниками. Поэтому про качество образования, работают ли выпускники по специальности и т. д. — точно говорить не могу. Это проще вам выяснить — возьмите водки и отправляйтесь в университетскую общагу к тем, кто заканчивает этот факультет. За пару часов непринужденной беседы вам все расскажут — как учат, кто, где и кем работает после окончания факультета и т. д. Я серьезно. Это, пожалуй, один из немногих способов узнать гарантированную информацию. Другой — посидеть на нескольких лекциях по профильным предметам и послушать, что на них говорят. Здесь принцип такой. Пришли вы, к примеру, на лекцию по внутренним датчикам систем физической защиты. И если лекцию ведет профессор, д. т. н., и начинает ее словами "Сделав обратное Фурье-преобразование, мы увидим, что...", это — профанация. Потому что ничего не увидите. Так как скорее всего профессор какой-нибудь выдающийся математик, но из датчиков физической защиты у него самого только престарелая болонка, которая, если проснется, то затыкает, когда кто-то в дверь позвонит. А вот если лекцию по этому предмету читает спец из ФСБ, ФАПСИ или серьезной фирмы с охранным профилем, который полжизни промотался по "объектам", эти самые датчики устанавливая, то тогда — да. Тогда вы по окончании факультета будете знать очень немало.

Вообще говоря, МИФИ — институт очень серьезный. И если бы вы пошли туда учиться на физика, я бы гарантировал, что физик из вас по окончании вуза выйдет хороший. Потому что МИФИ всю жизнь специализировался на "ядренной бомбе". Этим и известен. И физика там поставлена очень серьезно. Как и математика. В части же "информационной безопасности" есть опасение очередной профанации. В очень многих вузах сейчас созданы подобные кафедры. Потому что нынче это модно и за это платят деньги. То, что диплом МИФИ это серьезный диплом — здесь вопросов нет. А вот получите ли вы какие-то реальные знания — не знаю. Немало бы сказало, если бы вы посмотрели, какие в принципе лекции читаются на данном факультете и кто их читает. Если есть возможность списать или сфотографировать лекционный план по данному факультету — сделайте это. Можете прислать мне на почту — я вам тогда скажу достаточно конкретно и с высокой вероятностью — кем вы будете и что будете уметь после окончания факультета.

А по вопросу кем придется работать и что делать... В принципе И. Симдянов вам ответил, чего бы он ожидал от выпускника с такой специальностью. Я бы ожидал примерно того же. Только чуть больше. Потому что спец по безопасности — это не просто диплом — это образ жизни. Пусть даже и по информационной безопасности. Вернее — особенно по информационной. Вот, к примеру, одна из типовых задач. В фирме (госучреждении, семье, кулукс-клане — нужное подчеркнуть) утечка информации. Ваша задача, как спеца по информационной безопасности, выяснить, откуда безобразие это происходит. И если вы придете в фирму в разгар рабочего дня, обвешанный аппаратурой и сумками с дисками с программами по поиску "троянов" и во всеуслышанье заявите, что пришли искать источники утечки информации... Я бы, как руководитель конторы, занимающейся безопасностью, вас тут же уволил. Для вашей же пользы. Потому что с таким подходом вы можете сильно укоротить себе жизнь.

Другая задача. Есть ВЧ. В смысле, воинская часть. И вам надо сделать проект организации систем контроля доступа. Еще пример. Сайт взломали хакеры — их надо найти. Самим. Без помощи ФСБ, МВД и т. д. Вот примерно таких умений я бы ожидал от человека, у которого в дипломе написано "Специалист в области комплексного обеспечения информационной безопасности автоматизированных систем и специалист в области программно-аппаратных методов обеспечения безопасности информации" и который прослушал курсы "Внутренние датчики систем физической защиты", "Системы контроля доступа", "Системы видеонаблюдения", "Системы сбора и обработки данных", "Внешние датчики (с полигоном периметровой системы) физической защиты", "Банковские телекоммуникационные системы и их средства защиты" и т. д.

Глава 2



Как найти работу

Базовая техника устройства на работу

В этом разделе мы рассмотрим базовую (общую) технику устройства на работу, а затем более подробно остановимся на некоторых отдельных этапах.

Итак, надо устраиваться на работу. С чего начать? С собеседования? С составления резюме? С телефонного звонка секретарю? Позвольте спросить, а почему вы так считаете? Вы решили начать с составления резюме, потому что у вас ручка с бумагой рядом оказались? А вы с телефонных переговоров, потому что вам телефон ближе? А у вас та фирма, в которую вы собрались устраиваться, снимает помещение на первом этаже в вашем подъезде, поэтому вы решили зайти пособеседоваться? Ну что ж, если вы так считаете, то представьте на минутку себя верховным главнокомандующим. Представили? Сидите, вы, значит, в Первопрестольной и решили пойти войной. На восток. Стратегический интерес у вас там. Приходят к вам поутру воеводы, и докладывают:

— Царь-батюшка, войска готовы к выполнению боевой задачи. Что приказать изволите?

— Польшу брать будем, — брякаете вы.

— Как Польшу, почему Польшу? — изумляются воеводы, — мы ж на восток собрались!

— А потому что Польша ближе, — демонстрируете вы железную полководческую логику.

— Но как же так! Польша же с нами мирный договор подписала! Да и вообще мы на восток собирались!

— На восток, говоришь... Ну, тогда возьмите город Баку. Не восток, конечно, но зато я там в армии служил. Дорог он мне этот город. Как память. А восток... Восток подождет.

...Вот примерно такой же логикой руководствуются многие из нас, устраивающихся на работу. Плохая логика. Потому что все должно начинаться с цели. И устройство на работу тоже должно начинаться с цели. А не с составления резюме или собеседования. С цели, и только с нее. То есть с ответа на вопрос о том, для чего вы устраиваетесь на работу. Ответ "Потому что уже надо где-то работать" не проходит. Нужно четко определиться, зачем вам нужна работа. Потому что не стоит идти работать научным работником, если ваша цель — зарабатывание денег и извлечение прибыли. Цель хорошая. Но и средства достижения должны быть соответствующие. В науке эту цель достичь сложно. Смолоду — практически невозможно. У нас в стране, по крайней мере.

Примечание

Да и в других странах, насколько я знаю, примерно так же. Общаясь с зарубежными коллегами, я пришел к выводу, что наукой в большинстве своем занимаются небедные люди. Которые заработали достаточно денег, чтобы позволить себе заниматься наукой.

Поэтому, если хотите зарабатывать деньги, то по большому счету есть только три честных варианта, как это можно сделать. Или заниматься бизнесом, или стать руководящим работником в какой-либо организации, ворочающей миллионами, или стать специалистом экстра-класса в своей области. Экспертом. Таким спецом, чтобы кроме вас больше не к кому было обратиться.

А если все-таки хочется работать ученым и деньги получать? Что нет никакого варианта? Есть, конечно. Если подумать, то варианты всегда найдутся. Только для того, чтобы увидеть эти варианты, надо освободить свое сознание от некоторых догм. В нашем случае это догма о том, что наука делается только в вузах и научно-исследовательских институтах. Это не так. Вернее, наука делается не только в этих учреждениях. На западе, к примеру, немалый процент научных достижений приходится на долю различных фирм. Наиболее яркие примеры всем известны: корпорации Intel, Xerox, Kodak и т. д. У нас в России тоже к этому идет: уже сейчас во многих компаниях есть научно-исследовательские отделы и лаборатории, занимающиеся научными исследованиями в рамках основной деятельности компании. И, надо сказать, достаточно успешно.

Примечание

К примеру, научно-исследовательский отдел одной московской компании, работающей на рынке разработки устройств защиты информации, за 3 года успешно решил серьезную научную проблему, над которой несколько лет бились радиофизические факультеты не одного вуза. Суть проблемы заключалась в разработке верифицированного метода улавливания и распознавания низкоинтенсивных сигналов среди спектра шумовых сигналов много большей интенсивности. В частности, решение этой задачи нужно для улавливания очень низ-

коинтенсивных сигналов от ПЗС-матриц видеокамер, находящихся в помещениях. А последнее необходимо, к примеру, для распознавания видеокамер, ведущих скрытое видеонаблюдение.

Поэтому, я уверен, что каждый может найти себе такую работу, которую он хочет. Главное, четко знать, что нужно (определиться с целью), а потом уже начинать сбор информации на предмет того, где находится то, что вам нужно. Этот этап не менее серьезен. Грамотно собрать информацию и ее обработать — это задача не из простых. Этому разведчиков по несколько лет учат. Но учат же, и успешно. Значит, и мы сможем. Тем более что наша задача не в пример легче тех, которые стоят перед разведчиками. Итак, вначале цель, потом — сбор информации. А уже после того, как вы четко определитесь, что же вы хотите, и где это место находится, уже можно приступить непосредственно к "штурму": к составлению резюме, звонкам в отдел кадров, собеседованиям. Хотя нет: между сбором информации и "штурмом" должен быть еще один этап: так сказать, непосредственная рекогносцировка местности. То есть определились вы, скажем, что хотите работать в фирме X, значит, надо узнать как можно больше о том, что это за фирма непосредственно на месте. Да не о том, чем эта фирма занимается и что производит, — это вы уже должны знать, а о другом: кто директор, кто заместители, как из себя выглядит секретарша, как их всех зовут и т. д. Здесь кашу маслом не испортишь, — чем больше вы знаете, тем более высоки шансы на успех. Во-первых, эта информация необходима для того, чтобы наметить наилучшие пути, по которым вы будете осуществлять "наступление". (И — отступления: всякое в жизни может произойти, и чтобы не было ситуации "вход — рубль, выход — два", лучше многое предусмотреть заранее). Во-вторых, вы еще раз проверите себя на предмет того, правильные ли выводы вы сделали из анализа первоначальной информации.

Итак, еще раз. Базовая техника устройства на работу включает в себя следующие основные этапы:

- ☐ формулировка цели;
- ☐ первоначальный поиск информации;
- ☐ непосредственный сбор информации о выбранном объекте;
- ☐ техника устройства на работу в нужную организацию.

В этом списке нет пунктов, касающихся составления резюме и собеседования, потому что они входят в последний пункт. Более того, нередко устройство на работу в выбранную организацию приходится начинать с... совершенно другой фирмы. Но об этом подробнее в соответствующем разделе.

А теперь рассмотрим каждый из этих этапов подробно. Но прежде поговорим еще об одном важном моменте: о настрое на поиск работы. О правильном психологическом настрое на поиск работы.

Настрой на поиск работы

Во-первых, поиск работы это такая же работа. Которой надо заниматься ровно столько же, сколько вы бы занимались своей основной работой. Пахать, в общем, надо. Тогда что-то обязательно получится. Вернее, не что-то, а то, что нужно. Во-вторых, если это дело для вас новое, то стоит потратить время на освоение методологии поиска работы. А вы что хотели... Свои тонкости есть в любом деле, и если заниматься этим делом успешно, то надо знать как наилучшим образом и с наименьшими потерями достичь намеченной цели. А если не изучать методологию, а действовать абы как, если не заниматься поиском работы каждодневно, а звонить от случая к случаю по первым попавшимся телефонам, если только встать на биржу, а самому ничего не предпринимать, то ничего не выйдет. Хорошего, по крайней мере. Ну и, в-третьих, вы должны верить в успех. Обязательно. Без веры в успех ничего предпринимать нельзя. Потому что, если вы не уверены в своем успехе, его у вас и не будет. Стопроцентно. Так как здесь работает всем известный принцип: "как корабль назовешь, так он и поплывет". А для того, чтобы вам лучше поверилось, приведу несколько вдохновляющих тезисов.

Во-первых, катастрофической безработицы, как таковой, нет. Что бы кто не говорил, и что бы не писали газеты. Нет безработицы. А уж в IT-сфере, тем более. Я уверен в этом и постараюсь сейчас свою уверенность передать вам, приведя многочисленные примеры, эту мою уверенность подтверждающие. А уж когда я слышу, что для выпускников факультетов ВМК нет работы по специальности, я точно знаю, что это ложь. Работа-то есть. Другое дело, что есть неумение искать работу. Потому что не привыкли мы пока еще за место под солнцем бороться. Мы к сказкам привыкли. Когда лежит кто-то на печи лет тридцать, а потом "встал, увидел, победил". А иногда есть еще и нежелание работать. Встречаю недавно одного знакомого, и он жалуется мне, что его уволили, и он теперь нигде не может найти работу. Спрашивает, не найдется ли у меня местечка. Спрашиваю, где работал раньше, отвечаю, что подумаю. В принципе работник вроде неплохой, а тут как раз несколько новых направлений намечается... Отчего ж не помочь. Но, подумав, отказал. Потому что, когда позвонил навести справки в то место, где он раньше работал, узнал, почему его уволили. И понял, почему он нигде не может найти работу. А уволили за то, что он провалил переговоры, явившись на них в подпитии. И, естественно, когда кто-то из тех работодателей, к кому он обращается, наводит справки на предыдущем месте работы, откуда ему дают "волчий билет". И таких примеров я знаю десятки. То есть проблема-то, получается, не в недостатке предложений на рынке труда, а в неумении этими предложениями воспользоваться. А это уже совсем другая проблема, к безработице, как таковой, никакого отношения не имеющая.

Примечание

Дорогие работодатели! Запомните, пожалуйста, этот пример, потому что он и ему подобные являются красноречивой иллюстрацией правила "Не гуляйте там, где живете". Вот этому человеку теперь для того, чтобы найти работу, надо либо уезжать в другой город, либо менять специальность, либо идти на низкооплачиваемую работу. Причем первые два пункта могут и не помочь. Хотя по своему техническому уму он, в принципе, заслуживает гораздо большего. А вот жизненного ума, увы, не хватило. И еще один немаловажный момент. В IT-мире, как и в любом другом (медицинском, журналистском и т. д.) все в принципе друг друга более-менее знают. И ничего удивительного в том, когда один начальник наводит справки у другого, нет. Все стараются получить максимум информации, и, по мере возможностей, объективной. Я, к примеру, иногда даю рекомендации, положительные, конечно, наиболее активным участникам форума softtime.ru. Так мне откуда только не звонили, чтобы проверить подлинность этих рекомендаций. Иногда совершенно незнакомые люди. В чем тоже нет ничего удивительного. И я звоню, и мне звонят. Запомните это, пожалуйста, и в том месте, с которого уходите, старайтесь оставить о себе благоприятное впечатление. Дверьми хлопать не надо. Даже если вас уволили незаслуженно. Наоборот, и в этом случае постарайтесь сделать так, чтобы ваш начальник по звонку другого начальника, или по запросу рекомендательного письма, дал вам хорошую характеристику. И ни в коем случае не допускайте увольнения с "волчьим билетом", как в рассказанном примере: один прекрасно проведенный вечер может обернуться месяцами пасмурных дней.

Более того, в настоящее время в нашей стране на рынке IT наблюдается скорее переизбыток рабочих мест, чем недостаток. Есть из чего выбирать. И вот тут кроется еще одна неприятная вещь, которую в народе называют "с жиру бесия". Штука действительно неприятная, потому что многие, скажем так, утрачивают чувство реальности и начинают творить вещи, за которые их потом вполне закономерно увольняют. Работа хорошая, начальник хороший, зарплата в принципе устраивает, значит, можно творить, что хочешь. ...Будучи недавно в Венгрии, разговаривал с директором их Web-студии. Средняя зарплата по Венгрии составляет 90—95 тысяч форинтов на данный момент (форинт — денежная единица Венгрии). В пересчете на наши рубли это примерно 12 000 рублей. Работник Web-студии в Венгрии получает чуть меньше. По нашему около 10 000 рублей.

Примечание

Для сравнения. Бензин в Венгрии стоит более 1 евро, а батон около 20 руб. А при зарплате в 90 000 форинтов стоимость детского садика в месяц составляет 60—65 000 форинтов. Это я к тому, что жить там, точно, в денежном смысле не проще.

То есть в среднем зарплаты примерно такие же, как у нас. А жизнь даже подороже. Так вот, при этом уровне зарплат работают раз в 5 интенсивнее. Потому что знают, что чуть что не так, их автоматически увольняют. В Венг-

рии безработица на IT-рынке действительно наличествует. И поэтому работу свою стараются выполнять не на 100, а на 1000 процентов. У нас зачастую при не самой маленькой зарплате некоторые хотят свободный график. В этом, в принципе, нет ничего страшного, это всегда проблема только того работника, который так хочет, и того начальника, который это позволяет. И если начальник позволяет, значит, все нормально. Опасность в том, что "халява" имеет свойство быстро кончаться, и многие оказываются к этому морально не готовы. Вот и получается такая искусственная безработица.

Еще один пример. Тоже заграничный. Хорватский. Это я все к тому говорю, что бороться надо и за работу, и за клиента. Раз уж движемся мы к какому-то капитализму. Раз время такое. Когда просто красивых глаз и волевого подбородка уже мало. Когда надо показывать, что ты умеешь. А то часто как получается? Приходит человек на собеседование и говорит, что он программист и хочет работать программистом. А работ своих не приносит, почему-то считая, что ему должны поверить на слово.

— Это можешь?

— Могу.

— А вот это?

— И это могу.

— А покажи работы, где ты делал это и вот это?

— А зачем? Нет пока таких работ. Но я точно уверен, что смогу.

Он-то уверен. И вполне может быть, что эта уверенность не беспочвенная и он действительно сможет. Но старайтесь всегда смотреть на ситуацию не только своими глазами, но и глазами вашего собеседника. В данном случае — работодателя. Который видит вас первый раз и совершенно не уверен, что вы что-то можете. Которому это надо убедительно доказывать. А значит, надо обязательно брать с собой все свои работы. Вернее, не все вообще. А все те, которые вас, как работника представляют в наиболее выгодном свете. И не только работы. Любые грамоты, сертификаты, дипломы, рекомендательные письма... В общем, все то, что поможет вам "набить себе цену". Причем у нас пока это сделать намного проще, чем в той же загранице. В той же Хорватии, пример про которую обещал рассказать. Захотели вы, скажем, провести Интернет посредством Wi-Fi в какой-нибудь курортный поселок. Так для того, чтобы это сделать вы для начала покупаете все нужное оборудование (за свои деньги), потом все монтируете и показываете, как все это работает. И только потом, если заказчику понравится ваша работа, он заплатит деньги. А если не понравится — демонтируете оборудование и оставляетесь с этим оборудованием. У нас мы такое видим крайне редко, в виде исключений. А у них это нередко общепринятая норма. Потому что конкуренция.

И еще несколько примеров, подтверждающих тезис о том, что безработица это скорее не объективная реальность, а субъективная. То есть зависящая от каждого из нас. Вы видите, сколько китайцев, вьетнамцев и прочих тайцев работают сейчас в России? Бригадами заезжают сюда и находят работу. А сколько различных иностранцев организовали в России прибыльный бизнес, став миллионерами? Причем у себя на родине они были никто, сумев достать денег лишь на билет до Москвы и чтоб на некоторое время на пропитание хватило. Ну, натуральные заграничные бомжи. А поди ж ты — в уважаемые люди у нас выбились. И наши работники с высшими и очень высшими образованиями в очередь к ним стоят, чтоб хоть кем-то взяли. Они смогли, а мы не можем? Значит, получается дело не в безработице, дело в нас. И получается, что глупо пенять на правительство, и чиновников в частности, и общую неблагоприятную ситуацию в целом, если не удастся устроиться на работу. Надо думать о том, что мы сами делаем неправильно. И перестраивать себя в соответствии с существующими условиями. И те, кто пользуется таким подходом, имеют гораздо больше шансов найти хорошую работу, чем, те, кто сидит без дела и ноет. Не нойте! А действуйте. И тогда все получится. Потому что не может не получиться. Другой пример. О наших соотечественниках, которые, уехав на постоянное место жительства за рубеж, сумели найти там себе достойную работу. Иногда даже более высокооплачиваемую, чем у тамошних коренных жителей. Так вот, соотечественники, с которыми я общался, не понимают, как у себя на родине можно не найти работу. Они на чужбине сумели, а мы на родине не можем. Хотя у них по приезду не было практически ничего. А у нас и связи, и дипломы, и родной язык... Значит, еще раз повторяюсь, все дело конкретно в каждом из нас. И если что-то не получается, то виноваты только мы, а не кто-то другой. И чтобы все получилось, надо не на всемирные катаклизмы ругаться, а заниматься самосовершенствованием, рассматривая неудачу не как полный крах, а лишь как сигнал о том, что что-то мы делаем неправильно.

Во-вторых, нет неходовых специальностей. Ну, нет и все тут. Примеров сколько угодно. Вот, казалось бы: насколько неходовая специальность — учитель зоологии. Ну, кому эти кишечнополосатые, извиняюсь, кишечнополостные нужны? Нужны, оказывается, если подумать, а не просто так выбрасывать свою специальность в мусорную корзину. Приведу пример из книги М. С. Норбекова. Одна женщина, учительница зоологии, будучи уже на пенсии, всю жизнь еле сводившая концы с концами, сумела стать долларовым миллионером. С помощью той самой зоологии. Потому что стала продавать в аптеки очень редкие и нужные в фармации яды, а потом и вовсе организовала собственный фармацевтический завод. А вы говорите, — зоология... А уж о программировании и других IT-специальностях вообще говорить не стоит. Работа под ногами валяется. Так нет, мы, чуть что не понашему, сразу лапки складываем и ждем своей участи. Дело в том, что при устройстве на работу специальность вообще особой роли не играет. Играет

только умение. Простите меня за эту крамольную фразу, но это реальное положение вещей. Можешь что-то делать? Докажи. Доказал? Иди работай. Вот и все. Сейчас дипломы уступили место умению. Я знаю много историков по специальности, которые работают программистами. Профессионально работают, в очень нехилых конторах. И знаю программистов, которые, прочитав, что "требуется специалист по Java", не шли на собеседование (хотя и хотели идти) только по тому, что считали себя специалистами по C++, и были уверены, что для разработки проекта на Java их не возьмут. Хотя от C++ перейти к Java не очень сложно, и любой работодатель, видя перед собой хорошего C++-программиста, с большой долей вероятности возьмет этого человека на работу. Ну, если уж только не что-то совсем срочное надо делать. Одному такому парню, который не шел на собеседование по этой причине, я посоветовал все-таки сходить и честно сказать, что он программирует на C++. Показать свои работы. А потом сказать примерно так: "Да, я понимаю, что вам нужен программист на Java. Но от C++ перейти к Java возможно, и даже за не очень большое время. Мне очень хочется работать в вашей фирме, и поэтому я сейчас намереваюсь обучиться Java. Думаю, у меня уйдет на это около месяца. Скажите, могу ли я рассчитывать на повторное собеседование после того, как посчитаю, что готов работать?". Его взяли сразу, сказав, что учиться лучше не в одиночку, а при поддержке коллектива.

А теперь напоследок зададимся вопросом: если все так хорошо, то почему же многие IT-специалисты работают, мягко говоря, не по специальности? Каменщиками на стройке, охранниками... И вы таких знаете, и я знаю. Почему, если удача, можно сказать, на плече сидит, так много неудачников? Работающих не там, где хочется, не так, как хочется, или вообще не работающих... А потому, что живем мы в мифах. Нами выдуманных или другими подсказанных — не важно. Важно то, что очень часто мы совершенно не задумываемся о реальном положении вещей. Выдумаем себе черте-что и считаем, что это истина, ни на что иное внимания не обращая. И, живя в своем выдуманном мире, удачу свою в упор не видим и бежим от нее. Вперед. На встречу со своей неудачей. Или просто сидим и ничего особенного не предпринимаем. Но, тем не менее, ждем, что должно произойти что-то хорошее, и очень удивляемся, когда ничего хорошего не происходит. Как в том анекдоте.

Анекдот в тему

Сидит Абрам около лотерейного лотка и плачет. "Что ты плачешь, Абрам?" — спрашивают прохожие. "Да вот, никак в лотерею выиграть не могу!". День сидит, неделю, месяц, год... И все плачет и плачет и день ото дня все горше. В конце концов, даже ангелам не по себе стало от его слез и пошли они к Богу. Пришли и говорят: "Дорогой Господь! Видеть больше уже не можем, как он рыдает. Ну, дай ты ему хоть раз в лотерею выиграть, ну пошли ему выигрышный билетик!" "Да я не против, — отвечает Бог, — только пусть он билет хоть раз купит!".

Цель

Итак, формулируем цель. Самое важное и самое первое условие успеха. Условие, о котором еще Сенека замечательно говорил: "Тому, кто не знает куда плывет, нет попутного ветра".

Пример с наукой и зарабатыванием денег мы уже рассмотрели выше. Еще несколько примеров для того, чтобы усвоить важность этого пункта.

Вы, к примеру, хотите работать программистом, но так, чтобы работать спокойно и суметь сохранить свою индивидуальность и решать задачи, которые интересно решать вам, а не кому-то другому? Тогда, если вы начинающий программист, вряд ли стоит идти в большие софтверные компании (за некоторыми немногими исключениями). А вот если вы уже спец "с именем" — можете идти куда угодно: вряд ли на вашу индивидуальность кто покусится.

А вы хотите самостоятельности в решении технических вопросов? То есть как делать проект, на каких языках работать и т. д.? Тогда вам не нужно идти рядовым программистом (ему такой свободы никто не даст), а, как минимум, менеджером проекта.

А вы хотите вообще полной самостоятельности в решении всех вопросов, в том числе и определяющих политику компании? В этом случае надо либо организовывать собственное дело, либо работать в выбранной компании одним из первых лиц. И первый и второй варианты вполне возможны. Причем работать в топ-менеджменте компании иногда можно начать сразу, не тратя многие годы на прохождение всех предыдущих должностей.

Примечание

Один из таких способов — это предложить руководству новое направление бизнеса. То есть прийти в компанию со своим направлением. Многие руководители по понятным причинам идут на это очень охотно, и не исключено, что после некоторого испытательного срока вы вполне успешно вольетесь в руководящий состав компании.

Теперь немного остановимся на одном моменте, касающемся организации собственного дела. Есть мнение, что для того, чтобы организовать собственное дело в IT-сфере или какой-либо другой области, нужно поработать пару-тройку лет в большой компании, работающей в этой сфере. Может быть и так. Я не разделяю эту точку зрения, так как знаю немало отрицательных примеров такого подхода. Вернее, этот подход вполне может быть использован, но с рядом значительных оговорок. Все рассмотреть, конечно, не получится, так как каждый случай индивидуален, но о самом основном мы сейчас поговорим. А самое главное в том, что опыт работы в такой компании не обязательно может быть положительным. А при отрицательном опыте с организацией собственного дела как-то обычно не складывается. И таких

случаев я знаю немало. И здесь в принципе все понятно, и ничего необычного нет. Представьте себе: пришли вы в компанию. Вы-то хотите набраться опыта, чтобы открыть собственное дело. А теперь зададимся вопросом: а что хотят те люди, которые вас брали на работу? Чтобы вы набрались опыта и открыли свое дело? Будем реалистами: вряд ли. В лучшем случае им на ваше дело наплевать. И вы будете решать те задачи, ради которых вас и взяли на работу. В худшем — они сделают все для того, чтобы вы это дело никогда не открыли. Потому что никто, находясь в здравом уме, не захочет собственными руками плодить себе конкурентов. Хотя сложно сказать, какой случай лучший. Ведь даже при благоприятном раскладе вы потратите несколько самых плодотворных лет своей жизни, скажем так, не на себя. Но, оговорюсь, это только мое мнение. Да, есть и положительные примеры такого подхода: когда человек, работая в корпорации, дорастал до определенного уровня, после чего решал, что ему пора открыть свое дело. Но у всех успешных примеров, которые я знаю, есть одно общее: все эти люди ни на минуту не забывали о своей основной цели. Да, они работали хорошо и были на хорошем счету. Но параллельно разнюхивали что и как, разведывали, как и где, учились всему, чему только можно обучиться, т. е. вели себя как разведчик в тылу врага. И это, пожалуй, единственный способ, чтобы "не потеряться". Потому что иначе засосет рутина, обязательно засосет: хват, а пяти лет как не бывало. Поэтому, если вы решили организовать свое дело и идете набираться опыта в какую-либо корпорацию, помните, зачем вы туда идете. И опять же, все начинается с цели. Просто сказать "иду, чтобы набраться опыта" — значит, ничего не сказать. Вы должны точно знать, какого именно опыта вам не хватает, и, если уж идете за опытом, то и идите только за тем опытом, который нужен, а не за каким-то другим. Считаете, к примеру, что нет опыта руководства командой, значит, надо идти менеджером проекта, а не рядовым программистом, так как в последнем случае только время зря потеряете. Нет опыта ведения переговоров? Значит, надо идти туда, где работают с заключением договоров. И так далее. А то часто как бывает: пришел набираться опыта и сидит, делает только свою работу: программирует. То есть что умел, то и делает: развития никакого. И опыта, соответственно, тоже. А для открытия собственного дела опыта надо немало: и в ведении переговоров, и в руководстве людьми, и в бухгалтерских делах... Вот и идите на такую должность, в которой все это будет наличествовать. Проявляйте больше инициативы. Только при таком подходе можно идти набираться опыта. Потому что иначе, скорее всего, будет стандартная ситуация под названием "поматросил и бросил". Знаю немало умных и порядочных ребят, которых в больших компаниях просто "перемололи" и выбросили. Думаю, и вы таких знаете немало. Поэтому если уж идете за опытом в большую копанию, то, повторюсь, постарайтесь идти на руководящую должность (конечно, чем выше должность, тем лучше), так как только в этом случае вы сможете составить реальное впечатление о положении дел.

В случае если вы идете рядовым сотрудником, боюсь, все самое интересное будет протекать мимо ваших глаз и ушей.

Теперь о самом важном в части формулирования цели. О том, как важно не совершить стратегической ошибки. То есть о том, чего вы хотите. Да не от работы, а от жизни вообще. О самой вашей главной стратегической цели. Потому что работа — это еще не вся жизнь, хотя и не самая малая ее часть. И нужно умудриться сделать так, чтобы ваша работа не шла вразрез с вашими основными жизненными целями. В идеале, конечно, работа должна являться основным средством достижения этих целей. Это уровень писателей, поэтов, Билла Гейтса, на худой конец. А то, как часто бывает: хочет начинающий программист зарабатывать очень много денег. Причем не просто много, а очень много. Ну, как Билл Гейтс тот же. И что же он для этого делает? Работает простым программистом в компании. Причем в одной. Ну и что, что крупной. Ну и что, что зарплата намного выше средней по стране. Вы же много хотите зарабатывать. А много зарабатывать и работать "на дядю" это как-то не очень совместимо. Потому что всегда есть некий уровень, за который не прыгнешь. В своем городе я этот уровень знаю. И в Москве знаю. И когда оказываю консультации по решению проблемы устройства на работу, говорю примерно то же самое, о чем сейчас пишу. И честно предупреждаю: вот за эту сумму на этой должности вы никогда не выпрыгнете. И не выпрыгивают. Как бы ни хотелось, что бы ни делали, и что бы ни обещали работодатели про проценты со сделок и прочие прелести. И тут ничего страшного нет, просто это надо четко понимать. Как и то, что зарабатывать серьезные деньги можно только работая на себя. То есть либо вести свой бизнес, либо выйти в своей области на уровень эксперта. (Варианты работы на себя, так или иначе касающиеся воровства чего-либо, я не рассматриваю).

Или, хотите вы, скажем, большого и долгого карьерного роста. А идете работать программистом в заштатную компанию в заштатном городе с перспективой стать через несколько лет старшим программистом в этой компании и руководить группой из трех человек. Тоже явное несоответствие цели и методов ее достижения.

А вы хотите, условно говоря, сказать новое, свое слово в программировании в частности, и в IT-сфере вообще? И идете работать на штатную должность в очень большую корпорацию. То, что новое слово — это похвально. Я обеими руками "за". А вот с фирмой, в которую идете, осторожнее. Тысячу раз осмотритесь. Особенно, если фирма большая. Боюсь, что, во-первых, в ней вас могут не услышать. Во-вторых, могут и не дать сказать. Особенно это актуально, если над вами много начальников. Да и вообще, в больших корпорациях индивидуалистов не всегда жалуют и нередко стремятся подстричь всех под одну гребенку. И говорить в таких конторах новые слова это все равно что, работая каменщиком на стройке, в разгар производственного процесса отсестать в сторонку и начать писать стихи. Могут и кирпичом за-

светить за то, что сачкуете. А то, что стихи гениальные и их обязательно записать надо было, это мало кого волнует. Главное, чтоб кирпичи так, как надо ложились, и скорость кладки была соответствующей. Ну а, в-третьих, может так случиться, что и вам самим ничего говорить уже не захочется. Потому что суета и текучка задавят. И времени на новые слова не останется. ...Я бы, может, и не стал так категорично говорить, просто до обидного часто вижу, как многих молодых и талантливых специалистов в больших корпорациях просто "перемалывали" и выбрасывали. И статистика в этом смысле у меня скорее отрицательная, чем положительная. Хотя и хорошие примеры есть. И пишу я так для того, чтобы заранее убить некоторые неоправданные надежды, если у вас таковые имеются, и попытаться сделать так, чтобы вы выбирали себе работу не по принципу "В омут с головой, а там, может быть, куда и вынесет", а шли на это дело с открытыми глазами и реальным представлением о природе вещей. Тогда все будет нормально. Тогда вы сами увидите, стоит вам идти в ту или иную фирму или нет.

Поиск информации

Итак, цель сформулирована. Необходимо приступить к поиску информации. И вот на этом этапе многие ошибаются примерно так же часто, как и при формулировании цели: либо этот этап напрочь пропуская, либо подходя к нему недостаточно серьезно. Под "напрочь пропуская" я понимаю то, когда покупается какая-то одна газета в неделю, в ней находится пара-тройка вроде бы подходящих объявлений о вакансиях, и на этом все заканчивается. Ну, или не заканчивается, а еще делается пара-тройка звонков в выбранные компании. Не в этом суть. А в том, что такой подход поиском информации назвать нельзя. А под "недостаточно серьезно" понимается то, когда либо используется только небольшая часть каналов поступления информации (допустим, только один Интернет или только объявления в печати), либо не проводится вдумчивый и грамотный анализ собранной информации. Обе эти ошибки могут привести к плачевным результатам. В лучшем случае ничего не найдете. В худшем можете устроиться туда, куда лично вам устраиваться категорически противопоказано, и потеряете год-другой своей жизни.

Ошибка в использовании недостаточного количества каналов поступления информации приводит к тому, что мы нереально оцениваем окружающую обстановку, и в результате, к примеру, живя в провинции едем, устраиваться на работу в Москву, хотя такую же работу и за примерно такие же деньги можно было найти и в своем родном городе. Просто мы об этом не знали. Вернее, даже не знали, а были уверены совершенно в обратном: что в нашем городе такой работы, которая нам нужна, найти нельзя. Насколько я знаю из своего опыта общения со многими соискателями, уверенность эта совершенно ни на чем не основана. Просто не искали, а поверили в не-

удавшиеся попытки поиска работы другими. А зря поверили. Как мы уже говорили, люди в восьмидесяти процентах случаев не могут найти работу, которая их устраивает, по причинам, далеким от объективных. К примеру, человек не умеет "постоять за себя" и не может договориться с работодателем о той зарплате, которая бы его устраивала. В результате он приходит к выводу, что высокооплачиваемой работы в его краях нет в принципе, и рассказывает о своих выводах направо и налево. И многие верят. И приходят к таким же выводам. Вот примерно так и рождаются слухи и сплетни. А верить-то не стоило бы. Хотя бы потому, что совершенно неправильно негативное мнение о двух-трех организациях, пусть даже и объективное, распространять на все остальные.

Принцип пролонгирования неудачи

Этот эффект, когда человек, столкнувшись с неудачей на одном-двух объектах, пролонгирует свое негативное мнение на все остальные объекты того же рода, я называю "**принципом пролонгирования неудачи**". Этот принцип справедлив, конечно же, не только в отношении рабочих мест, а и в отношении многих и многих объектов, с которыми сталкивается человек. Фразы "Все бабы — стервы", "Все мужики одинаковы" и т. д. — следствие того же принципа пролонгирования неудачи. Встретилась женщина с несколькими мужиками-алкоголиками, и вот она уже уверена, что все остальные представители мужского пола тоже алкоголики. Что пьют все. А те, которые не пьют — это всего лишь затаившиеся алкоголики, которые просто ждут удобного момента, чтобы показать свое истинное лицо. ...И попробуй, разубеди.

Более того, мнение других людей не часто бывает объективным, и верить на слово, не попробовав самому, вряд ли стоит. А необъективным оно бывает потому, что человеку очень сложно признать собственную неудачу: увольнение с работы, низкий оклад, плохое отношение начальства... Для некоторых это почти то же самое, что признать собственную несостоятельность. Поэтому в который раз повторяю: не верьте, не проверив. Может, у того человека, который вам сейчас рассказывает про ужасно низкие зарплаты и скотское отношение на предприятии X, просто не сложились отношения с начальством? А у вас, вполне возможно, сложатся, и зарплата у вас будет больше раз в пять, и отношение к вам будет хорошее. Честно скажу, я не верю, что в достаточно крупном населенном городе, в котором проживаю, нельзя найти многое из того, что есть в Москве. Конечно, по сравнению с Москвой найти сложнее, никто не спорит. Но можно. Не верил — и находил. И работы, и подработки, и не только себе. Потому что опыт показывает, что деньги проще всего искать там, где живешь. Но и Москву тоже не сбрасывал со счетов, потому что глупо спорить с тем, что рынок труда там больше. Но и с Москвой мы работаем, не живя в Москве. Что вполне возможно, и, главное, менее накладно. **Поэтому, не верьте, а проверяйте!** Или собирайте статистику. Статистике уже более-менее можно верить. Имея статистику, можно увидеть некоторые закономерности.

Итак, работа с информацией подразделяется на два основных пункта:

- сбор информации;
- анализ информации.

Касательно анализа информации мы уже о многом поговорили "по ходу дела", напомним лишь основные моменты. То, что собранную информацию надо подвергать анализу и осмыслению, вроде бы всем известно, так как неосмысленная информация представляет собой просто кучу мусора в голове и ничего более. Во-первых, для того, чтобы информацию анализировать, и, желательно, быстро, ее надо уметь систематизировать. Так, чтобы вы всегда знали, где и про что у вас написано. Об этом мы подробно поговорим в абзаце про файлики-тетрадошки, которые обязательно надо завести и в которые складывать всю поступающую информацию. Во-вторых, к анализу информации нужно подходить трезво, т. е. в том массиве данных видеть не то, что вам хочется увидеть, а то, что на самом деле есть. И третье, и самое главное, при анализе информации нужно все время помнить о вашей основной цели, это тот самый критерий, на основании которого делаются выводы.

А вот на сборе информации остановимся поподробнее.

Сбор информации

Образно говоря, сбор информации можно сравнить с ловлей рыбы. И от величины площади, где вы раскинете свои сети, и от того, какого размера будут ячейки в этих сетях, зависит, какую рыбу вы поймаете. Раскинете небольшую сеть и с крупной ячейкой — ничего не поймаете. А если и поймаете крупную "дичь", то она вашу сетку прорвет. А если хотите золотую рыбку поймать, то ячейку нужно делать как можно мельче. Переводя на наш язык — не нужно чураться никаких источников информации, которые мы сейчас попытаемся классифицировать.

Но до того как непосредственно приступить к сбору информации, нужно ...завести тетрадошку. А лучше несколько тетрадошек. Или несколько файликов, кому как удобнее. А в этих файликах-тетрадошках нужно записывать все, что касается интересующего вас вопроса: поиска работы. Всю информацию абсолютно. Потому что поиск работы — дело достаточно трудоемкое, связанное с большими объемами информации, запомнить которые и ничего не перепутать для большинства из нас нереально. Куда вы обращались, точные координаты этого места, что вам там сказали, когда и куда нужно позвонить или направить резюме. Не будет лишним, если вы будете конспектировать и содержание ваших разговоров с работодателем: здесь лишнего не бывает — чем больше информации вы будете заносить в ваш "конduit", тем лучше. Записывайте все обязательно: системность еще никому не повредила. А вот несистемность повредила многим. Потому что, если вы, к примеру,

забудете, что в пятницу в 12.30 вам нужно позвонить и договориться о собеседовании на понедельник, то, боюсь, собеседование может и не состояться.

Итак, рассмотрим доступные нам источники информации.

□ Интернет.

Безусловно, первое, что приходит в голову. Интернет незаменим тем, что может помочь без особого труда получить информацию о том, что творится на "дальних рубежах". В той же Москве, если вы живете не в ней, или Владивостоке. И в Америке, и во Франции, и в Парагвае. Владивосток не подходит, потому что вы живете от него за тысячу километров? И что? Работать можно и удаленно, и вполне успешно. Одна дорожно-строительная фирма получила выгодный контракт на совместную работу с американцами, просто сбросив им по электронной почте свое резюме. А найти заграничное финансирование по строительству дорог куда сложнее, чем в IT-сфере. Чем многие, кто ищет, и пользуются. Знаю одну Web-студию, которая работает исключительно на Австралию. Очень продуктивно работает. Находясь при этом в России. И даже не в Москве и Питере. И даже в не крупном городе. В неотмеченном на картах ауле, можно сказать, они находятся. Насколько я знаю, они за все время сделали только два или три проекта на Россию. Потому что, справедливо рассудив, что в их ауле сайты никому не нужны, а в близлежащем населенном пункте сайтостроителей уже переизбыток, решили руководствоваться мудрой пословицей: "Не лезть на чужие рынки, а изобретать свои". Нашли за границей наших соотечественников и забросали их своими резюме с примерами работ (кстати, нарисованными не для кого-то, сугубо для портфолио), и в Австралии нашли отклик. И имеют сейчас очень приличный денежный оборот в валюте. А вы говорите, Владивосток далеко...

□ Информация о вакансиях в прессе.

Обязательно читайте все объявления о вакансиях, которые печатаются в ваших местных изданиях. Причем такие газеты желательно покупать каждый день, чтобы не пропустить ничего важного. Нужные вам объявления подчеркивайте и, еще лучше, переписывайте в ту тетрадь, о которой мы говорили. Чтобы потом целенаправленно и ничего не пропустив начать работу с этими вакансиями.

□ Информация о предприятиях.

Здесь имеется в виду, что вычитывать только явные объявления о вакансиях иногда недостаточно. Лучше в дополнение к этому определить список тех предприятий, на которых могла бы быть востребована ваша профессия, и рассылать туда резюме. Причем стремиться надо к тому, чтобы ваше резюме попало на глаза не рядовому сотруднику отдела кадров, а одному из руководителей, желательно первому лицу. Да не отдела кадров, а самого предприятия. Потому что отдел кадров это такая штука, которая

действует только автоматически: у них есть "спущенный сверху" список вакансий, с которым они и сверяются. И если вашей специальности там нет, то последует автоматический отказ. То, что причина может быть достаточно курьезной, думаю, вас радовать не будет. К примеру, у них написано, что им требуется программист, а вы в резюме укажете, что претендуете на должность системного программиста. Не удивляйтесь, что вам откажут. Таких курьезов я знаю десятки. Работники отдела кадров не обязаны и, как правило, не разбираются в тонкостях вашей специализации и им надобно "чтобы все буквы совпали".

О необходимости встречи с первым лицом

При устройстве на работу изыскивайте любую возможность попасть именно к первому лицу, или, на крайний случай, к одному из первых заместителей. Что уже хуже. Но, в принципе, все же лучше, чем просто отдел кадров. Который плох не только курьезами, а еще и тем, что больше, чем ему положено знать, он знать не будет. Все о своем предприятии, а главное, о планируемых перспективах, знает только руководитель. И только он принимает решения о кадровых назначениях. Согласитесь, будет обидно, если руководитель задумал открыть программистский отдел месяца через два, в котором вам бы только и работать, а вы туда не попадете. Потому что прислали резюме в отдел кадров, а у них на тот момент разнарядки на вакансии программистов не было. А вот если бы попали к руководителю, который знал о том, что через некоторое время ему понадобятся программисты, то был бы совсем иной расклад. Более того, знаю не один случай, когда руководители предприятий, прочитав грамотно составленное резюме, приглашали человека на собеседование даже, если специалисты его профессии не требовались. И, поговорив раз-другой с человеком, принимали решение о создании нового направления "под него", что я считаю одним из самых лучших вариантов трудоустройства. Поэтому: не упускайте своих шансов и пробивайтесь только к первым лицам. Как угодно, но пробивайтесь: обольщайте их секретарш, подлавливайте их на выходе с предприятия, узнавайте личные номера телефонов и адресов электронной почты... Потому что только руководители реально принимают решения и только они смогут вам дать реальные ответы на ваши вопросы, в том числе и на вопрос: нужен им такой специалист, как вы, или нет? А заместители различных рангов нередко могут человека месяцами муржить и, в конце концов, так ничего не решить. И не бойтесь встречаться с первыми лицами: зачастую с ними общаться намного проще, чем с их заместителями. Потому что им не надо за счет вас повышать свое чувство собственной значимости и городить ради этого различные условности, чем нередко грешат их многочисленные заместители, борющиеся между собой за то, кто ближе "к первому".

□ Ваши знакомые.

Это, пожалуй, один из самых основных моментов в нашем списке. Во-первых, потому, что им часто пренебрегают. По разным причинам. Во-вторых, потому что работа, которую вы найдете через своих знакомых, или у своих знакомых, по понятным причинам нередко является наилучшим вариантом трудоустройства. Теперь немного о том, почему мы нередко пренебрегаем нашими знакомыми и родственниками. Во-первых, из-за того,

что нам нередко (а некоторым и очень часто) кажется что "хорошо там, где нас нет". Ну, а где мы есть, следовательно, не очень хорошо. Вот из-за этого мы часто и лезем за тридевять земель в непролазные дебри, забывая начисто о ближнем круге. А зря забываем. Во-вторых, нам часто ошибочно кажется, что мы о своих знакомых знаем все, что только можно, и сами за них решаем: "Ну какую он мне работу может предложить, я ж знаю, чем он занимается... Никакой не может. Значит, и сообщать ему о том, что я ищу работу, незачем". В эту ошибку впадают очень многие, хотя все, здраво рассудив, приходят к разумному выводу, что мы нередко свою семью-то знаем не так, как надо бы, не говоря уж про родственников и знакомых, тем более, не самых близких. Тем более тех, с которыми не общались уже несколько месяцев, а нередко и лет. И многие упускают нередко хорошие шансы. К примеру, многие мои, близкие знакомые и родственники совершенно не в курсе некоторых направлений, которыми я занимаюсь и занимался. И в этом ничего удивительного нет. И о ваших знакомых и родственниках вы наверняка знаете не все: курьезных случаев на предмет того, как многие даже близкие люди не подозревали о некоторых видах деятельности друг друга, я знаю очень много, — к сожалению, объем книги не позволяет пересказать их, чтобы еще раз убедиться в том, как мало, зачастую, мы знаем друг о друге. Отчасти это происходит из-за того, что нам свойственно мыслить "подсознательными категориями". Простой пример на тему, что это такое. К примеру, ваш сосед вам сказал фразу "Я сегодня намок под дождем". При этом в ваше подсознание вкладывается то, что: во-первых, сосед был там, где идет дождь, и он промок. А далее наше подсознание делает обобщение "Сосед был на улице": ну потому что привыкли мы к тому, что под дождем промокают обычно на улице. И вы уже считаете, что сосед промок, когда шел по улице. Что, вообще говоря, неверно и ваш сосед вам такого не говорил. А промокнуть он мог, к примеру, пока стоял и курил на открытом балконе. И вот на эту психологическую удочку попадают очень многие. И очень немало приемов скрытого управления подсознанием основано именно на этом нехитром приеме. Еще один пример на эту тему. Я, к примеру, уже серьезно "окопался" в IT-отрасли и обзавелся новыми знакомыми, которые знают меня как успешного специалиста в IT-сфере. Я езжу приглашенным специалистом на собеседования по приему на работу программистов и других IT-специалистов, пишу книги по программированию, рецензирую программистские работы, руковожу программистским отделом в крупной компании и т. д. Это все факты, которые все видят и знают. А вот дальше их подсознание делает, казалось бы, логичные домыслы, однако ничего общего с реальными фактами не имеющие. К примеру, о том, что я закончил факультет ВМК. А это не верно: я заканчивал другой факультет. Тем не менее, процентов 70 наших клиентов уверены в том, что я закончил именно ВМК. Другой домысел недавно услышал во время совещания на одном крупном химическом предприятии, где я попытался вставить свое скромное слово при обсуждении одной химической проблемы. На что мне

было одним руководителем замечено, что при всем уважении ко мне он отзывается мне в обсуждении этого вопроса, так как "программисты в химии ни черта не соображают". Он был очень удивлен, когда узнал, что моя базовая университетская специальность "фотохимия и фотофизика", и по сути обсуждаемого вопроса я даже написал когда-то несколько научных работ. Совершенно обескураженный руководитель только кричал и все повторял "Как же так, вот ведь как бывает". Бывает, и очень нередко. Поэтому узнавайте, чем занимаются ваши знакомые, и предлагайте им свои услуги. Всем предлагайте: поиск работы такая штука, где лишнего не бывает. Более того, даже если вы пока при работе, но она вам не очень нравится, сообщайте всем вашим знакомым, что вы на грани безработицы, и что "не сегодня-завтра"... И чем больше вы об этом говорите и чем чаще напоминаете, тем лучше: информация имеет свойство запоминаться только при многократном повторении. Вот и делайте так, чтобы у всех ваших знакомых в голове засело, что вам нужна работа, чтоб они скорее имя свое забыли, чем то, что вам нужна работа, — тогда, возможно, и помогут. Я специально так подробно остановился на этом разделе, так как считаю, что поиск хорошей работы через "своих людей" один из самых эффективных методов решения "рабочего вопроса". Более того, это вообще прекрасный универсальный способ решения почти всех проблем. Потому что людям могут помочь только люди.

Техника устройства на работу в нужную организацию

Ну, а теперь, когда цели определены, информация собрана и определена та фирма, в которую нужно устроиться (или круг таких фирм), можно приступить ко всем прочим атрибутам трудоустройства: составлению резюме, собеседованию и т. д. В который раз повторяюсь, устраиваясь на работу, вы должны знать, для чего вы туда устраиваетесь, т. е. иметь цель. Сделать карьеру, получать много денег, переждать армию, найти жену... Что угодно, лишь бы цель была. Здесь мы рассмотрим только первый, наиболее интересный, вариант: т. е. когда вы устраиваетесь на работу на предприятие, на котором хотите сделать карьеру и стать первоклассным специалистом в своей области.

Примечание

На этом этапе вспомним о сборе и анализе информации. Дорогие карьеристы! Когда вы выбрали предприятие, которое вам вроде бы по душе, в первую очередь соберите информацию о его руководителе на предмет того, является ли он сам карьеристом. Потому что только в этом случае имеет смысл устраиваться на работу в эту фирму. Если руководитель не карьерист, не повышает сам свое мастерство, не стремится к новому, — вы не достигнете своей цели.

Итак, устраиваемся. Предположим, выбрали мы фирму Х. Теперь идем устраиваться. Но... не в фирму Х, а, скажем, в фирму Х-4, т. е. в родственное предприятие, но которое в вашем списке трудоустройства стоит одним из последних. Потому отстает по всем показателям от того предприятия, в котором вы мечтаете работать, порядка на четыре. Таким образом, суть в том, что вначале надо устраиваться на наименее желательное предприятие. А потом — по возрастающей, все более приближаясь к тому предприятию, которое вы выбрали.

Примечание

Один из тех ребят, которых я тренировал при устройстве на работу на очень хорошую должность в очень солидную компанию, тренировался больше полугода. Сменив три компании за это время. И при собеседовании в той компании, ради проникновения в которую все и было затеяно, ему задали вопрос на предмет того, что это он компании как перчатки меняет. Наилучший вариант ответа на этот вопрос — это сказать правду. Мол, тренировался я. Для того чтобы в компанию своей мечты поступить на работу без осечки. И в нескольких словах описал суть тренировок. Это — беспроигрышный вариант. Во-первых, часто после такого ответа работодатели оказываются в состоянии этакое психологического гротеска и забывают о том, что хотели спросить дальше. Во-вторых, всем приятно слышать, что человек полгода тренировался только для того, чтобы пройти собеседование.

Такой подход имеет много плюсов. Во-первых, если устраиваетесь в первый раз, вы приобретете необходимый опыт составления резюме, ведения беседы с руководителем. Во-вторых, при таком подходе вы наименее эмоциональны, потому что не боитесь сделать ошибку: ведь если вам откажут (что для первого раза очень вероятно), вы не очень-то и огорчитесь, поскольку работать на этом предприятии и не собирались. Зато постепенно вы будете действовать все увереннее, и более четко научитесь контролировать свои эмоции. Практика показывает, что трех-четырёх раз такого "пробного" устройства бывает достаточно практически всем для того, чтобы потом на решающем собеседовании не сделать ошибок.

И еще раз повторюсь на тему того, что беседовать нужно только с первым лицом на предприятии, — потому что окончательное решение за ним. А если на предприятии кадровые вопросы решают все кому не лень, то в такую фирму лучше и не устраиваться. Если вы уже определились, в каком подразделении собираетесь работать, то неплохо перед беседой с руководителем предприятия поговорить с начальником этого подразделения. В том случае, если вы ему подойдете, он сам наилучшим образом представит вас руководителю. Если возможности для такой беседы по каким-то причинам нет, то идите к секретарю руководителя. Секретарю оставьте свое резюме с небольшим письмом, в котором, как рекомендует М. Е. Литвак, нужно написать примерно следующее.

Глубокоуважаемый В. А.!

Я очень хочу работать на Вашем предприятии. Я слышал о Вас как о прогрессивном человеке, дающем возможность расти молодым специалистам, что для меня очень существенно. В свою очередь я, в случае моего устройства к Вам на работу, сделаю все от меня зависящее, чтобы принести как можно больше пользы Вашему предприятию.

Я готов работать с испытательным сроком, но готов также в любую минуту уволиться, если стану в тягость фирме, не дожидаясь при этом окончания испытательного срока, и могу даже написать заявление об увольнении с открытой датой.

Это письмо я пишу для того, чтобы не отнимать у вас время: в случае, если мое письмо и мое резюме Вас заинтересуют, вы назначите мне удобную для Вас дату приема. Если же нет, то можете отказать мне в приеме: об отказе я узнаю от Вашего секретаря.

С уважением, N. N.

Обычно при помощи подобных писем удастся попасть на прием к первому лицу. Ну, а уж дальше все будет зависеть от того, как вы себя на этом приеме будете вести. Ну и конечно, ваше резюме должно быть грамотно составлено. О составлении резюме и о том, как вести себя на собеседовании, мы и поговорим в следующих разделах.

Составление резюме

Для начала о том, что не надо писать в резюме. Примеров того, как не надо писать слишком много, поэтому здесь рассмотрим лишь характерные. В скобках или в примечаниях даны мои комментарии, почему именно приводимые фразы писать не следует. Явный бред, который тоже видишь нередко, типа того, что "я просто тащусь от C++", я здесь все-таки приводить не буду. Примеры неудачных фраз взяты из моей коллекции резюме, написанных в разное время разными авторами.

- ☐ Самостоятельно разрабатывал программы... (Из этого следует, что есть много программ, которые вы разрабатывали несамостоятельно.)
- ☐ Рабочие модули данной программы предназначены для того, чтобы... (Становится очень интересно, для чего же предназначены нерабочие модули данной программы.)
- ☐ Хобби — гулять на природе.

Примечание

О хобби. Вообще, о хобби в резюме можно и не писать. А в некоторых случаях делать это просто категорически не следует. Но уж если пишете, то пишете правильно. Не надо писать, что ваше хобби это хождения на дискотеки, — это никому не интересно. Если пишете про хобби, так пишите только то, что может быть интересно вашему работодателю. К примеру, "игра на гитаре" написать можно, — будете полезны на корпоративных вечеринках. Писать "чтение книг" тоже не следует, потому что для образованного человека чтение книг это не хобби, а жизненная потребность, без которой он не может обходиться.

- ☐ ПК владею в совершенстве. (Это подразумевается, если вы ищите вакансию IT-специалиста.)
- ☐ Люблю размышлять над поставленной задачей... (Надо же, какое счастье... Написал бы еще, что "не терплю думать над поставленными задачами и вообще ненавижу все задачи.")
- ☐ Готов работать круглые сутки... (Это — неправда. Уже на вторые сутки ваше желание работать резко пойдет на убыль.)
- ☐ Немного говорю по-английски...
- ☐ Легко обучаем, очень исполнительен, порядочен и находчив... (Избегайте собственных личностных оценок. Во-первых, вы можете ошибаться, во-вторых, это ваше субъективное мнение — у других оно может быть не таким, ну, и, в-третьих, нужно быть скромнее. ...Дайте руководителю самому потренироваться в определении ваших личностных качеств.)
- ☐ Готов работать за небольшую зарплату... (Неужто, правда, готовы? А если серьезно, то никогда нельзя снижать себе цену, — поднять уже упавшую планку удастся очень немногим.)
- ☐ В школе, в 7 классе побеждал на олимпиадах по черчению... (Это сообщение лучше оставить для вашего вечера воспоминаний — в серьезном резюме такие фразы ничего кроме смеха не вызывают. Еще более смешно, когда человек, устраиваясь на работу программистом, пишет, что он "занял третье место в 9 классе на городской олимпиаде по биологии"... Ладно бы еще в 11 классе первое место на областной олимпиаде по программированию. Это еще, куда не шло.)
- ☐ Прошу рассмотреть мое резюме и дать ответ... (По непонятной причине эту фразу многие вставляют в конце резюме. Между тем, она совершенно не нужна: и так понятно, что резюме нужно рассмотреть, если его прислали.)
- ☐ Готов выполнять любые работы... (Очень нередкая фраза и в резюме, и на собеседованиях. Ну не готовы вы к этому... И никто не готов: зачем сразу обрекать себя на рабство...)

Образец грамотно составленного резюме приведен ниже. В примечаниях даны необходимые пояснения и комментарии.

Примечание

Чтобы не выдумывать вымышленных персонажей, для удобства при составлении резюме я буду иногда основываться на своем опыте, естественно, изменив некоторые данные. Для простоты предположим, что "я" ищу должность программиста, и поэтому в резюме отражаю только то, что связано с моей программистской карьерой. И вам советую то же самое: если вы специализируетесь во многих областях, но получаете работу по какой-то одной из них, — не валите все в кучу, только про одну и пишите, если этого достаточно. Про вашу разносторонность лучше сказать при личной встрече. К примеру, моя базовая университетская специальность фотохимия и фотофизика, а окончил я химический факультет. (С которого, кстати, вышло немало программистов. Вообще, химики очень нередко перекавалифицируются в программистов.) Но, чтобы не пугать людей с самого начала, я об этом в резюме вообще постараюсь не упоминать, а опишу только ту часть своей карьеры, которая связана с программированием.

Образец резюме

Иванов Иван Иванович

Адрес: 245625, Нижний Новгород, пр. Гагарина, д. 25, кв. 1. Электронная почта: ivanov@yandex.ru

Телефоны: (8312) 651111 (рабочий)

(8312) 651122 (домашний)

(8312) 652222 (мобильный)

Примечание

Полужирным шрифтом выделены основные разделы резюме — наличие каждого из них в резюме необходимо. Порядок следования пунктов тоже лучше оставлять без изменения: он общепринят и многие работодатели уже привыкли к подобной структуре резюме, и когда они видят другую структуру документа, это их внутренне немного раздражает.

Цель

Получение должности руководителя департамента IT-технологий ГК "ИнтерСвязьСтрой".

Примечание

Цель должна быть краткой — не более двух предложений, а лучше одно, как в приведенном примере.

Текущая должность и место работы

Ведущий программист ГК "ИнформСвязьСтрой". Текущие обязанности включают в себя:

- ☐ разработка программного обеспечения;
- ☐ руководство группой по эксплуатации и монтажу радиоэлектронного оборудования;
- ☐ переговоры с клиентами по вопросам заключения договоров;
- ☐ проектирование программных продуктов.

Краткое профессиональное резюме

Специалист в области Web-программирования, защиты информации от несанкционированного доступа и разработки научно-технического программного обеспечения. В течение 8 лет участвовал в разработке крупных проектов в Web-программировании и создании научно-технического ПО. Область профессиональных и научных интересов: проектирование программных продуктов, проектирование охранных систем любой сложности и любого периметра объекта, проектирование систем связи. Автор 38 научных работ.

Образование

Сентябрь 1996 г. — июль 2001 г.. Нижегородский государственный университет им. Н. И. Лобачевского. Специальность — фотохимия и фотофизика. Кафедра — спектроскопии. Тема дипломной работы: Исследование фотохимических и фотофизических реакций методами спектроскопии электронного парамагнитного резонанса.

Профессиональный опыт

- ☐ Сентябрь 1999 г. — июнь 2000 г. Инженер ОНИЛ (отраслевой научно-исследовательской лаборатории) фоторезистов НИЧ (научно-исследовательской части) ННГУ им. Н. И. Лобачевского.
- ☐ Июнь 2000 г. — декабрь 2005 г. Младший научный сотрудник ОНИЛ фоторезистов НИЧ ННГУ им. Н. И. Лобачевского.
- ☐ Октябрь 2000 г. — июнь 2002 г. Архитектор проекта АСУП на ФГУП "Завод им. Я. М. Свердлова". (Работа по трудовым договорам.)
- ☐ Март 2004 г. — по настоящее время. Ведущий программист ГК "ИнформСвязьСтрой".

Профессиональные достижения

- ☐ Разработка крупных программных проектов для решения научно-технических задач:
 - программный комплекс "Photon", предназначенный для расшифровки сложных УФ-спектров;

- программный комплекс "Quantum", предназначенный для визуализации и обработки данных квантово-химических расчетов;
- программа "VisualESR" для оцифровки и начальной обработки спектров электронного парамагнитного резонанса;
- программный комплекс расшифровки спектров ЯМР (ядерного магнитного резонанса) и др.;

Примечание

Для экономии места я ставлю "и др.", а вы пишете все, что имеете. Вернее не все, а наиболее презентабельное: всякие мелочи, если можете показать рыбу покрупнее, перечислять не стоит. Но если мелочи приятные, то их можно записать на компакт-диск с вашими проектами, который прилагается к резюме. Компакт-диск — вещь обязательная, если вы претендуете на должность программиста или должность менеджера проектов. Если же метите выше, то его можно и не прикладывать: к примеру, от руководителя департамента IT никто требовать того, чтобы он сам кодировал, не будет.

- ☐ разработка программных продуктов в области Web-программирования;

Примечание

Здесь точно также перечисляете наиболее существенные проекты.

- ☐ аналитическая работа. Составлен перспективный план развития компании на ближайшие 3 года. Проанализирована тенденция развития рынка IT-услуг на ближайшие 3 года;
- ☐ написание книг. Написано 6 книг по Web-программированию и способам защиты информации от несанкционированного доступа в Web.

Знание платформ и инструментов

- ☐ Операционные системы: Windows (9x, 2000, XP), UNIX.
- ☐ Языки программирования: C, C++, Pascal/Object Pascal, Fortran 77/90/95, Java, Visual Basic, HTML/DHTML/XML/XSL, PHP, Perl, UML.
- ☐ Системы управления базами данных: InterBase, MS SQL, PostgreSQL, MySQL.
- ☐ Среды разработки и инструментарий: Microsoft Visual Studio, C++ Builder, JBuilder, Delphi, Compaq Visual Fortran, Rational Rose.

Награды, грамоты, дипломы и сертификаты

Примечание

Здесь указываете все свои награды и сертификаты, которые имеете.

Иностранные языки

Английский разговорный и технический свободно, немецкий разговорный.

Рекомендации

Будут представлены по запросу.

Собеседование

Собеседование — это один из самых важных этапов. Потому что именно на нем все и решается. И подготовиться к нему надо очень ответственно. От вашего стиля общения на собеседовании, от того, какое вы произведете впечатление, зависит то, примут вас или нет. Потому что это провозжать вас потом по уму будут. А встречают всегда по одежке. И на собеседованиях особенно хорошо видно, как можно "одеться не так".

Номер первый. "Ой, простите. Я, это, значит, к вам, ну на это, ну для собеседования". Садится на краешек стула, всем своим видом демонстрируя полную готовность немедленно упасть в обморок при любом осложнении ситуации.

Номер второй. Вытаскивает из заднего кармана брюк вчетверо сложенную и немало помятую бумажку, глядя на которую рождаются самые нехорошие предположения относительно ее происхождения. Ан, нет, — оказывается, это рекомендации.

Номер третий. При решении задания иногда исподлобья смотрит на членов комиссии взглядом, в котором такая смесь тоски и агрессивности, что с него впору рисовать картину "Допрос пионера-героя гестаповцами-садистами".

Номер четвертый. Забыл свою ручку, а потом в задумчивости обмуслякал ручку, любезно предложенную ему председателем комиссии.

Номер пятый. Говорит достаточно уверенно, складно, но от волнения плюется. Понятно, что более всего достается председателю, поскольку по большей части испытуемый обращает свое внимание на него.

Номер шестой. Все тестовые задания сделал правильно, но практически ничего не говорит, ограничиваясь кивками головы. На просьбу пояснить свое решение сжал скулы и с ненавистью осмотрел всех членов комиссии. Уже дважды за сегодня пострадавший от чужих слюней председатель на всякий случай отодвигается подальше.

Номер седьмой. На вопрос, чем не устроило предыдущее место работы честно ответил, что там начальник — придурок. Понимая, что в скором времени также будут говорить про них, члены комиссии осуждающе качают головами.

Номер восьмой. Девушка. Размалевала себя так, как будто пришла устраиваться не в службу технической поддержки, а в бордель.

Номер девятый. Очень долго рассказывал комиссии, что его дедушка однажды видел живого Ленина, когда встречал бабушку, работавшую уборщицей в Смольном.

И так далее... Это из раздела "как не надо". Поговорим о том, как надо.

Примечание

Более подробно о некоторых вопросах психологии общения см. главу 4.

Во-первых, надо быть опрятно одетым. Это очень важно. Одеваться нужно в классическом стиле. Лучше всего заранее узнать о стиле одежды, принятом в той организации, в которую устраиваетесь.

Во-вторых, возьмите с собой все свидетельства, дипломы, сертификаты, трудовые договоры, обязательно компакт-диск с проектами... В общем, все то, что может хоть как-то повысить ваши шансы, вплоть до водительских прав на бронетранспортер и разрешения на хранение дедушкиного маузера, доставшегося вам по наследству. Все эти бумаги (вернее, их хорошо сделанные копии) нужно сложить в красивую, но строгую папку, которую вы предъявите по требованию.

Не забудьте блокнот с ручкой: очень неплохо смотрится, когда проходящий собеседование нечасто что-то записывает за членами комиссии.

При разговоре будьте спокойны и уверены и смотрите в глаза собеседнику. А если даже вы не спокойны и волнуетесь, то вы все равно должны быть спокойны и уверены. Тренируйтесь! Мысли формулируйте четко, говорите не спеша и внятно. Но и не медлите. Как можно меньше вдавайтесь в подробности по поводу предыдущего места работы, даже если вас будут вызывать на откровенность. Вернее, тем более, если вас будут вызывать на откровенность. Безусловно, не стоит говорить, что начальник на прежнем месте был очень хороший, потому что разрешил вам уволиться по собственному желанию после того, как вы, будучи в хлам пьяным, избивали монитор, потому что только что с его помощью прочитали письмо, в котором любимая девушка написала, что в ваших услугах больше не нуждается. Конечно, такие интимные подробности пребывания на предыдущем месте работы лучше вообще никогда не вспоминать. А на вопрос о том, почему ушли или уходите с предыдущего места работы лучше сказать что-нибудь более нейтральное. И вообще, старайтесь говорить только о том, что подтверждено вашим резюме и приложенными к нему документами. Откровения лучше оставить до будущих корпоративных вечеринок. И говорите только о том, что раскрывает ваши лучшие стороны. Обо всем не очень хорошем вообще не упоминайте, даже в порядке самокритики, даже если вы к ней очень склоны! Вы на собеседовании, а не на приеме у врача. Конечно, всю свою

речь нужно заранее хорошо отрепетировать. Хорошо — это не значит просто написать конспект разговора, и посмотреть на него два раза в транспорте по пути на собеседование. Хорошо — это отрепетировать все до мельчайших оттенков голоса, до жестов рук, до движений мимических мышц... Очень неплохо, если будете репетировать не только вместе с зеркалом, а еще и вместе с магнитофоном, на который себя запишете. Чтобы когда вам по вашему же сценарию надо говорить вдохновенно и проникновенно, вы не шипели, словно взбесившийся щитомордник, которого потревожили проходящие рядом туристы, а когда надо улыбаться вы действительно улыбались, а не демонстрировали звериный оскал, которому вас еще в детстве обучил проживавший по соседству подполковник ГРУ.

И точно так же нужно отрепетировать ответы на вопросы, которые вам начнут задавать сразу после вашего прочувствованного монолога о собственной персоне. Ответы, конечно, нужно заготовить заранее. Потому что джентльменский набор вопросов, задаваемых на собеседованиях, давно известен.

- ☐ Почему вы ушли с прежнего места работы? Что вам там не понравилось?
- ☐ Почему вы выбрали именно нашу фирму?
- ☐ На какую должность вы претендуете? И почему именно на нее?
- ☐ Как вы представляете себе свою будущую работу?
- ☐ Чего вы хотите добиться в карьерном плане?
- ☐ На какую зарплату претендуете?
- ☐ Каковы, на ваш взгляд, ваши достоинства и недостатки?

Одним из очень хороших вопросов является следующий: "Почему вы считаете, что достойны большей зарплаты, чем та, которую имеете сейчас (или чем та, которую получали на предыдущем предприятии)". Очень многие из тех, кому я и мои коллеги задавали такой вопрос, отвечали на него, скажем, так, не очень умно. И почти никто не догадывается сказать, что мол, выросла моя квалификация, человек должен расти и в материальном плане, увеличился объем труда и т. д. Как правило, отвечают примерно следующее (в скобках даны комментарии, поясняющие, почему так или иначе отвечать не стоит):

- ☐ У меня появилась семья. (А причем здесь я?)
- ☐ У меня родился ребенок. (Его тоже не я вам рожал.)
- ☐ Я взял кредит и мне нужно его выплачивать. (Как будто я лично под дулом пистолета принуждал вас брать этот кредит.)
- ☐ Я уже третий год работаю на предприятии, а зарплата все та же. (Очень частый ответ. И не только на этот вопрос. Причина состоит в недопонимании того факта, что платят (ценят, уважают) не за количество (проработанных лет, прожитых дней и т. д.), а за качество работы на данном

временном отрезке. Очень часто можно слышать примерно следующее: "Мне уже 40 (45, 50) лет, а я получаю меньше, чем тот парень, которому всего двадцать три стукнуло!" Ну и что из того, что кому-то сорок? С возрастом только артериальное давление закономерно повышается, а все остальное (зарплата, карьерный рост) — как заслужишь.)

- Я стал позже уходить с работы. (Если это результат увеличения объемов труда, то так и надо говорить. Потому что очень нередко на работе засиживаются из-за того, что не могут грамотно спланировать свой день, в результате чего рабочий день возрастает, а объем выполняемой работы прежний или даже меньше. А за такое не зарплату повышают, а увольняют, точно так же, как списывают вхолостую работающий дизель, который топлива и масла жрет больше, а электроэнергию в час выдает все меньше и меньше.)

Что делать, когда увольняют

Да, бывает, приходится увольняться. Не хочется об этом говорить, добавляя ложку дегтя в оптимистический настрой данной главы, но никуда не денешься — из песни слова не выкинешь. Тем более, что правильно, поумному увольняться у нас не очень умеют. По поводу увольнения говорить особо много нечего кроме одного и самого важного: при любом увольнении нужно пользоваться заповедью "Не навреди". Да не начальнику, а себе. Поэтому не стоит слать небесные кары на головы ваших недоброжелателей, даже если таковые имелись в неограниченных количествах, не стоит резко вспоминать все свои высокие связи и грозить ими на каждом углу... Всего этого делать уже не стоит. Хотя бы потому, что после драки кулаками не машут. Потому что для вас этот бой уже закончен, и вы его проиграли. Потому что, если увольняют именно вас, значит, вы по каким-то причинам или по их совокупности слабее тех, кто остался. И единственное, что можно сделать в данной ситуации, — это уйти с наименьшими потерями. В лучшем случае, "сыграть в ничью", если вам сразу удастся найти примерно такую же работу, с которой вас уволили. Ну и в самом лучшем случае найти работу лучшую. Вот к этому и надо стремиться. Для чего надо взять... Да не бейсбольную биту, которую вы очень хотите сломать о голову вашего начальника, а взять себя в руки. И трезво подумать, почему уволили именно вас. И трезво себе на этот вопрос ответить, для того, чтобы в будущем не повторить тех же ошибок. И еще надо смотреть на вещи позитивно: т. е. рассматривать увольнение не как конец света, а как шанс на поиск новой, лучшей работы. И тогда все будет нормально. А то ведь как бывает: уходит человек с визгами, воплями и проклятиями. Ищет, ищет виноватых. Конечно, находит. Это самое простое — найти виноватых. С этим делом на Руси издревле все хорошо было. Только виноватых находят, а лучше почему-то не стано-

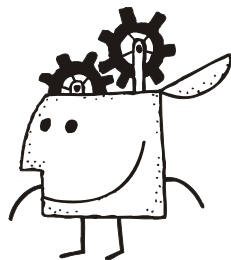
вится... Устраивается этот человек в другое место, и снова его увольняют. И та же песня. Только виноватых уже больше... А нет бы ему подумать: "А почему? Почему именно меня?" Потому что, если не думать, то все будет повторяться от раза к разу, только с худшими вариантами. **Потому что нельзя устраиваться на новую работу, не искоренив того, что приводило вас к ошибкам на старой работе.**

Ну, а если вас уволили несправедливо, к примеру, по статье за тот проступок, который вы не совершали, то приходится отстаивать свои права. Здесь тоже ничего страшного нет. Не вы первый, не вы последний. Только, опять же, — кричать и плакать не надо. И подкарауливать начальника у подъезда, чтобы натравить не него вашего бультерьера, тоже не надо. Потому что в любом деле самое главное — трезвая голова. И потому что бюрократическую машину голыми руками не возьмешь. Бюрократическую машину может сокрушить только более сильная бюрократическая машина. В вашем случае, это, скорее всего, суд. Потому что надо срочно идти ...в библиотеку или книжный магазин. И взять или купить Трудовой кодекс. И комментарии к нему (обязательно). И изучить это все тщательнейшим образом, даже если вы по складу ума математик и от юриспруденции вас тошнит так же как страдающих морской болезнью в семибальный шторм. Изучать надо даже в том случае, если вы собрались не сами защищать свои права, а поручите это дело адвокату. И, изучив, вы, скорее всего, что-нибудь неправомерное в действиях вашего руководства найдете. Потому что закон, как известно, что дышло...

Примечание

А для того, чтобы вам веселее было искать, сообщу, что сейчас существует немало людей, которые на этом зарабатывают деньги. И неплохие, надо сказать. Они устраиваются в фирму, работают там чуть-чуть, и потом делают так, чтоб их уволили. А потом идут в суд и судятся со своим начальником до тех пор, пока он либо деньги не отдаст, либо пока его на скорой помощи прямо из зала суда не увозят. Я лично знаю двух человек, которые зарабатывают на жизнь таким способом. И очень неплохо, повторяюсь, зарабатывают. Причем не только на компенсациях, которые им выплачивает фирма, а и на консультациях, которые они оказывают ...юристам. Потому что в вопросах трудового законодательства эти товарищи, не имеющие не то что юридического, а вообще никакого образования, такие доки, что куда до них некоторым юристам...

Глава 3



Работа в команде

Будем считать, что при правильном выполнении рекомендаций предыдущей главы работу мы нашли. Теперь нам надо на ней "окопаться". Причем желательно так, чтобы было комфортно. Приходя на новое место работы, мы неизбежно сталкиваемся с новыми людьми, попадаем в те или иные группы, общаемся с коллегами. И от того, насколько мы правильно это делаем, зависит, насколько нам будет хорошо работать на новом месте.

Введение в социальную психологию

Прежде всего о том, что такое социальная психология.

Социальная психология — это наука о поведении человека в обществе. Причем под обществом понимается что угодно: и большой коллектив, и небольшая группа, и толпа.

Примечание

В изложении социальной психологии в этом разделе я буду ориентироваться на фундаментальные работы Ле Бона и М. Литвака.

Группа — это люди, между которыми существуют реальные **интеллектуальные** отношения и которые направлены на достижение общей цели. Когда между людьми нет никаких отношений, группой их назвать нельзя. К примеру, люди, идущие по улице, — это не группа. Люди на демонстрации — это тоже не группа, а скорее уже **толпа**. Болельщики на стадионе — это тоже толпа. А вот семья, класс, коллектив — это группы. Теперь определим, чем же группа отличается от толпы. Ведь в толпе между людьми тоже есть отношения, и тоже можно предположить наличие общей цели. К примеру, у болельщиков на стадионе есть общая цель: ожидание победы своей любимой команды. У демонстрантов и митингующих тоже есть общая цель. Но, тем не менее, ни одна из этих категорий группой не является. В чем же дело? А дело все в разнице отношений, которые объединяют людей в группе и

толпе. В группе людей объединяют **интеллектуальные** отношения, а толпе — **эмоциональные**.

Из этого небольшого абзаца с определениями следуют важные практические выводы. Во-первых, и это в большей степени касается руководителей, из определения группы следует, что у людей в группе есть общая цель. Так вот, чтобы успешно управлять группой, надо определить цель этой группы. К примеру, вы руководитель отдела в фирме. Отдел — это группа. И от того, насколько вы правильно определите цель, для которой эта группа сформировалась, во многом зависит то, насколько эффективно вы своим отделом будете управлять. К примеру, если я, как руководитель отдела, буду считать, что группа собралась для того, чтобы писать программы, я сделаю большую ошибку. Потому что собрались не за этим. Не может человек приходить в группу для удовлетворения чужих потребностей. **Человек в любую группу приходит только ради удовлетворения собственных потребностей!** Это одно из основных правил социальной психологии. А ведь, если я буду думать, что он пришел только ради программирования, то это — удовлетворение чужих потребностей, в данном случае — учредителей фирмы, продающей эти программы. Поэтому честный и грамотный ответ будет звучать так: человек, в первую очередь, пришел ко мне в отдел за зарплатой. Безусловно, этот ответ справедлив не для всех, но для большинства он именно таков. И при таком подходе я, как руководитель, уже буду четко знать, чем его заинтересовать. И чем наказать. Потому что заинтересовывать и наказывать нужно только тем, зачем человек приходит в группу. То есть, если, к примеру, человек пришел за хорошей зарплатой, а я его за проступок буду наказывать словами "Вася, как тебе не стыдно", при этом продолжая платить зарплату, то какие бы я слова не говорил, это не возымеет ровным счетом никакого действия. Поэтому пришедшего за зарплатой нужно наказывать и поощрять только зарплатой. Точно так же, если, допустим, моя фирма дает "бронь" от армии, и человек пришел именно за этим, то наказывать его за проступок зарплатой — бесполезное дело. Ему можно вообще ничего не платить, все равно никуда не денется. А вот если ему сказать "Вася, я пришел к выводу, что по тебе кирзовые сапоги обрыдались", то работоспособность резко повысится. И понимать такие вещи — крайне важно.

Дорогие руководители!

Не впадайте в ошибку и не считайте, что люди пришли к вам удовлетворять ваши потребности! Они пришли в вашу группу удовлетворять свои потребности. И если вы дадите им такую возможность, то они удовлетворят и ваши потребности. Казалось бы, все просто. Но... Руководители фирм считают, что их сотрудники должны, засучив рукава, работать на процветание руководителей. И страшно возмущаются, когда сотрудники этого не делают. Научные руководители в институтах сплошь и рядом считают, что их подопечные пришли заниматься наукой, наукой и еще раз наукой. Три из ста может быть. А все остальные пришли написать диссертацию. И только. Ко-

гда я сказал это одному вузовскому профессору, он мне ответил, что я очень пошло мыслю, и что не все люди так низко пали. Я ему в ответ предложил провести простой эксперимент: собрать всех своих аспирантов и объявить, что диссертацию они не получают. Ну, или получают, но лет через двадцать. И посмотреть, кто после этого у него останется. Боюсь, что уйдут даже те "трое из ста". И таких примеров можно приводить сколько угодно. Поэтому: от того, насколько четко вы определите потребности каждого входящего в группу и потребности всей группы в целом, и насколько эти потребности сумеете удовлетворить, зависит ваше процветание. Повторюсь: к определению целей нужно подходить трезво и не наделять других своими целями. Для чего следует как можно лучше вжиться в другого человека и попытаться посмотреть на мир его глазами. Многие удивительные вещи открываются... И когда вы поручаете сотруднику какое-то задание, спросите себя: почему оно может быть ему выгодно, зачем он это должен сделать. Ответ: "потому что я приказал" не проходит. Если выгодно — он его сделает. Если невыгодно — не сделает. Выгода может быть самая разная: от заработной платы до возможности во время выполнения задания заняться личными делами, к примеру, совместить поездку в командировку с любовным романом.

Примечание

Это правило, конечно, действует и в жизни вообще. И точно также о нем стараются забыть. Простой пример. Жена хочет, чтобы муж бросил пить. Мотив понятен. Но муж, как часто бывает, имеет на этот счет свою точку зрения, и бросать пить не хочет. Поэтому жене лучше бы не ругаться, а спросить себя: "А почему бросить пить мужу должно быть выгодно?" И найти то, что в его системе ценностей (именно в его, а не жены) лежит выше стакана. Потому что только тогда он сможет бросить пить. Только тогда, когда он найдет занятие более выгодное, т. е. лежащее в его системе ценностей выше выпивки. А если стакан с вином это вершина его системы ценностей, то тут ничего не сделаешь. Надо с этим либо смириться, либо дальше жить без него. И с сотрудниками также. Если вы видите, что ваш сотрудник, скажем, деньги получать любит, но желательно, чтоб за просто так, то ему надо объяснить, что за ваше хорошее дело вы ожидаете от него тоже хорошего дела. А если не понимает, если может только брать, а отдавать не способен, значит, он достиг своей вершины в системе ценностей, и разговаривать дальше бесполезно. Надо либо платить ему как в партийную кассу, не требуя взамен никакой работы, либо расставаться. Без обид, нервов, слез и слюней с обеих сторон. Чисто по-деловому.

Дорогие сотрудники!

Вы тоже не допускайте распространенную ошибку, и не считайте, что вам кто-то что-то очень много должен, и все они на протяжении всей вашей жизни просто отдадут вам свой долг. Деньги, хорошее отношение, высокий социальный статус — это все зарабатывается, а не дается просто так. И если ваш руководитель понимает ваши потребности и старается их удовлетворять,

вы пойдите ему на встречу и удовлетворите его потребности. Потому что самая выгодная сделка — это та, в которой выигрывают обе стороны. У нас же часто происходит непонимание этого правила как со стороны руководителей, так и со стороны подчиненных, и все просто стараются друг друга обмануть, а потом удивляются: как же так, вроде все так хорошо было, и почему-то вдруг все развалилось. А потому что, хорошо, никакого не было, и никакого "вдруг" тоже не было. А была лишь возрастающая социальная напряженность. О том, что нужно делать руководителям, чтобы такого не было, мы поговорили чуть выше. Но и подчиненным это тоже надо понимать. Часто же происходит совсем по-другому. Если начальник сполна удовлетворяет потребности сотрудника, его начинают считать "лохом", и начинают "садиться на шею". Можно, конечно, и так делать. Но только не надо после этого удивляться, что вам будут попадаться плохие начальники. Потому что с хорошим с таким подходом не сработаетесь. Потому что хороший начальник, он ваши потребности удовлетворит, но и о своих не забудет. Он даже может дать сначала "посидеть на его шее", но быстро сбросит, если не увидит ожидаемого им результата. Потому что хороший начальник — это не бесконечный альтруист, как многим хочется думать. А совсем наоборот. Это человек, думающий в первую очередь о своей выгоде. Но так как он хороший, т. е. умный, начальник, он знает, что получить свою выгоду не сможет, если не сделает так, чтобы подчиненным тоже было выгодно делать то, что ему надо. И создает для подчиненных хорошие условия работы. Что нередко некоторыми подчиненными воспринимается как возможность получать блага просто за факт своего существования в этом мире. Поэтому знайте: вы должны быть выгодны фирме. Выгода может быть разной. Но, как правило, она имеет материальный аспект. И если вы получаете высокую зарплату, то просчитать вашу выгодность элементарно. Надо просто посмотреть, отрабатывается ли ваша зарплата, рентабельны ли вы. Если вы, скажем, получая \$1000 в месяц, приносите фирме доход на \$100 000 в месяц, будьте спокойны — вас никто пальцем не тронет. И вы можете требовать повышения зарплаты, улучшения условий работы труда, свободный график и все, что душе угодно. И вам это дадут. А вот если наоборот... Если ваша зарплата не окупается, то и гроза не за горами. И вас вполне могут уволить, каким бы хорошим человеком вы не были. Если таким способом вашу "выгодность" вам просчитать сложно, то можно попробовать другой. Также очень простой. Заключающийся в вопросе о том, сможет ли работодатель найти за те же деньги, которые он вам платит, человека, который будет либо выполнять больший, чем вы объем работы, либо тот же объем, что вы делаете, но за меньшие деньги. И если ответ положителен, то тоже ничего хорошего не ждите. Потому что тот товар, который вы предлагаете, покупать не выгодно. Значит, и покупать не будут. И забудьте как страшный сон то, что руководители этих двух способов проверки "рентабельности сотрудника" не знают. Знают прекрасно. Просто часто сотрудникам

просто дается время "на раскрутку", и от тех, кто это время использовал не по назначению, избавляются.

...В принципе, на этом в данной главе можно ставить точку. Потому что все самое главное сказано: если вы будете следовать только этому одному правилу, то и с социальной психологией, и с психологией управления у вас все будет нормально. Вы и в команде сработаетесь, и руководить командой сможете в случае возникновения такой потребности. Ну, а чтобы вам было легче следовать этому правилу, мы его раскроем чуть подробнее и еще поговорим на тему взаимоотношений в группе.

Какие бывают группы

Группы подразделяются по многим признакам. Мы из этих признаков рассмотрим самые интересные для нас.

Во-первых, группы бывают **формальные** и **неформальные**. Формальные группы можно определить как группы, созданные на основании каких-либо документов: это, к примеру, работники предприятия, школьники в конкретном классе, гости и т. д. Неформальная группа — это группа, сформированная на основании каких-либо личностных предпочтений: любви, дружбы, личных симпатий и т. д. Конечно, одна и та же группа может быть одновременно и формальной, и неформальной. Хорошая семья является самым простым примером такой группы: с одной стороны, эта группа формальная, так как есть документ, регламентирующий ее образование (свидетельство о браке), с другой стороны, эта группа также является и неформальной, так как члены группы относятся друг к другу с любовью и каждый из них вступил в эту группу добровольно.

Примечание

В неформальную группу всегда вступают только добровольно. В формальную вы можете вступать как добровольно (устраиваетесь работать на предприятие), так и недобровольно (список учеников конкретного класса).

Формальной группой управляет руководитель, а неформальной — лидер. Ясно, что власть лидера намного предпочтительнее, так как его власть абсолютна, а руководитель вынужден действовать в рамках тех или иных должностных инструкций.

Примечание

Дорогие руководители! Независимо от того, чем вы руководите, вы должны стремиться к тому, чтобы быть не только руководителем, но и лидером.

В своих работах известный отечественный психотерапевт М. Е. Литвак выделяет три типа неформальных групп:

- учебно-карьеристскую;
- культурно-развлекательную;
- алкогольно-сексуальную.

Учебно-карьеристские группы

В учебно-карьеристские группы входят те, кто стремится к профессиональному росту, продвижению по службе и вообще любому личностному совершенствованию. В вузах, к примеру, студенты, входящие в такую группу, сидят на лекциях в первых рядах и внимательно слушают лектора. Конечно, такие группы есть и на производстве. С ними и просто, и сложно. Дело в том, что для того, чтобы успешно работать в такой группе и успешно ею руководить, нужно постоянно самосовершенствоваться. У руководителя или члена группы, которые заботятся о своем личностном росте, никаких проблем при работе с такой группой не возникает. Наоборот: они находятся в своей тарелке и избавлены от постоянных "Ты меня уважаешь?!", которые по пять раз на дню можно слушать в других группах. А вот тем, кто в своем личностном росте остановился, работать с такой группой сложно.

Примечание

Именно на том, что руководитель и подчиненный входят в состав разных групп, основаны многие конфликты. Очень распространенный пример. Подчиненный, к примеру, аспирант, является карьеристом, стремится к личностному росту, а его зав. кафедрой в своем личностном росте уже давно остановился. (В этом случае, как мы увидим далее, он входит либо в культурно-развлекательную группу, либо в алкогольно-сексуальную). Конфликт в данном случае неизбежен. Поэтому при приходе на работу в учреждение обращайтесь основное внимание на то, в одной группе вы находитесь с вашим руководителем или в разных. Если в разных, то ничего хорошего не ждите — конфликт рано или поздно произойдет. И, если вы карьерист, а ваш руководитель "культурник", или хуже того, "алкоголик", то лучше уйдите из этого учреждения, так как развиваться вам там не дадут.

Культурно-развлекательные группы

Что собой представляют такие группы, думаю, неплохо понятно из названия. В нее входят те, кто остановился в своем личностном росте, но, набивши руку на рутинной и часто повторяющейся работе, неплохо ее выполняет. На предприятиях в эту группу часто входят те, кто имеет какие-либо интересы "на стороне". Причем интересы могут быть самыми разными: это

и свой бизнес, и ведение домашнего хозяйства, и вечерняя выпивка с друзьями... В принципе, если речь идет о членах такой группы, то особых хлопот с ними нет: свои обязанности они выполняют ровно, но "от и до". Задерживаться на работе сверхурочно их лучше не просить. И к какой-либо "карьеристской" работе, вроде освоения новых методов, привлекать их тоже не стоит.

А вот о руководителе-культурнике стоит поговорить особо. Прежде всего потому, что для тех, кто хочет делать что-то новое и интересное, этот тип руководителя вряд ли подходит. Во-первых, руководитель-культурник почти всегда автократ. Он никому не доверяет, самолично во все вмешивается. Никакой вопрос, даже иногда самый мелочный, никто, кроме него решить не может.

Примечание

Поэтому если видите, что во главе организации стоит "культурник", не тратьте время на решение вопросов с его заместителями: они ничего не решают.

А поскольку сам он не успевает следить за всеми делами, подчиненным самостоятельности не дает, то нерешенных дел становится все больше и больше. Помощники такого руководителя нередко являются просто "козлами отпущения". Ни в коем случае не донимайте руководителя-культурника новаторскими идеями. Единственный способ более-менее нормально с ним сосуществовать — это давать ему прокричаться и не донимать его никакими новшествами. Карьеристам с таким руководителем, понятное дело, живется несладко, и к ним все время придираются. Единственный способ карьеристу выжить под таким руководством — это начать расти на стороне, а в основной организации выполнять свою работу "от и до", и не приставать ни к кому ни с какими реформами. Приведу типичный пример. Научный сотрудник одного института являлся карьеристом, а все его руководители, и непосредственные, и прямые, культурниками. Сотрудник публиковал в год около 10—15 научных работ (продукция целой кафедры), и был, в принципе, на хорошем счету, но ни о каком развитии речь не шла. Поняв, что в данном заведении ничего не светит, он создал группу на стороне (стал "расти сбоку", выражаясь терминологией М. Литвака). Постепенно, все больше своего времени он отдавал работе в этой своей группе, которая была, конечно карьеристской. Но в силу высокой квалификации он и на кафедре работал весьма неплохо. Так, постепенно, он все больше выходил из основного места работы. Настал момент, когда он там практически перестал появляться, но поскольку продолжал по инерции писать статьи и делал это в силу профессионализма все так же хорошо, никто никаких замечаний ему не делал. И вот когда в родном вузе у него "осталась только пятка", он ее спокойно вырвал и ушел заниматься своим делом.

Примечание

Дорогие руководители! Еще раз повторюсь о том, что работники-культурники, в принципе, неплохие работники (чего не скажешь о руководителях-культурниках), и с ними можно долго и плодотворно работать. Единственное, надо разобратся в причинах того, почему они являются культурниками. Дело в том, что человек хоть где-нибудь стремиться быть первым. И если на вашем предприятии, он не входит в карьеристскую группу, значит, он развивает "свое первенство" где-то в другом месте. Вот этот момент и интересен. Если этим местом является семья, то можете за работу с этим сотрудником быть спокойны. А вот если это другая организация, то рано или поздно он в нее перейдет.

Примечание

Дорогие сотрудники! Научитесь разбираться в своих руководителях, и если вы карьерист, а ваш руководитель — культурник, то даже и не пытайтесь с его помощью решать свои проблемы, а "растите сбоку".

Алкогольно-сексуальная группа

О том, что представляет собой такая группа, тоже понятно из названия. Также понятно и то, что это наиболее неудобная группа в любом коллективе. Заметим, что когда мы говорим "алкогольно-сексуальная", речь идет не об алкоголиках и сексуальных маньяках в прямом смысле этого слова, а о стиле поведения людей, входящих в эту группу.

Если вы руководитель, и на вашем предприятии есть такие группы, то разбивать их нужно быстро и жестко. Попросту говоря, пьяниц надо увольнять, и чем раньше, тем лучше. Никакие переговоры и уговоры ни к чему не приведут. Если же вы подчиненный-карьерист, то как можно быстрее выходите из такой группы, если не хотите собственной деградации. Потому что уклониться от выпивки в такой группе невозможно, а материальные и иные блага в такой группе распределяются только среди тех, кто принимает наиболее деятельное участие в "процессе". Бегите из таких групп быстрее, что бы вам не обещали, и ни о чем не жалейте: вам могут обещать не мало, так как, если начальник-алкоголик не совсем "спился", он понимает, что пока он отдыхает, кто-то должен работать. Но, повторюсь, уходите, и не слушайте, какие бы блестящие перспективы не рисовались: нет у такой группы никаких хороших перспектив. Это — закон. Единственное исключение, это когда создается такая ситуация, при которой руководитель является единственным алкоголиком в возглавляемой им формальной группе, а большинство группы составляют карьеристы. В этом случае они имеют возможность развиваться "сами по себе", так как руководителю не до них: он просто запирается в своем кабинете и тихо делает свое алкогольное дело.

О референтных группах

В социальной психологии есть очень важное понятие так называемой **референтной группы**. Референтная группа — это эталонная группа. То есть это такая группа, которая является эталоном для личности и в которую он стремится попасть. К примеру, для какого-либо мелкого чиновника референтной группой может являться, скажем, правительство Российской Федерации, а для члена правительства референтной группой может быть ближайшее окружение Президента. Так вот, если человек ту группу, в которой он находится, не считает референтной, он будет пытаться из нее уйти и, конечно, ничего не будет делать для укрепления этой группы. Работник в нереферентной для него группе никогда не будет работать с полной отдачей. Скорее, он, наоборот, будет вредить этой группе каким-либо образом.

Сотрудникам

Дорогие сотрудники! Стремитесь найти в учреждении референтную группу и попасть вам надо именно в нее. Разумеется, этот призыв относится только к сотрудникам-карьеристам. Если же вы культурник, то вам в такой группе может быть сложно.

Руководителям

Уважаемые руководители! Всем знакома ситуация, когда некоторые работники уходят с предприятия, казалось бы, без видимых причин. А причина может быть в том, что они считают группу, в которой они работают, нереферентной. Поэтому из всех сил стремитесь к тому, чтобы группа, которой вы руководите, была референтной, чтобы работать в ней было престижно. Это избавит вас от очень многих проблем, таких как текучка кадров (в референтных группах она минимальна и уходят, как правило, те, кто не в состоянии "выдержать темп"), прием на работу высококвалифицированных специалистов (любой хороший специалист стремится попасть в референтную группу) и т. д.

Приведем некоторые примеры референтных групп.

К примеру, в мире медицины некоторые выдающиеся главные врачи сумели создать из своих больниц и отделений референтные группы, в которые сотрудники стремились попасть на работу, а больные — на лечение. Это всем известные офтальмологический центр С. Н. Федорова, больница выдающегося врача-травматолога Г. А. Илизарова, Институт сердечно-сосудистой хирургии Н. М. Амосова и т. д.

Многие военные, выбравшие своей специальностью службу в войсках специального назначения, мечтают попасть служить в группы "Альфа" и "Вымпел", так как эти группы являются для них референтными.

А для многих Web-дизайнеров, к примеру, референтной группой является студия Артемия Лебедева.

Вывод

Таким образом, приходя на новое место работы, вы должны стремиться попасть в учебно-карьеристскую группу, и совсем хорошо, если она еще будет и референтной (что бывает очень часто). И вообще, когда устраиваетесь на новое место работы, старайтесь как можно лучше оглядеться, чтобы не попасть впросак. И для начала попытайтесь "поставить диагноз" всему учреждению в целом. Я не раз видел учреждения, которые изнутри целиком и полностью были алкогольно-сексуальными. Хотя на начальном этапе их "алкоголизации" с внешней стороны они производили очень благоприятное впечатление.

Групповые процессы

Групповой процесс — это налаживание отношений между членами группы или взаимодействие между различными группами. Очень хорошо видно, как протекают групповые процессы и формирование групп на лекциях в институтах. Первые ряды обычно занимают "карьеристы", которые на лекции занимаются только тем, чем на ней и надо заниматься. Представители культурно-развлекательной группы сидят где-то посередине аудитории. Они, в общем-то, и лекцию записывают, и поболтать успевают. И, наконец, на задних рядах располагаются члены алкогольно-сексуальной группы, которые ничего не записывают, а просто "отбывают время": болтают, играют в карты, иногда выпивают. На границах групп неизбежны конфликты: так, к примеру, "культурники" могут мешать своей болтовней слушать лекцию сидящим рядом "карьеристам".

Конфликт, не перешедший границу группы, называется **интригой**. Об интригах руководство узнает от **осведомителей**. Сразу замечу, что осведомители — люди не очень умные по определению, потому что осведомителями они становятся в силу своей профессиональной некомпетентности. Ну, нечего им больше делать, вот они и интригуют. Профессионал никогда не будет заниматься интригами хотя бы потому, что у него на это просто нет времени. Осведомителей при приеме на работу старайтесь выявить в первую очередь, и вести себя с ними очень аккуратно. Правда, из того же определения следует, что начальство, которое пользуется услугами осведомителей, тоже не очень умное, потому что умный человек не станет прислушиваться к мнению человека неумного (а осведомители, как мы выяснили, умом особо не блещут). Осведомители есть практически во всех организациях, где руководство смотрит сквозь пальцы на такие вещи, и практически в каждой большой организации. К новичкам, и это надо знать, осведомители относятся наиболее "трепетно": потому что только что устроившийся на работу человек для них то же самое, что новая игрушка для ребенка. Но еще раз повторюсь, умный начальник услугами осведомителей пользоваться не будет: он лучше сам поговорит с людьми и сам составит свое мнение о происходящем.

Примечание

Если вы начальник и хотите нейтрализовать осведомителя, то сделать это очень просто: надо только устроить ему очную ставку с тем, про кого он говорит гадости (хороших вещей осведомители не сообщают). Этим самым вы также проведете и профилактику появления других осведомителей в вашей организации.

Об антилидерстве

Если интриги направлены на лидера группы, то это уже называется агитацией, а людей, которые этим занимаются, называют **агитаторами** или **антилидерами**. О них мы сейчас и поговорим, так как то, что представляют собой агитаторы и как с ними бороться, стоит знать, если вы собираетесь в недалеком будущем стать руководителем, причем неважно какого масштаба: будете ли вы менеджером проекта или директором отдела, с антилидерами вы, скорее всего, столкнетесь. И здесь очень важно действовать "по науке": потому что только так можно нейтрализовать антилидера.

Примечание

Если вы сотрудник, то никогда не входите в группировку антилидера. Ничем хорошим это не заканчивается: лучше заниматься своим профессиональным ростом, чем участвовать в чужих революциях. А входить в антилидерскую группировку, даже если вы в принципе согласны с мнением антилидера, не стоит хотя бы потому, что антилидеры, как правило, глубоко закомплексованные люди с серьезным комплексом неполноценности, у которых не хватило каких-то личностных качеств для того, чтобы стать настоящим лидером. Поэтому антилидеры, как правило, своих целей не добиваются, а если и добиваются, то от этого всем становится только хуже.

Первое и основное правило борьбы с антилидером заключается в том, что с ним ни в коем случае ...нельзя бороться. Ему надо только уступать. Дело в том, что антилидер — это всегда ваш подчиненный. И, борясь с ним, вы проигрываете в любом случае: даже если он и проигрывает схватку с руководителем, то все равно получает психологическую компенсацию, так как приобретает в своих собственных глазах (да и в чужих нередко) репутацию смелого и принципиального человека. Вы же, как руководитель, проигрываете даже, если вы победили, потому что все только и подумают: тут мол, нет ничего удивительного, он же начальник, он же сильнее! А если вдруг вы потерпите еще и поражение, то авторитет ваш будет стремительно уменьшаться. Коллективу же, особенно состоящему из культурников, борьба руководителя с антилидером тоже выгодна: "под шумок" можно заняться своими делами. А тех карьеристов, которые есть в вашей группе, вы во время борьбы запросто можете потерять, потому что они пришли работать и

лично расти, а не наблюдать чемпионат отдельно взятой фирмы по борьбе. Поэтому, пока вы боретесь с антилидером, группа, как правило, разваливается.

Примечание

Один научный руководитель большую часть своего времени посвящал борьбе с антилерами. Тем же карьеристам, которые у него работали, надоело бесконечно наблюдать эти схватки, к участию в которых их еще и пытались привлечь, и они ушли из научной группы. И группа попросту развалилась. Вернее, формально она осталась существовать, но ничего стоящего уже не производила. Хотя поначалу это была очень сильная группа с большим "процентным содержанием" карьеристов.

Другой, более масштабный пример, приводит М. Литвак. Пример Горбачева и Ельцина. Ясно, что Ельцин был антилидером. А у М. С. Горбачева в окружении просто не было высококвалифицированных психологов. А будь они, кто знает, как бы все сложилось... Ведь Горбачеву для того, чтобы победить Ельцина, надо было ему просто ...уступить.

— Программа 500 дней? Пожалуйста! (Тем более что ждать не долго).

Провалилась программа — долой Ельцина! Не провалилась — ура Горбачеву!

Примечание

Из этого примера очень хорошо видно, что законы психологии групп одинаковы, что для группы из 10 человек, что для целого государства. Поэтому тот, кто научится **психологически грамотно** управлять десятью, сможет управлять хоть целым государством. Это действительно так. Обратное тоже верно: если не получается "в малом", "в большом" тоже не получится. К сожалению, примеры непонимания этого можно видеть очень часто: в мэры городов приходят люди, не сумевшие справиться даже с коллективом фирмы из трех человек. Причем они искренне убеждены в том, что с большим коллективом они справятся лучше, так как на более высокой должности у них будет больше помощников и больше финансовых средств. А дело-то не в средствах и не в помощниках... Дело в неумении управлять и в незнании элементарных психологических основ. Что в конце концов и становится видно. Ну, не может человек, не умевший договариваться даже с "мелкими" клиентами о мелкооптовых делах, вдруг начать договариваться с "большими людьми" о крупнооптовых поставках и больших инвестициях.

Поэтому не боритесь с агитатором, а по возможности обращайтесь на него как можно меньше внимания. И пусть это не тревожит ваше самолюбие: вы наносите ему рану похлеще, чем если бы в запале исхлестали его железными прутьями. Потому что самое страшное наказание для скандалиста — это когда его лишили скандала.

Примечание

Существует интересная (и правдивая) теория, согласно которой скандал — это суррогатный заменитель ...секса. Объяснение этого положения выходит за рамки книги, если кому интересно, я могу рассказать об этом подробнее на нашем форуме. А пока просто запомните это положение. Так вот, когда вы лишаете скандалиста скандала, не поддавшись на его провокации, он чувствует примерно то же самое, что чувствует изготовившийся к прекрасной ночи муж, которому жена вдруг неожиданно дала "от ворот поворот". Кстати, особо злостным скандалистам можете рассказать об этой теории (желательно при большом скоплении народа) и попросить "не заниматься прилюдно сексом в извращенной форме". Или попросить прекратить по отношению к вам сексуальные домогательства (предварительно, конечно, объяснив аудитории, почему вы так говорите). Действует неплохо — проверено на практике. Даже если кто-то и решается потом повторить скандал, те, кто слышал ваше выступление, начинают похихикивать, и скандалист утихает.

Еще один прием борьбы с антилидером — это дать ему повышение. К примеру, попросить его замещать вас во время вашего отпуска. Антилидеры, как правило, очень плохие психологи, и за время вашего отпуска они, прикоснувшись к власти, могут так всем насолить, что к моменту вашего возвращения сторонников у вашего "противника" заметно поубавится. Это тоже весьма действенный прием, который многократно применялся. Дело в том, что антилидеры не знакомы с пословицей "легка гиря в чужих руках" и считают, что то, что делаете вы по части управления, они смогут делать ничуть не хуже, причем сразу. Так дайте им такую возможность!

Примечание

Кстати, нередко антилидер действует неосознанно, на сознательном уровне ничего не имея против руководителя.

Теперь поговорим еще об одном интересном персонаже антилидерской группировки, который М. Литвак называет "Цепной Собакой", а мне больше по душе название Промокашка — так звали героя фильма "Место встречи изменить нельзя", который подходил "проверять на вшивость" Шарапова, внедрившегося в преступную группу. Промокашка — это член антилидерской группы, который первым подходит ко вновь прибывшим и проверяет их "на вшивость". Поскольку Промокашки как правило "лают", Литвак и обозвал их Цепными Собаками. А цепные они из-за того, что не имеют реальной власти, хотя и очень к ней стремятся. Особенно хорошо роль Промокашек заметна в армии и в различных уголовных группировках. В армии они, конечно, первыми подходят к новобранцам и всячески их провоцируют. И выдержать эту первую проверку очень важно, от этого может зависеть вся дальнейшая служба. Но у нас задачи скромнее, и нас при устройстве на работу, даже если что и не так сделаем, по крайней мере, бить не будут. Тем более что и задача у нас обратная той, которая стоит перед новобранцами:

им нужно стать "своим" в антилидерской армейской группе, потому что жить они будут среди солдат, а не среди офицеров, а все солдаты в армии это есть одна большая антилидерская группировка по отношению к офицерам. А нам при устройстве на работу нужно, наоборот, бежать от антилидерской группы как можно дальше. Поэтому с Промокашками держите себя твердо, можно чуть насмешливо. Промокашки не всегда лают, иногда они подлизываются ко вновь прибывшим членам группы, пытаясь вовлечь их в свою группировку. Как правило, они играют при этом в игру "Кто есть кто", ласково беседуя с новичком и вводя его в курс дела (кто есть кто в этой конторе). На это тоже не поддавайтесь — тот, кто вас первым обласкал, первым же потом и укусит.

Как найти свое место в группе

Так получается, что, как правило, когда мы приходим в новую группу, у нас в ней поначалу не главная роль. Значит, встает вопрос об адаптации, и чем менее болезненной она будет, тем лучше. В предыдущих разделах мы, в принципе, рассмотрели все основные вопросы, влияющие на адаптацию в группе, поэтому сейчас мы лишь конспективно все повторим.

Итак, мы пришли на новое для нас место. Что делать? Делать нужно всегда одно и то же, независимо от того, в какую группу вы попали: в первые дни нужно сосредоточиться только на работе и проявить себя как специалиста с лучшей стороны. Во всех остальных процессах нужно не принимать никакого участия. Если вы сразу попадете в карьеристскую группу, то можете считать, что вам повезло: максимум, что вам грозит, это некоторая настороженность. А некоторые члены этой группы вас могут вообще не заметить и не замечать еще несколько недель. Ну, некогда им. А вот если попадете к "культурникам" или "алкоголикам", то вас начнут "прошупывать". Отвечать им не надо, а вот послушать стоит. В принципе они могут дать вам неплохую информацию на предмет того, что происходит в организации. Только не присоединяйтесь ни к какой группировке. Вообще, присоединяться к какой-либо группировке, кроме группировки вашего руководителя, дело чреватое. Особенно для частных фирм, где руководитель, как правило, еще и полноправный хозяин этой фирмы. А лучше поначалу вообще ни к каким группировкам не присоединяться. И, повторюсь, избегайте любых интриг, даже тех, которые кажутся безобидными. Потому что интрига живет по своим законам, и если не вы сами ее организовали, то разобраться в ее конечных целях — дело трудное. Особенно это относится к новому месту работы: потому что, чтобы понять кто с кем и против кого дружит, надо минимум пару-тройку месяцев отработать. Точно также не стремитесь с места в карьер принимать чью-то дружбу: вначале разберитесь, с кем стоит дружить, а с кем нет. А пока разбираетесь, будьте со всеми одинаково вежливыми, пре-

дупредительными и отстраненными. А если вас начинают активно вовлекать в свои междоусобные войны, то лучше прикинуться полным идиотом во всех этих интриганских делах и на всех углах рассказывать, что вы во всех этих интригах разбираетесь как пингвин в коллоидной химии. Да, ваш рейтинг в глазах окружающих может несколько снизиться, но иногда лучше быть немного глупым, чем много битым. Но это не значит, что вы не должны вообще никак участвовать во всех этих групповых процессах. Должны, и еще как должны. Только не языком, а глазами и ушами. То есть вы должны присматриваться и прислушиваться, так как ваша задача сейчас это понять, что из себя представляет группа в целом и каждый ее член в частности. И определить цели, перед ними стоящие. Для того чтобы понять с кем вам по пути, а от кого лучше побыстрее свернуть в сторону. При всем при том вы не должны забывать и о руководителе, которому из ваших действий должно быть понятно, что вы работаете только на процветание фирмы и, следовательно, на процветание руководителя. Поэтому избегайте интриг, налаживайте отношения с руководителем и старайтесь не входить ни в какие группировки. В этом случае не исключено, что когда руководитель пойдет на заслуженный отдых или на повышение, то он оставит вместо себя именно вас. Во-первых, потому, что вы неплохо работали на его благо, а во-вторых, вы можете руководить коллективом в целом, так как не входите ни в одну из его группировок.

А теперь в заключение проведем небольшой практический тренинг. Допустим, вы попали во вновь созданную группу, в которую вместе с вами попали еще несколько человек. И вы хотите в этой группе стать лидером.

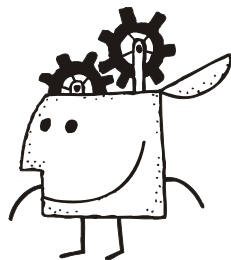
Примечание

Причем то, какая именно это группа, особого значения не имеет. Это может быть и вновь создаваемый отдел, и туристическая группа, и новобранцы в казарме: групповой процесс везде протекает примерно одинаково. Условия задачи таковы, что все в группе друг друга не знают или малознакомы, а вам надо стать лидером в этой группе.

Вначале при формировании группы всегда идет обмен ритуалами. Ритуалы могут быть самыми разными: и рукопожатия, и косые взгляды. Неважно. На этой стадии никогда ни в одной группе ничего важного не происходит. Затем начинается стадия развлечений: у представителей алкогольно-сексуальной группы это выпивка и сопутствующие ей вещи, а представители культурно-развлекательной группы играют в игру "А вот, когда я был (в Париже, Ницце, на экскурсии в центре управления полетами и т. д.)". Вам нужно молчать и делать свое дело. Такой подход имеет два плюса: во-первых, не попадете не в ту группировку, во-вторых, вас заметят (потому что вы выбиваетесь из общей картины), хоть даже и не подадут виду. На стадии развлечений обычно появляется какой-то лидер, но это псевдолидер, и начинаются выборы "актива группы" — тех, кто ближе к лидеру. Здесь тоже нужно

отмалчиваться и внимательно наблюдать за тем, что происходит. Не рвитесь сразу в "первачи" — это делают голодные до власти люди, а вам это делать нет никакой необходимости: пускай все знают, что вы "не голодны", этим вы, кстати, вложите в подсознание членов группы информацию о том, что вы больше похожи на истинного лидера, чем тот, кто сейчас лидирует. А этому псевдолидеру дайте возможность сделать свои ошибки. В актив группы, конечно же, тоже не стоит идти: потому что в активе никогда нет лидеров, там всегда находятся Промокашки. Где-то через неделю-две начнут формироваться неформальные группы, в которых появятся свои лидеры. Как правило, эти первые неформальные группы в большинстве своем состоят из не очень компетентных людей. Причина этому то, что некомпетентные люди всегда, пусть даже и в глубине души, осознают свою некомпетентность, и у них возникает подсознательное стремление "найти защиту", т. е. чем-то заглушить свою некомпетентность. Часть людей в такие группы не войдет, — на них следует обратить особое внимание, скорее всего они профессионалы своего дела и им тоже нет смысла участвовать в начальных групповых процессах: они тоже, как и вы определяются "кто есть кто", и не спешат зря тратить свое время, вступая в ненужные им группировки. Постепенно из этих групп выйдут и другие умные люди, которые попали туда по недоразумению. И вот тут в группе, скорее всего, начнутся раздоры: потому что актив из Промокашек не сможет должным образом организовать групповой процесс, а умные люди из актива либо вышли, либо изначально там не присутствовали. Вот тут вам и надо браться за управление. Для этого надо определить те задачи, в решении которых вы находитесь на вершине компетентности, и сделать выполнение этих задач основными целями группы. И лидерство к вам перейдет автоматически.

Глава 4



Взаимодействие с заказчиком

Часть работы программиста, касающаяся взаимодействия с заказчиком, крайне важна, потому что направлена непосредственно на оплату его труда. Независимо от того, работаете ли вы частным образом, либо в организации, от результатов ваших переговоров с заказчиком зависит ваша зарплата.

По большому счету, дело даже не в зарплате, а в самореализации. С этой точки зрения талантливый программист ничем не отличается от талантливого певца (гитариста, пианиста и т. д.). Певцу для того, чтобы "пробиться", так или иначе нужно взаимодействовать с шоу-бизнесом. Иначе о нем с большой вероятностью никто не узнает, даже если у него великолепные голосовые данные. К сожалению, во всех наших школах (высших, музыкальных и др.) практически не преподают практическую психологию, менеджмент и т. д., а все внимание уделяют только технике (игры, программирования, выполнения различных работ). В результате на эстраде вместо талантливых исполнителей, которых приятно слушать, мы по большей части имеем разновозрастных дебилов, не знающих азов нотной грамоты, слушать которых можно, только будучи в сильно нетрезвом виде. А нормальные люди занимаются чем угодно, кроме того, чем должны. В программировании то же самое. Я знаю не одного прекрасного программиста, которые программируют только вечерами. Потому что днем они работают стропальщиками, каменщиками, слесарями. И часто годам к 30 спиваются.

Зададимся вопросом: почему же так происходит? Почему талантливые люди, которых в России как продавцов на центральном рынке, зачастую бедствуют, занимаются не тем, чем должны, а всякая, в прямом смысле слова, малообразованная шушера наводнила Олимп?

Примечание

Вы можете возразить, что в программировании не так. Что тут, чтобы чего-то добиться, все-таки нужно обладать какими-то техническими умениями. Увы, и здесь все точно так же: каждый третий наш проект это доделка или переделка того, что уже было начато другими "коллегами", причем сполна получившими деньги за сделанную работу.

Итак, почему? В том числе и потому, что встречаются и в прямом, и в переносном смысле по одежке. Почти всегда. По походке, по разговору, по умению отвечать на вопросы.

Встречают по одежке

В прямом и в переносном смысле. Сначала о самом прямом. Подойдете ли вы в электричке к бомжу, если даже точно будете знать, что он выдающийся математик? Вряд ли. Хотя, между прочим, среди бомжей есть очень образованные люди. Я точно знаю, что на маршруте Н. Новгород — Вязники есть бомж, бывший в своей прошлой, небомжовой жизни аж кандидатом физико-математических наук.

Примечание

Этот пример, кстати, лишний раз говорит еще о двух серьезных вещах. Во-первых, о том, что даже очень высшее образование не гарантирует ничего. Это всего лишь одна из ступеней, которую стоит одолеть. Фундамент, после закладки которого постройка дома только начинается. Не более. Во-вторых, о том, что стать бомжом в прямом смысле слова намного легче, чем многие из нас думают. Нужно только позволить себе чуток расслабиться...

Теперь представьте, что к заказчику пришел этот бомж-математик. Ни ученая степень, ни что иное ему не поможет. Конечно, я утрирую, но в частности некоторые из нас, бывает, иногда смахивают на этого бомжа. Я, к примеру, куря в машине по дороге на деловую встречу, имею устойчивую привычку в задумчивости обсыпать себя пеплом с ног до головы. Один мой знакомый нередко при переговорах, задумавшись, переходит на свист. Другой никак не отвыкнет от привычки хрустеть костяшками пальцев в момент принятия судьбоносного решения. Это, так сказать, примеры внешних проявлений и далеко не самых худших. А сколько внутренних? Особенно хорошо они заметны на собеседованиях, о которых мы говорили в *главе 2*. Беседа с потенциальным клиентом от собеседования практически ничем не отличается.

Когда мы говорим "встречают по одежке", имеется в виду не только прямой смысл этой фразы, но и то, что для того, чтобы достичь осязаемых успехов, мало только технических знаний. Нужно еще уметь "себя подать".

О внешнем имидже

Выкиньте те книжки, в которых говорится, что клавиатура должна быть в печенье, а одеты вы можете быть как угодно, лишь бы программа работала. Лучше пускай программа не работает, а одеты вы должны быть "с иголочкой". Потому что ваш приятный внешний вид значит очень много для ус-

пешных продаж. Он должен подчеркивать вашу успешность. Я не имею в виду, что вы должны обязательно быть в костюме от Бриони, галстук от Версаче и ботинках от Валентино. Хотя это было бы и не плохо. Просто нельзя одеваться так, как будто вы только что вылезли из сарая, в котором стоит ваш компьютер. Даже если так оно и есть. Вернее, тем более, если так оно и есть. В этом случае ваш имидж должен говорить о том, что вы вышли минимум из собственного двухэтажного особняка. А то, что брюки слегка уличной грязью запачкали, так это из-за того, что пришлось 50 метров по улице пройти, так как ваш 15-метровый джип не смог въехать на автостоянку перед офисом клиента. Повторюсь: особенно безукоризненность имиджа важна на начальном этапе. Потом, когда станете всемирно известным "крутым эксклюзивщиком", можете приезжать к клиентам хоть в одеяле и тапочках на босу ногу. Спешут на экстравагантность гения. А пока до всемирной известности еще далеко, помните, что небрежность и неаккуратность в одежде могут эту самую известность отодвинуть на неопределенный срок.

Не пренебрегайте одеколонами. Дело здесь совершенно не в том, что кто-то там пахнет чем-то очень неприятным. Совершенно не обязательно. Просто сексопатологи вкупе с психологами, исследуя, что люди больше всего не терпят в других людях, выяснили, что непереносимость по запаху — одна из самых страшных непереносимостей. То есть вы можете в человеке вынести много неприятных моментов, но если он не так для вас пахнет, то о совместной жизни можно даже и не думать. И хотя нас с клиентами никто не заставляет озаботиться вопросами совместного проживания, учитывать этот момент необходимо. Поэтому свой запах, каким бы он приятным вам не казался, и сколько бы раз в день вы не мылись, желательно нейтрализовать. Одеколоном. Один мой знакомый таксист добился того, что стал одним из самых высокооплачиваемых таксистов в городе одной простой вещью: несколько раз в день он одеколонил салон своей машины. Очень приятным и по этой причине не самым дешевым одеколоном. А еще он на время пути включал приятную музыку. Не стоит удивляться тому, что большинство клиентов из его машины вылезают, словно загипнотизированные кролики. И тут же просят его визитку.

Примечание

А теперь антипример. По роду своей прошлой деятельности мне часто пришлось бывать на международных научных конференциях. Заметил такую деталь. Все западные ученые всегда одеты очень опрятно. Независимо от своего научного статуса. На наших же профессоров (и даже академиков) в сравнении с их западными коллегами порой просто стыдно смотреть. К примеру, для нашего профессора ничего не стоит прийти на доклад в спортивных штанах и тапочках на босу ногу. Это не авторская метафора, а реальный случай. Западные коллеги, разговор которых я один раз ненароком подслушал, совершенно справедливо презрительно называли наших выдающихся ученых "ragged fellow". Что в переводе с английского означает "оборвыш".

Встреча начинается задолго до встречи

Часто ведь бывает: вроде и придаться не к чему, и понравилось на словах все, а — отказ. Совершенно непонятно почему. Ну, не захотелось ему с вами работать и все тут. Спроси его — он, возможно, и сам толком не объяснит почему. А дело может быть в том, что... В том, что любая наша встреча с человеком начинается задолго до того, как вы начали обсуждать конкретные вопросы. И даже до того, как обменялись словами приветствия. Вот вы входите в кабинет к заказчику. Даже если он не ставит своей целью по вашей походке прикинуть ваш психотип, на подсознательном уровне такие "мелочи" фиксируются. Как вы вошли (уверенно или заползли в кабинет, словно улитка), каков ваш внешний вид, как руку пожали, как сели и т. д. Это называется невербальным общением (подробнее об этом см. ниже в этой главе). Сели вы, предположим, на краешек стула, глазами пол буравите, а потом удивляетесь, что заказчик вместо обсуждения деловых вопросов прочитал вам отеческую лекцию. И правильно. Потому что, после того, как вы так сели, его подсознание автоматически перевело вас из деловых партнеров в разряд студентов-троечников, пришедших на экзамен, или провинившихся сотрудников. И так далее — масса вариаций. Суть в том, что если вы объективно видите, что вы ничем не хуже других по части предлагаемого продукта, а вам систематически ничего не заказывают — одной из причин может быть ваша "плохая невербальщина", извиняюсь за такой вольный термин. Последите внимательно за собой: как вы одеты, как ходите, как говорите, как жестикулируете...

Не думайте плохо о заказчике

Очень важное правило. Если вы о своих клиентах думаете плохо, презираете их, вам ничего не светит. Лучше вообще не заниматься переговорами в частности и бизнесом вообще. Потому что, как бы вы не скрывали свое отношение, невербальные реакции вас выдадут. Уследить за всем не удастся, даже если вы неплохо собой владеете. Если только вы, конечно, не разведчик-нелегал по совместительству, которого не один год натаскивали на мастерство перевоплощения, причем так, чтобы никакая "невербальщина не отзеркаливала". Кстати, один из приемов состоит в том, чтобы полюбить своего партнера по общению. Не в прямом смысле слова, конечно, а в том смысле, что проникнуться к нему искренней симпатией. Хотя бы только на время встречи.

Примечание

В дальнейшем в этом разделе вместо громоздкого и неискреннего оборота "проникнуться искренней симпатией" я буду использовать глагол "любить" как более искренний и короткий.

Должен сразу сказать, что сделать это не просто, но зато, если этим приемом овладеете, за свои невербальные реакции можете быть спокойны. Никакой детектор лжи вас не раскусит. Вопрос в том, как полюбить? Особенно если он никакой симпатии не вызывает. Для начала надо понять, что огромным барьером на пути к тому, чтобы кого-либо полюбить (причем не обязательно заказчика), является наше самомнение. Безусловно, очень хорошее. О нем мы дальше будем говорить еще много в разных вариациях и с многочисленными примерами, а сейчас, чтоб не уйти в сторону, скажу лишь то, что для выполнения этого приема самомнения требуется ровно столько же, сколько соли в супе. Так, чуть-чуть для вкуса. А если переборщить, можно блюдо безнадежно испортить. То есть с этим определились: самомнение у нас сейчас абсолютно в норме: если что не по-нашему, в "ежиков" играть не будем. А если так, то можно считать, что больше полдела сделано. Дальше — это уже вопрос индивидуальной техники. Дальше надо у себя вызывать положительные чувства к партнеру. Вспомнив, к примеру, про какой-то хороший поступок из его биографии, если вы с ней хоть чуть-чуть знакомы. А если не знакомы, то просто представить, что этот поступок в его биографии был. Сидите вы, к примеру, перед клиентом и представляете, как он когда-то в прошлом бросился с высокого обрыва в речку спасать тонущего человека. Если все это реально представлять, у вас такие положительные невербальные реакции будут! Даже всегда натянутая улыбка станет искренней и доброжелательной.

Примечание

Кстати, этот прием можно использовать и как антиприем. Если хотите, чтобы у кого-то встреча сорвалась, расскажите перед встречей тому, кто будет встречаться, что-то плохое о том, с кем будут встречаться. Даже если человек неплохо собой владеет, процент того, что невербальные реакции его выдадут, далеко не нулевой. Излюбленный прием профессиональных интриганов. А если хотите быть уверенным в успехе, то рассказывать плохое надо перед самой встречей. В этом случае человек, скорее всего, не успеет обдумать ваше сообщение и определиться со своей реакцией на него, и во время встречи будет подсознательно искать доказательство/опровержение ваших слов. Что успеху переговоров обычно не способствует. Я это, конечно, не для того рассказал, чтобы вы кому-то переговоры срывали, а для того, чтобы вы умели распознавать подобные попытки со стороны ваших "доброжелателей".

Итак, еще раз повторяю: думайте о своих клиентах хорошо. И о бизнесе своем. И о поездке предстоящей. И тогда все получится.

Анекдот в тему

И почему меня никто не любит и никто со мной общаться не хочет? Чтоб вы все сдохли...

А вот если поступать как персонаж этого коротенького анекдота, то советую вам запастись терпением и ничему не удивляться.

Заказчик дурак?

Заказчик дурак? Конечно, дурак. Такой же, как и мы с вами в той деятельности, которой занимается он. Однако самая популярная тема для разговоров у дизайнеров и программистов, — "Какой заказчик кретин, не знает что такое домен". "Ой, что ты, — вторит ему другой, — вчера работал с таким идиотом, который не знает даже, что такое IP-адрес!" А с какой стати он должен это знать? Увы, такая нехорошая черта характера, как профессиональный снобизм, сильно развита у многих программистов. Но наша профессия ничем не лучше и ничем не хуже других — просто одна из многих профессий. И глупо требовать от человека другой профессии (заказчика), чтобы он в ней разбирался. Представьте себе, что вы пришли на прием к платному врачу, он вам дает кардиограмму, в расшифровке которой написано, что у вас нарушение проводимости по пучку Гиса. На ваш вопрос, что это означает, врач смотрит на вас взглядом, в котором смешано презрение к вашим умственным способностям с восхищением собственными. После этого он выходит в коридор и начинает жаловаться своему коллеге: "Слушай, ну у меня что не пациент, то дебил! Представляешь, какой попался — не знает, что такое пучок Гиса! А?! Ну, как с такими можно работать?"

Почему они такие?

Заказчики, имеется в виду. И так им не так, и эдак не эдак. Я вам скажу. Из-за чувства собственной значимости. Как и мы. Ведь все относительно. Сегодня вы исполнитель, и ублажаете клиента всеми доступными вам способами, а завтра вы уже заказчик. В магазине, к примеру, когда компьютер выбираете. Или принимаете работу сторонней организации, которая вам кондиционеры устанавливает или рекламный ролик монтирует. Что-то мне подсказывает, что вы и к тому придеретесь, и к этому, и к пятому, и к десятому. И если в тот момент, когда вы придиричливо выбираете вещь, принимаете работу сторонней организации и т. д., вам удастся взглянуть на себя со стороны, я думаю, вам станет весело. Потому что, совершенно забыв, как вчера сами уговаривали клиента, вы сегодня поменялись с ним местами, и теперь уговаривают уже вас. А вы произносите примерно те же самые слова и выражения, которые говорил вам вчера клиент. Понимаете? И клиент также. Он, продавая вчера, скажем, автомобили, такого наслушался про их ходовые качества и внешний вид! Сегодня, когда деньги платит он, тоже хочет чего-то такое сказать. "Да я вам такие деньги плачу, а вы мне тут...". Понятно, в общем. Поэтому если вы видите, что клиент не в духе и обругал все, что вы сделали, хотя вы объективно видите, что все нормально, не пе-

речьте ему. Скорее всего, кто-то просто недавно обругал продукцию, которую он продает. Или просто его кто-то разозлил перед вами. В этом случае ни в коем случае не спорьте и не доказывайте, что вы правы. Спокойным тоном спросите клиента, что конкретно ему не понравилось. Попросите перечислить. А сами вытащите ручку и записывайте. Все, что он перечислит, каким бы абсурдом вам это не казалось.

После того как записали, говорите стандартные фразы ("Все учтем, все поправим") и откланивайтесь. А до следующей встречи можете ничего не менять (если, конечно, в замечаниях клиента не было конструктивных моментов. Скорее всего, когда он взглянет на вашу работу другими глазами, ему все понравится. Особенно если вы поменяете несколько незначительных вещей и свою следующую встречу начнете с перечисления того, что вы сделали. "Вот тут мы поправили, что вы говорили. Эти ваши замечания мы тоже учли..." И т. д. Даже если вы правили не то, что говорил клиент, а что-то по своему усмотрению, все равно представляйте все так, будто он сделал эти замечания в прошлый раз, а вы их исправили. И обязательно акцентируйте на этом внимание в речи: "Как вы говорили, как вы предложили..."

Рожденный ползать — уйди с взлетной полосы!

По отношению к заказчику у исполнителей крайне нередко фраза "Я ему кучу писем послал, столько его ждал, теперь он пусть меня подождет". Такому исполнителю можно только повторить фразу, которой назван этот подраздел. Потому что никто нас ждать не будет — Web-студий сейчас на рынке как песка в пустыне. И заказчик, когда платит деньги за сайт, рассчитывает на то, что исполнитель понимает, что на данный момент сайт для него в лучшем случае на 10 месте среди предметов второй необходимости. Нет ничего плохого в том, чтобы ему напомнить о себе. Иначе через несколько дней заказчик вспомнит о сайте, сам вам позвонит и услышит: "А вы мне ничего не прислали-и-и, хотя я вас просил две недели назад". И не надо после этого печалиться, что заказчик разнервничался и возмущался тем, что не идет работа. А потом вообще взял и разорвал договор. Безынициативные исполнители — это кандидаты для естественного отбора.

Невербальное общение

Грубо говоря, невербальное общение — это общение без слов: жестами, мимикой и т. д. По этой теме психологами написаны целые трактаты, и обсудить даже основные моменты этой темы в этом небольшом разделе не представляется возможным. Важно следующее: наше внутреннее состояние проявляется нашими невербальными реакциями и может быть "считано"

подсознанием собеседника. По этой причине, когда вы идете на встречу, если у вас нет положительного внутреннего настроя, не стоит удивляться, что встреча будет провалена. Как бы вы не "гримировались", что-нибудь да выдаст: жесты, оттенки голоса, мимика лица, поза. Обратное тоже верно. Если вы настроены на успех, хоть и немного волнуетесь (не без этого), действительно доброжелательно относитесь к вашим собеседникам, то все ваши невербальные реакции будут работать на вас.

Примером одной из самых распространенных невербальных реакций является скрещивание собеседником рук на груди при его настороженности. (Этот жест символизирует защиту. Его аналогом является сцепление кистей рук.) Когда во время разговора собеседник скрещивает руки на груди, значит, вы сказали что-то, с чем он не согласен. Даже если внешне ваш собеседник соглашается с вами, продолжать разговор бессмысленно до того, как напряжение спадет. Потому что до тех пор, пока руки остаются скрещенными, негативное отношение сохраняется. Есть простой, но эффективный метод борьбы с этой позой. Если есть возможность, подайте собеседнику что-нибудь — к примеру, ручку с блокнотом, чтобы записать вам его адрес электронной почты, который как раз время спросить. В этом случае он будет вынужден изменить позу, раскрыть ладони и наклониться вперед, т. е. вы сможете сделать собеседника более открытым. Другой способ заключается в том, чтобы попросить собеседника наклониться вперед, чтобы что-то рассмотреть, допустим, какую-то схему, которую вы начертили. Так он тоже будет вынужден отказаться от принятой позы.

Примечание

Вариант беседы, когда вы что-то рисуете или чертите и просите собеседника наклоняться и разглядывать нарисованное, является очень неплохим, так как, во-первых, не дает собеседнику принимать защитные позы, а, во-вторых, незаметно происходит сближение расстояния между собеседниками.

Примечание

У многих людей, которые постоянно бывают на людях, вырабатывается замаскированный вариант защитной позы. То есть жест, целью которого является скрещивание рук, не доводится до конца, а ограничивается касанием часов, браслета или какого-то предмета, находящегося поблизости от руки. В таком случае барьер тоже образуется, что дает человеку чувство уверенности.

Если вы часто общаетесь с одним и тем же собеседником, вы можете достаточно точно "срисовать" его индивидуальные положительные и отрицательные невербальные реакции на те или иные события. Для этого нужно искусственно вызвать в нем положительное и отрицательное настроение и запомнить его поведение в этих случаях. Наибольшее внимание следует обращать на жесты, голос, движения глаз. На первых порах развивать такую наблюдательность достаточно сложно, однако потом это воздастся сторицей.

Постепенно приучившись обращать внимание на эти "мелочи", вы в один миг вдруг поймаете себя на мысли, что, к примеру, ваш собеседник вам врёт или чего-то недоговаривает. Хотя вербально (словесно, подконтрольной мимикой и т. д.) как будто бы все нормально. К примеру, некоторое время назад, обсуждая с коллегой план будущих работ, я после разговора почувствовал, что что-то меня гнетет, хотя внешне все было нормально: собеседник ругался, кривился и не соглашался со мной (в той ситуации именно так и должно было быть). На следующий день, вернувшись к разговору, я попытался, насколько смог, проследить его невербальные реакции. И понял, что меня гнетет. Невербальные реакции расходились со словами: на словах он ругался, а жестами своими, интонациями голоса, движениями глазных яблок вел себя точно так же, как когда предвкушал поездку в отпуск. Своевременное выяснение причин того, чему же он так возрадовался, помогло предотвратить нежелательное для нас развитие ситуации.

Примечание

Все мы слышали о "женской интуиции". Действительно, по статистике интуиция у женщин развита выше, чем у мужчин. Психологи связывают это с тем, что женщины более четко на подсознательном уровне понимают невербальные реакции. По мнению ряда исследователей это является одним из проявлений материнского инстинкта (женщины растят детей, которые в первые годы, а особенно месяцы жизни общаются сугубо невербально). По этой причине, мужчины, когда ваш заказчик женского пола, лукавство, особенно если ваш собеседник вас старше, на неосознанном уровне распознается очень быстро.

Что в имени тебе моем

Общаясь с человеком, обращайтесь к нему по имени. Как можно чаще — человеку приятно слышать свое имя.

Примечание

Кроме тех случаев, когда партнер по общению свое имя не любит. Не буду в этом разделе раскрывать этот тезис, поскольку вопрос о том, почему человеку может не нравиться его имя, к чему это приводит, и что в этом случае делать, очень многогранен. И его рассмотрение уместно, скорее, для психотерапевтической книги или философской. Остановимся лишь на одном практическом моменте, который имеет отношение не только к взаимодействию с клиентом, а вообще к общению с людьми. Обращайтесь к человеку так, как он сам вам представился. На первых порах — только так и никак иначе. Просто Дмитрий? Значит, просто Дмитрий. И не надо в этом случае спрашивать отчество — возможно, человеку оно не нравится. Всенастрот Слуссович? (Сокращения от "ВСЕгда На СТРаже Отечества" и СЛужу Советскому Союзу). Значит так. Потом, когда выясните, какие производные от имени можно еще использовать, можете обращаться и с их помощью. И то, не особо экспериментируя. К примеру, Игорям очень часто не нравится, когда их называют Гариками. А те, кото-

рым нравится имя Гарик, не очень любят, когда их называют Гошами. С именем Александр вообще проблемы, так как уменьшительно-ласкательных и всяких иных производных от него существует очень много. К примеру, если представился вам при знакомстве человек как "Сашка", не надо называть его "Шуриком" или "Саньком". Сашкой и зовите. Может он и к другим производным от своего имени относится нормально, но пока вы этого точно не знаете, лучше не экспериментировать.

Имя человека является очень важным сигнализатором отношений. Точнее то, как часто к человеку по имени обращаются. Если один человек относится к другому не очень хорошо (подсознательно или сознательно), то он может, даже работая с ним в одном кабинете, месяцами не обращаться к нему по имени. Хотя при этом оба могут быть уверены, что у них неплохие отношения.

Примечание

Вообще, имя человека, вернее, как часто люди обращаются по имени друг к другу, неплохая лакмусовая бумажка человеческих отношений. Суть реакции, как мы уже выяснили, проста: если люди друг друга недолюбливают (иногда даже сами того не осознавая), то они могут вообще по имени друг к другу не обращаться. Присмотритесь к вашим знакомым, приятелям и понаблюдайте за тем, как они друг к другу обращаются: не исключено, что заметите много интересного. Точно также, когда слышите, что кто-то ругается, но при этом ругающиеся часто называют друг друга по имени, — почти стопроцентно, что ничего страшного между ними не происходит. А вот когда, к примеру, считается, что люди относятся друг к другу нормально, но один из них про другого говорит примерно так "То, что вам сказал этот человек"... Это значит, что между ними никаких реально хороших отношений нет.

Поэтому, обращаясь к человеку по имени, вы как бы демонстрируете свое к нему расположение. И наоборот: игнорирование имени человека при общении является дополнительной отрицательной невербальной реакцией, работающей против вас.

Примечание о роли имени в судьбе

Это еще одно небольшое примечание о том, насколько важно имя человека и какую, без преувеличения, большую роль оно может играть в его судьбе. Имя, по крайней мере, у нас, выбирается не нами, а родителями, и не всегда они попадают в точку. А ведь "как корабль назовешь, так он и поплывет"... В психологии известно немало случаев, когда смена имени человеком полностью меняла его судьбу. Причем часто даже не имени, а способов его произношения. Согласитесь, что "Ванек" и "Иван", несмотря на то, что это разные вариации одного имени, по сути, являются разными именами. "Ванек" он и есть Ванек, а "Иван" уже что-то намного более обстоятельное. Но сменить имя, причем так, чтобы с ним сжиться и новое имя стало вторым именем, не каждому под силу. Если только, конечно, вы не писатель, и не собираетесь брать себе псевдоним. И в этом смысле очень неплохую службу играет Интернет. Ники, по сути, есть те самые имена, которые человек выбирает осознанно. Зачастую люди сживаются со своими никами настолько, что как их зовут "в миру", уже никто и не помнит...

Основной закон психологии общения

Часто приходится слышать:

— Ну вот, ведь я же убедил его, по полочкам все расписал, он говорил, что все понял, а все-таки мне отказал. Выходит, ничего он не понял.

Все он понял, ваш партнер по общению. **Но понять и принять — это совершенно разные вещи!** Между ними может быть пропасть. Это и есть основной закон. Приведу пример. Вспомните себя в институтские годы. Как сидите вы на лекции, допустим, по квантовой механике. Лекция идет первый раз. Выходит лектор, мужчина 60 лет, профессор, зав. кафедрой этой самой механики, и говорит:

— Квантовая механика очень важный предмет! Я бы даже сказал, самый важный во всей физике! И про того, кто не знает квантовую механику, я могу с уверенностью сказать, что он не знает и не понимает физики и не достоин носить гордое звание физика!

Вы поняли, что хотел сказать профессор? Уверен, что поняли: профессор хотел сказать, что "кванты" это очень важный предмет. Более того, вы поняли даже больше: подсознание профессора хотело сказать, что "тот, кто не понимает предмета, в котором я являюсь специалистом, — просто моральный урод". То есть все вы поняли, и даже больше, чем он сказал. **А вот приняли ли?** Уверен, что нет. Понять то поняли, а про себя подумали: "Ну, все понятно, старая обезьяна кривляется".

Понимаете, что из этого вытекает? **Не важно, кто и что понял, а важно принял ваш собеседник вашу мысль или нет!** Конечно, идеальный вариант тот, при котором вас и поняли и приняли. То есть когда вы и доказали, и убедили. Но это не всегда выходит. А если идеального варианта не получилось, то главное, чтобы приняли, а что при этом поняли — не очень важно.

Примечание

Более того, некоторым людям совершенно не важно, чтобы им что-то доказали. Им важно, чтобы человек им просто понравился. То есть они работают практически всегда только на уровне принятия.

Особенно важно понимать это правило, когда вы что-то продаете. То есть убеждаете человека расстаться с деньгами! То есть работаете в изначально неблагоприятных психологических условиях, поскольку клиент имеет на подсознательном уровне отрицательную мотивационную установку! (То, что клиент за деньги приобретет товар, это для него вторично.) Рассмотрим два варианта общения с клиентом в этом случае. Как водится, один вариант плохой, другой нормальный. Усложним задачу, добавив к ее условию то, что вам не только надо продать продукт, но еще и увеличить за него цену.

Первый вариант (наиболее типичный при переговорах).

— Вы хотите сделать усовершенствование в этом программном блоке? За эту сумму это у вас не получится. Это будет дороже.

— Почему?

— Потому что это потребует от нас дополнительной работы.

— А насколько дороже?

— Не знаем. Надо посчитать.

Понял вас заказчик? Конечно, понял. Он прекрасно понял, что вы хотите дополнительной оплаты. А вот принял ли? Вряд ли. Ведь что в этом случае думает заказчик? То, что "эти программисты совсем оборзели и за полчаса дополнительной работы требуют еще кучу денег. Совсем люди стыд и совесть потеряли, всем бы только меня ободрать". То есть получилось то, что изначальная отрицательная мотивационная установка не только не снята, а наоборот усилилась. Кроме того, фраза "надо посчитать" в данном контексте уж совершенно неуместна, так как подсознание клиента переведет ее примерно следующим образом: "Ага, этим бесстыжим тварям еще и посчитать надо, насколько меня обдурить". Помочь вам выиграть заказ в таких неблагоприятных условиях могут только три фактора:

- ❑ то, что ваши соперники разозлили заказчика еще больше, чем вы, и он по принципу "из двух зол..." выберет зло меньшее — вас, то есть;
- ❑ то, что вы монополист, и заказчику в принципе больше не к кому обратиться;
- ❑ то, что вы очень удачно всех загипнотизировали своей PR-политикой, в силу чего все считают, что вы круче всех разрабатываете программы, делаете сайты и т. д., и по этой причине обдираете клиентов как липки.

Второй и третий варианты встречаются не так уж и часто, поэтому мы их принимать во внимание не будем (Хотя бы потому, что если вы либо монополист, либо достигли такого состояния развития фирмы, когда все уже под гипнозом от вашей крутости, то вам читать этот раздел совершенно не обязательно. Клиенты и так к вам идут, словно овцы на заклинание, и разговаривать с ними можно как угодно.)

А теперь давайте рассмотрим более грамотный вариант общения, при котором клиент переходит на позицию принятия ваших доводов. Основная суть состоит в переносе его раздражения с вас ("твари бесстыжие") на внешние, от вас не зависящие факторы ("увы, так действительно обстоят дела"). То есть вы должны показать, что та работа, которую вы предлагаете клиенту, действительно стоит денег. Что это не на вас блажь нашла или жадность напала, а так обстоят дела. Здесь есть важный момент. Ваше представление и представление клиента о том, как обстоят дела с разработкой программных блоков, совершенно различны. Это надо четко понимать. Клиент совершен-

но может не представлять (и не обязан этого делать), почему одна программа стоит триста долларов, а другая — две тысячи. И объяснять ему, что во второй присутствует какая-то эдакая внутренняя начинка, дело совершенно бесполезное. Хотя бы потому, что клиент, как мы уже говорили, в большинстве случаев на уровне подсознания настроен к вам не очень благожелательно. Здесь лучше всего дать клиенту сделать выбор самому. К примеру, продаете вы счетчик посетителей. За который берете сто долларов. Клиент может возмутиться, сказав что-то типа "За что деньги берете, чего там делать-то! Посчитать и все". На что вы ему отвечаете, что если "просто посчитать", то это стоит десять долларов. (Желательно наглядно продемонстрировать, как это выглядит: просто цифры внизу главной страницы сайта). Что там действительно нечего делать, и вы даже вообще можете подарить его клиенту бесплатно. А вы предлагаете не "просто посчитать", а предоставить полную статистику по посетителю: сколько раз он посетил ту или иную страницу, с какой поисковой системы пришел, какую поисковую фразу набрал, какой у него IP-адрес и т. д. И при этом демонстрируете красиво раскрашенный полнофункциональный счетчик посетителей. После чего замолкаете и предлагаете клиенту сделать выбор. С большой долей вероятности выбор будет сделан в пользу второго варианта.

Во-вторых, вы должны продемонстрировать, что готовы сделать клиенту, к примеру, сайт за те деньги, что он готов потратить, и он будет хорошим, но просто не будет некоторых дополнительных программных блоков, что, в общем-то, не страшно.

Важный лингвистический момент

В предложении выше есть фраза "за те деньги, что он готов потратить". Этой фразой вы в подсознательное клиента вкладываете положительную установку, поскольку из этой фразы видно, что вы считаете, что, во-первых, клиент человек не бедный, но просто не считает нужным в данный момент тратить больше и, во-вторых, клиент тратит деньги разумно. А вот если бы вы сказали по-другому, к примеру, "за те деньги, что у вас есть", эффект был бы прямо противоположный, так как на подсознательном уровне вы клиента оскорбили, по сути, назвав его бедняком.

Ну и, в-третьих, вы должны показать клиенту что-то дополнительное, что вы ему предлагаете, либо очень красиво (к примеру, презентационная flash-страница), либо эксклюзив. Не забыв при этом добавить, что в принципе можно обойтись и без этого. (Этими повторениями, что "в принципе все будет нормально и без этого", вы убиваете двух зайцев. Во-первых, как бы отстраняетесь от решения вопроса, предоставив клиенту право делать выбор самому. В силу чего его подсознание уже не может вас заподозрить ни в чем негативном, к примеру, в том, что вы что-то навязываете и просто хотите содрать деньги. Вы как бы говорите: "В мире есть вот такие красивые вещи, и если хотите, то можете их купить. А я тут вообще не при чем, просто могу помочь вам их приобрести".)

В любом из нас спит гений. И с каждым днем все крепче

Часто возникает вопрос: "О советах по применению тех или иных приемов общения наверняка слышал мой собеседник. Он же тоже их будет применять, и что из этого получится?" Здесь два варианта ответа. Если ваш собеседник действительно может квалифицированно общаться, то вы в любом случае в выигрыше. Потому что быстрее достигнете желаемой цели. Однако это, увы, случается не так часто. Дело здесь в том, что наличие тех или иных книг на полке не означает владение теми приемами, которые в них описаны. Наверняка у многих из ваших знакомых есть классические книги Д. Карнеги. А теперь вспомните, кто из них в совершенстве овладел ораторским искусством или "перестал беспокоиться и начал жить"? В лучшем случае процентов пять. Потому что устойчивые знания и навыки достигаются только каждодневными тренировками. К умению общаться это относится так же, как к умению программировать, драться и т. д.

Личная встреча — лишний шаг к успеху

Не доверяйте важные переговоры электронной почте и телефону. По возможности организуйте личную встречу. При личной встрече у вас всегда есть возможность сразу видеть, как реагирует собеседник на ваши слова, и быстро подстроиться под него. Есть мудрая пословица "Хотите провалить переговоры — договаривайтесь по телефону". То же самое относится и к электронной почте. Даже после того, как договор подписан, первое время и ключевые моменты всегда следует обсуждать лично, даже если заказчик сам предлагает "переслать ему дизайн по e-mail". По электронной почте стоит общаться только для обсуждения незначительных технических деталей (т. е. уже ближе к концу работы над проектом), либо с уже знакомыми партнерами.

Не возражайте в лоб

Проведите со своими знакомыми такой простой тест: начертите на бумаге линию, сверху напишите "Самый умный", внизу "Самый глупый". И попросите их оценить себя по этой шкале. Почти все поставят свои точки выше середины. Тест этот называется **тестом Дембо**. Его результаты означают, что почти никто на самом деле, а не на словах, не считает себя дураком. Когда вы идете на встречу, то знайте, что ваш собеседник считает себя как минимум не глупее вас. Если вам двадцать, а ему сорок, то почти всегда — умнее. Теперь, после этого краткого предисловия, представьте, что сразу в лоб воз-

ражаете на какое-то предложение вашего собеседника. Ему это не понравится, так как это не нравится никому. Уважайте своего клиента. Возьмите за правило, даже если вы не согласны, некоторое время подумать. После того как ваш собеседник увидит, что к его предложению отнеслись со вниманием, ваше возражение на его слова уже не будет воспринято болезненно.

ЯЗВа

Когда то в Интернете нашел такое определение заказчика — ЯЗВа (сокращение от "Я Знаю Все"). Увы, не помню фамилию того остроумного человека, который это придумал, но определение чудесное. Если вы столкнулись с таким типом заказчика, то в первую очередь помните, что ваша цель все-таки получить деньги, а не учить кого-то бесплатно уму-разуму. От этого и пляшите. Если же хотите вдобавок, чтоб в портфолио был сайт, который было не стыдно показать людям — без привлечения психологии не обойтись. "Птичку вверху, чтоб крыльями махала? Пожалуйста!" А на следующей встрече можно сказать, что птичка не очень хорошо смотрится. Отставленное несогласие будет восприниматься вашим всезнающим собеседником не так болезненно, как если бы вы возразили сразу. Кстати, если вам встретился заказчик типа "ЯЗВа", это значит, что у него для вас припасено немало достоинств. К примеру, его несложно испугать. И пугайте на здоровье! Если человек принадлежит типу "Я знаю все", значит, его давно никто не пугал, и вы просто восполняете допущенный природой пробел. Несколько примеров. Заказчик хочет Flash, вы о нем не договаривались, а заказчик типичный холерик по темпераменту и сразу рвет договор при малейшем с ним несогласии. С выражением лица "только вам по секрету" говорите ему, что Flash отрицательно влияет на индексацию поисковыми системами, и сайты с Flash индексируются в последнюю очередь. И т. д., с подробным объяснением того, чем это грозит заказчику и парой-тройкой ужасных примеров "из жизни".

АнтиЯЗВа

Иной пример. Вы пришли к заказчику, и слышите что-то типа "Господа, я в этом деле ничего не понимаю, вы уж на меня, на дурака, внимания не обращайтесь". Вот в этом случае — максимум внимания и серьезности. На 50% перед вами достаточно компетентный человек, и если вы получаете от него отказ, это означает, что количество лапши на его ушах превысило предельно допустимую концентрацию. Еще на 30% — компетентный человек находится или где-то рядом, или потом прослушает запись разговора с вами. Ну и 20% — перед вами честный с самим собой и с вами человек, работать с которым просто удовольствие.

Закон об объеме оперативной памяти

Этот закон гласит, что оперативная память человека способна усвоить от пяти до девяти (7 ± 2) информационных знаков, независимо от их природы. (Число семь вообще магическое: семь нот, семь цветов радуги, семь дней недели, "семеро одного не ждут", "семь раз отмерь"). Наша оперативная память работает около 5—20 секунд. По истечении этого времени информация либо поступает в кратковременное хранилище (где может находиться до 5—7 дней), а затем в долговременное хранилище (от 5—7 дней до бесконечности), либо забывается. Установлено, что более половины взрослых людей не в состоянии запоминать на слух предложение, насчитывающее 14 слов. Одна треть взрослых людей забывает начало фразы уже тогда, когда произносится одиннадцатое по счету слово. Если цепочка произносимых слов длится более 6 секунд, слушатели теряют нить рассуждения.

Примечание

Закон об объеме оперативной памяти называется **законом Миллера** и был открыт в 50-х годах прошлого века группой психологов под руководством Д. Миллера во время их работы по заказу ВМФ США.

По этой причине, если вы хотите, чтобы заказчик вас понял и усвоил, что вы говорите, в разговоре произносите не более 2 слов в секунду, при этом стараясь строить фразы, состоящие из 7—9 слов.

Зная о законе оперативной памяти, понятно также, что делать, если нужно наоборот, чтобы заказчик и окружающие мало что поняли (тоже нередкая задача).

Закон края (закон Эббингауза)

Этот закон многим известен: лучше всего запоминается то, что находится в начале и в конце (речи, текста и т. д.). Применительно к нашей ситуации то, что вы хотите, чтобы лучше запомнилось собеседником, говорите в начале и в конце беседы. По этой же причине будьте наиболее доброжелательными в начале и в конце беседы. Это и запомнится. И когда ваш собеседник будет потом в памяти прокручивать разговор с вами, у него останется приятное впечатление. Соответственно, старайтесь не начинать и не заканчивать беседу с выяснения "щекотливых" вопросов.

Закон контрастов

Запоминание становится более эффективным при чередовании контрастных объектов умственной деятельности, учебных предметов, идей.

Примечание

По этой причине в расписании занятий стараются чередовать дисциплины естественно-математического и гуманитарного циклов. Такое чередование балансирует работу левого и правого полушарий и способствует успешному усвоению учебной информации.

Что дает нам этот закон применительно к беседе с заказчиком? Очень много. На этом законе построены многие из технологий убеждения, запоминания, развития способностей. К примеру, при прямой попытке изменить точку зрения собеседника, стоящего на противоположной позиции, неминуемо приходим к конфликту. Здесь срабатывает эффект контраста — чем больше мы его убеждаем, тем он становится упрямее и наши шансы убедить его становятся все меньше. Изменить позицию собеседника можно лишь терпеливым и последовательным убеждением, помня при этом, что каждый шаг в этом направлении приводит лишь к частичному (!) изменению позиции вашего партнера. Желательно делать перерывы в беседе, во время которых партнер постепенно переходит на вашу позицию, а вы частично принимаете его позицию.

Принцип ледокола или еще одно следствие из закона контрастов. Убеждая собеседника, можно поступать примерно так. Через какое-то время уводить собеседника от обсуждаемого вопроса. Потом снова к нему возвращаться. Одно "но": переводить тему надо естественно, а без тренировок это сделать непросто. Если подходящего повода для смены темы нет, то желательно делать перерывы в беседе, во время которых партнер постепенно переходит на вашу позицию, а вы частично принимаете его позицию. Перерывы тоже должны быть естественными: вы можете попросить воды (в горле пересохло), попросить срочно позвонить по телефону и т. д.

Примечание

Этот прием я называю "**принципом ледокола**". Вспомните, как работает ледокол: для того, чтобы расколоть льдину он наваливается на нее и затем отходит назад. Набирает нужную скорость и снова наваливается. И так до тех пор, пока не расколется. Это единственный способ расколоть льдину. Если просто в нее упереться и все время "работать полный вперед", это ни к чему не приведет (хорошему). А теперь вспомним, как часто мы поступаем наоборот при убеждении собеседника. При этом удивляясь тому, как он непроходимо туп, когда с нами не соглашается.

Не уходите от скользких вопросов

Точнее говоря, от них надо уходить всеми силами, но если уж неприятная тема затронута, то не уходите от нее явно. Убегая от неприятного вопроса, вы пробуждаете в собеседнике инстинкт преследователя, и следующие вопросы

будут только хуже предыдущего. В конце концов, раззадорившийся собеседник, скорее всего, вам откажет. По такому сценарию срывается много беседований. Ситуация для примера: нервный заказчик на предыдущей встрече сказал вам что-то дорисовать, вы согласились с его мнением, а дорисовать забыли. Рассмотрим несколько возможных вариантов поведения.

Заказчик: А где тот рисунок, который я вас просил нарисовать?

Вы: Ой, простите, забыл.

Этой фразой вы вводите себя в роль жертвы, которую надо догнать и уничтожить. Конечно, лучше не забывать, но если такое случилось, сделайте по рецепту Д. Карнеги "из лимона лимонад".

Заказчик: А где тот рисунок, который я вас просил нарисовать?

Вы: Не волнуйтесь, я о нем помню, эта идея очень хорошая и обязательно будет реализована, просто для ее полноценной реализации необходимо провести некоторые подготовительные работы.

При таком варианте вы не только отводите огонь от себя, но и делаете скрытый комплимент заказчику (фраза "эта идея очень хорошая").

Еще вариант.

Заказчик (с раздражением, переходящим в гнев): Я хотел бы знать, где тот рисунок, который я вас просил нарисовать?

Вы, тоже с гневом, смотря то на заказчика, то на монитор: "Я тоже!" (При таком варианте развития событий заказчик теряется, так как он ожидает от вас всего чего угодно, только не гнева в его адрес.) "Я тоже хотел бы это знать!", — повторяете вы, еще больше распаляясь. И продолжаете: "Как мне надоели эти хостинг-провайдеры, не могущие настроить свои прокси-серверы, которые кэшируют все на свете!" После этого можете понизить тон и объяснить заказчику, что картинку вы поместили, но она просто не отображается, так как негодный прокси-сервер еще не обновил свой кэш.

Несколько правил ведения дискуссии

В этом разделе мы рассмотрим несколько основных правил ведения дискуссии. Причем природа дискуссии не имеет никакого значения: это могут быть деловые переговоры, выступление перед аудиторией, беседа с начальником.

Никогда не перебивайте собеседника

Этим недостатком грешат многие из нас. Между тем, когда вы перебиваете, кроме того, что это просто невежливо, вы еще на подсознательном уровне

считаете собеседника глупцом, и он это на своем подсознательном уровне прекрасно понимает. Потому что, перебивая, мы как бы говорим партнеру: "Послушай лучше меня, потому что я умнее тебя. А все что ты хочешь сказать, я уже давно понял". Даже если вы уже поняли, что хочет сказать собеседник, в любом случае его внимательно дослушайте. Точно также следует поступать и на больших совещаниях, когда вам задают вопросы. Даже если вы уже поняли, о чем вопрос, даже если он действительно не очень умный, все равно обязательно дослушайте вопрос до конца. Сколь бы глупым он не был, и сколь бы долго он не произносился. В этом случае на вашей стороне будет и задающий вопрос, который будет благодарен вам, что вы его выслушали, и остальная аудитория, которая будет поражаться вашему терпению.

Не рассуждайте о том, чего сами не знаете

Или знаете не очень твердо. Этим грешат очень многие. По двум причинам. Одни просто стараются казаться умнее, чем они есть на самом деле, и пускаются в любые дискуссии, даже по тем вопросам, о существовании которых только вчера услышали по радио. Между тем выглядит это достаточно смешно. Другие надеются, что "по ходу дела" найдут ответ на вопрос. Если это и происходит, то очень нечасто. Намного чаще ваш оппонент прекрасно замечает, что вы "плаваете", и не упускает случая "утопить" вас окончательно. Я сам до некоторой поры грешил этим недостатком, и, выступая перед аудиторией на научных конференциях, пытался сходу отвечать на вопросы, на которые сразу дать ответ весьма сложно. Ничего хорошего из этого не выходило. Даже если по ходу ответа вас и "осеняет", то оформить свою мысль в доступную форму удается далеко не всегда. И в результате, если посмотреть видеозапись такого ответа, видно, что выглядит отвечающий весьма неуверенно. Поэтому, даже если вас принуждают к ответу на вопрос, на который вы ответить не готовы, гораздо выгоднее сказать "Я не могу сразу дать ответ, мне нужно подумать", чем сломя голову кидаться в омут дискуссии. Как говорится, поздно учиться плавать, когда уже барахтаешься в воде.

Не спорьте с очевидным

Даже если это очевидное вам очень не нравится. Самое жалкое и самое проигрышное, что только можно видеть, это когда кто-то спорит с очевидными фактами. Максимум, что можно сделать, если уж факт вам очень не нравится, это несколько принизить его значение или, если получится, постараться обернуть этот факт в свою пользу. Простой пример. Приехали вы на тендер, где кроме вас есть еще представители нескольких компаний. И вам говорят, что ваша продукция дороже, чем у остальных. Если это правда так, то спорить с этим глупо. Проще согласиться. Но прежде, чем

соглашаться, можно задать собеседнику вопрос, казалось бы, не имеющий отношения к данной проблеме. К примеру, так.

Собеседник: "Ваш продукт намного дороже, чем у остальных".

Вы: "Вы не считаете меня глупым человеком?" (на этой "уводящей" фразе многие теряются и всегда отвечают стандартно, хотя бы из приличия).

Собеседник: "Конечно, нет".

Вы: "Значит, на то есть свои веские причины. Если вы пожелаете, я вам один на один их озвучу".

А уж потом один на один говорите про высокое качество, прекрасную гарантийную поддержку, за которую конкуренты берут деньги, поэтому, в конечном счете, у них получается дороже, а вы, как честный человек, сразу озвучиваете реальную цену.

Да поможет нам НЛП

НЛП расшифровывается как нейролингвистическое программирование и представляет собой одно из направлений психологии. Часть "нейро" этой аббревиатуры относится к нервной системе, а часть "лингвистическое" — к нашей способности использования речи и к тому, каким образом употребляемые нами речевые обороты характеризуют наш внутренний мир. Ну, а программирование здесь появилось затем, чтобы показать, что наши мысли, чувства и действия являются результатом выполнения наших "внутренних программ", изменив которые можно изменить свою жизнь.

Примечание

Более подробно о НЛП мы поговорим в соответствующем разделе *Приложения 2*, а в этой главе из НЛП заинтересует лишь небольшая часть, которая серьезно может облегчить общение с клиентом, а именно — *системы представления*.

Итак, о системах представления. Согласно концепции НЛП, мы обрабатываем информацию, пользуясь четырьмя основными системами представления:

- ☐ визуальной (зрение);
- ☐ аудиальной (слух);
- ☐ кинестетической (ощущения);
- ☐ аудиально-дискретной (внутренний диалог с самим собой).

В той или иной мере мы используем все системы представления, но у каждого из нас есть "своя любимая". Знать систему представления собеседника — неплохо, так как подстройка к его системе помогает значительно по-

высить конструктивность диалога. Приведу пример. Пусть вы склонны к аудиальной системе представления, а ваш собеседник — к визуальной. Это значит, что вы лучше всего воспринимаете информацию со слуха, а он — путем анализа зрительных образов. То есть, то, что вы говорите, ему гораздо легче было бы воспринимать, если бы он видел перед собой какие-то схемы. Поэтому — не поленитесь, понаблюдайте немного за вашим партнером, и постарайтесь подстроиться под его систему представления. Основные признаки этих систем мы сейчас и рассмотрим.

Примечание

В НЛП существует понятие "сенсорных предикатов", которое нам понадобится в дальнейшем. Грубо говоря, это "слова-ощущения" или "мыслеформы". То есть, к примеру, выражение "звучит неплохо" — аудиальный сенсорный предикат, характерный для человека с аудиальной системой представления.

□ Визуальная система.

Люди этого типа запоминают информацию с помощью мысленных образов, и их внимание сложно отвлечь звуками. В разговоре пользуются визуальными предикатами, к примеру: "У меня сложилось такое видение ситуации", "как мне видится, это надо сделать так-то", "итак, что мы видим". Такому человеку в разговоре лучше начертить схему и сказать "давайте посмотрим на эту схему". Если человек с визуальной системой долго слушает только одну речь, то его внимание начинает "плыть", что тоже нужно учитывать. Люди визуальной системы практически всегда собраны и опрятны (чего, кстати, ждут и от вас). Манера речи — быстрая, высота голоса, как правило, немного повыше нормальной, взгляд направлен чуть вверх.

□ Аудиальная система.

Люди аудиальной системы склонны пользоваться аудиальными предикатами (в скобках для сравнения приведены визуальные предикаты): "звучит неплохо" (вместо "смотрится неплохо" для визуальщика), "что я слышу?!" (вместо "что я вижу?!"), "вы слышали, что вчера по телевизору сказали?" (вместо "вы видели..."). Люди с аудиальной системой говорят ритмично, с богатыми голосовыми интонациями. При обдумывании проблемы склонны отводить глаза в сторону, как бы прислушиваясь к чему-то. Аудиальщики легко усваивают информацию на слух и предпочитают слышать отклик собеседника на свои высказывания.

□ Кинестетическая система.

В речи кинестетики склонны пользоваться кинестетическими предикатами, обозначающими различные ощущения: "меня в озноб ударило от этой мысли", "все идет гладко", "я получил заряд бодрости", "это нагоняет

тоску", "проблема обрушилась на меня" и т. д. Когда кинестетик пребывает в задумчивости, взгляд его направлен, обычно, вправо вниз. Темп речи в спокойном состоянии медленный, нередко с большими паузами между словами. Кинестетики хорошо реагируют на дружеские прикосновения, похлопывания по плечу, и сами могут себе такое позволять. Процесс запоминания нового происходит путем мысленного повторения всего процесса (к примеру, если надо запомнить основные моменты беседы, кинестетик, мысленно "проигрывает" ее всю от начала до конца). В общем это — человек ощущений. Кстати, такие люди неплохо относятся к решению различных вопросов во время застолий, совместных походов в баню и т. д.

□ Аудиально-дискретная система.

Люди с такой системой представления много времени уделяют мысленному общению с самими собой. Для них важно только то, что имеет логику. Обороты их речи достаточно сложные. В речи любят пользоваться обилием подробностей. Как правило, очень эгоцентричны. Не дай вам бог такого человека похлопать по плечу или предложить "порешать вопрос в баньке" — такого обращения они не терпят, считая это недопустимым панибратством. Подобные методы они могут позволить лишь только очень узкому кругу близких людей. Во время внутреннего диалога смотрят, как правило, влево вниз.

Примечание

Иногда людей с аудиально-дискретной системой представления называют **дигиалами** (от англ. *digital* — число). Это связано, в частности, с тем, что наибольшее количество дигиталов среди программистов и шахматистов.

Не знаю, везение это или нет, но, мне, как правило, нередко попадают люди именно с этой системой представления. С одной стороны, везение. Потому что это, как правило, очень глубокие, неординарно мыслящие натуры, встреча с которыми — счастье. Но, как говорится, чем более блестяща одна сторона медали, тем более темна другая: если человек, обладающий такой системой представления, излишне самозациклен, то общение с ним никакого удовольствия не доставляет. Потому что никто, кроме него, ему не интересен. Да, он может быть крайне эрудирован, но мало кому хочется выслушивать незапланированные двухчасовые лекции на тему "Ах, неужто вы не читали дневников Набокова? Напрасно, напрасно". И говорит он это, конечно, не для того, чтобы вас просветить, а для того, чтобы показать собственный ум. Остановлюсь еще на двух отрицательных моментах, характерных для излишне самозацикленных аудио-дискретчиков. Это — совершенно потрясающие хронофаги (пожиратели времени). С ними можно говорить три часа, и ничего полезного не вынести. Вы узнаете об их мнении по пово-

ду палестино-израильского конфликта, будете все знать об их болезнях и сопутствующих им душевных переживаниях, но — ничего по делу. Поэтому берегите свое время и не давайте уводить себя в сторону. Кроме всего прочего, после такого разговора вы еще будете выжаты, как лимон и раздосадованы на то, что столько времени пропало даром. Другой момент состоит в том, что такие люди, как магнитом притягивают к себе различные неприятности, рискуя и вас затянуть в свою жертвенную орбиту (см. разд. "Немного о виктимологии" главы 10). Но, возвращаясь назад, повторюсь, что если вам встретится несамозащищенный аудио-дискретчик, а действительно глубокий человек, это — подарок судьбы.

Мы снова говорим на разных языках

Очень часто мы не можем "найти подход" к собеседнику именно из-за различия в наших ведущих сенсорных системах. Поэтому если хотите установить хороший контакт, употребляйте те же слова, что и ваш собеседник. То есть на вопрос "Вы это видите?" не надо отвечать "Да, я это чувствую". Ваш собеседник визуал, и отвечать ему тоже надо визуально, к примеру: "Да, я это заметил".

Очень часто в семьях, где живут люди с ярко выраженными разными системами представления, можно увидеть примеры того, как одна сенсорная система вступает в конфликт с другой. К примеру, имеем пару: муж с аудио-дискретной системой представления и жена с визуальной. Мужу, конечно, совершенно наплевать на беспорядок в комнатах, а жена его все время за это "пилит". И ее очень даже можно понять: для нее, как визуала, беспорядок очень страшен, он причиняет ей серьезный душевный дискомфорт. Если муж, скажем, компьютерщик, то потрясение жены, когда она видит беспорядок, можно сравнить с потрясением мужа, если у него отобрать компьютер.

Поэтому при переговорах, да и в жизни вообще, не ленитесь подмечать, какая система у ваших коллег, партнеров, друзей является ведущей. Это обезрежет вас от многих ненужных конфликтов и сделает существование более приятным и продуктивным.

Примечание

Возможно, вам доведется увидеть людей, у которых все четыре сенсорные системы являются ведущими. Редкий случай, очень редкий. Но — вполне объяснимый. Ведь у ребенка именно так и происходит — он без особого труда может переключаться между различными системами. Эдакое "не пойми что" от психологии. С возрастом и воспитанием какая-то одна система становится ведущей. И взрослый "не пойми что" просто сумел сохранить в себе эту "детскую психологическую непосредственность".

Подстройка и ведение

Вы наверняка не раз замечали, что люди, давно знающие друг друга и симпатизирующие друг другу, нередко ведут себя одинаково: у них одинаковый тембр речи, жесты, мимика и т. д. В связи с этим в НЛП существует метод подстройки, который заключается в том, что вы как бы "присоединяетесь" к вашему собеседнику путем подстраивания к его телодвижениям, речи и т. д.

Внимание

Поначалу очень часто многие путают подстройку с передразниванием. Поэтому нужно твердо усвоить, что подстройка — это не обезьянничание, не подражание и не передразнивание, которое очень заметно и оскорбительно. Подстраиваться нужно с уважением. И очень четко и чутко. (Иначе лучше этим приемом совсем не пользоваться — будет только хуже). Приведем аналогию с обычной беседой. Пусть собеседник высказал какой-то довод. Вы можете эпатажно воскликнуть "Да я согласен!!!", всем своим видом и голосом выражая мысль о том, что, мол, еще в общении с дураком остается делать, как не соглашаться. Ясно, что это — обезьянничание. А можно сказать "Да, я согласен" спокойно и без эмоций, что воспринимается, как мы все знаем, совершенно нормально и говорит о том, что два собеседника достигли понимания. Вот подстройка и есть то самое спокойное и безэмоциональное согласие по второму варианту. Только соглашаетесь вы на невербальном уровне.

Различают разные виды подстройки. Можно подстраиваться к движениям рук, к позе собеседника, к частоте его дыхания, тембру речи... Только, опять же, подстраиваться нужно аккуратно и постепенно. К примеру, подстройку к движениям рук на начальном этапе можно выполнять аналогичными движениями, но только кистей рук, а не всей руки. Если подстраиваетесь к голосу, то сначала подстройтесь к какой либо одной его характеристике: громкости, тембру и т. д. И старайтесь не подстраиваться к голосу полностью: согласитесь, достаточно странно будет выглядеть, если вы со своим глубоким басом вдруг полностью подстроитесь к писклявому голосу вашего собеседника. При подстройке к голосу подстраиваться лучше всего к громкости.

Примечание

Подстройка с отставанием по фазе. Достаточно эффективный способ подстройки состоит в том, что вы подстраиваетесь не сразу, а спустя некоторое время. Конечно, спустя секунду-две, а не час-второй. Этот прием я назвал "**подстройкой с отставанием по фазе**". Суть в том, что, скажем, движение руки собеседника вы повторяете не сразу, а спустя пару секунд, после того, как он его совершил. Под изменение позы собеседника можно подстраиваться уже спустя более длительное время: через минуту-другую.

Очень мощным средством является подстройка к частоте дыхания. Если осуществлена грамотная подстройка к дыханию, то, как говорят "энэлписты", —

это признак глубокого **раппорта**. Кроме того, подстройка к частоте дыхания является самой незаметной для собеседника.

Примечание

В терминах НЛП подстройка называется — **раппорт**. Еще говорят "установить раппорт", что означает "присоединиться" к другому человеку. Раппорт и подстройку еще очень часто называют отзеркаливанием, потому что при раппорте мы как бы зеркально повторяем действия собеседника.

Кстати, подстраиваться нужно только к положительным или нейтральным жестам. Если же жесты и мимика вашего собеседника выражает недоброжелательное к вам отношение, то здесь наоборот его нужно как то отвлечь: к примеру, если он размахивает кулаками и кричит — попросите его что-то нарисовать, дав ему в руки авторучку. И так далее.

Примечание

Очень действенный прием выведения собеседника из отрицательного состояния состоит в том, что вы сначала подстраиваетесь к этому состоянию, а потом начинаете постепенно и очень понемногу отстраиваться, ведя тем самым собеседника за собой (этот прием так и называется — **ведение**). Прием этот работает не только в конкретных тактических ситуациях в частности, но и в жизни вообще. Примерно так умные люди выводят своих легковых родственников из различных сект, вступая туда вслед за ними и постепенно доводя своих подопечных до того состояния, когда они сами понимают, что делают глупость.

Подстройка и ведение имеют много различных применений, и не только при ведении переговоров. Так, к примеру, можно успокаивать рассерженных собеседников, слегка подстроившись к их гневу, и потом постепенно начав успокаивать себя — вслед за вами успокаивается и собеседник. Очень просто, к примеру, используя комбинацию этих приемов, укладывать детей спать. Это очень нередкая ситуация, когда ребенку вроде бы уже и отдыхать пора, а он все играть хочет, и капризничает, когда родители волевым голосом сообщают ему, что ему уже пора в койку. А можно и по-другому. Играть? Пожалуйста! Начинайте играть вместе с ребенком, постепенно подстраиваясь под него (хотя достаточно и того, что вы просто начали с ним играть). И потом постепенно начинайте зевать, прикрывать глаза, задремывать на пару секунд и, встряхиваясь, просыпаться... Десяти-пятнадцати минут, как правило, достаточно, чтобы ребенка убаюкать. Кстати, этим нехитрым приемом прекрасно убаюкиваются и многие взрослые. Можете на какой-нибудь вечеринке провести такой эксперимент. Подстройтесь под человека и начинайте делать так же, как описано для ситуации с ребенком. Зачастую не надо даже особо подстраиваться, достаточно просто, чтобы те, кого вы хотите убаюкать, занимались одним делом — к примеру, смотрели одну телевизионную передачу. Другой классический пример подстройки и

ведения состоит в том, что если ваш собеседник барабанит пальцами по столу (достаточно часто встречающаяся неприятная привычка), начинайте вместе с ним барабанить в том же темпе, постепенно затем снижая скорость постукивания до полного прекращения. Собеседник в 9 случаях из 10 также прекратит постукивать.

Теперь немного об отстройке. Естественно, если к человеку можно подстроиться, то от него можно и отстроиться. Причем отстройку тоже не нужно понимать как способ поругаться. Наоборот — это достаточно вежливый прием, к примеру, окончания разговора. Скажем, телефонного. В котором вы сначала подстраиваетесь к голосу собеседника, а потом отстраиваетесь, что служит для него сигналом к окончанию разговора.

Влияние установок

Попросту говоря, **установка** — это наш настрой на то или иное событие. Идем мы на концерт известного и уважаемого певца — понятно, что доминирующая установка положительна. И даже если он опоздает, и концерт начнется с задержкой, ничего страшного не случится. А если установка нейтральна, или частично отрицательна, то задержка на то же время может обернуться народным гневом, сдачей билетов и прочими неприятными вещами.

Примечание

Помню, когда первый раз пошел на концерт А. Розенбаума, выступление немного задержалось. Но поскольку установка на этого уважаемого человека у всех была, естественно, положительной, то люди только шутливо перешептывались на тему "Мол, в кои-то веки в народ вышли, не грех и подождать немного". И другой пример, тоже концертный. С не менее известным Ю. Антоновым. В то время, когда он приезжал с концертами в наш город, против него в прессе прошла серия очернительных публикаций на тему его выступлений под фонограмму. Потом все разобрались и поняли, что это просто какая-то обиженная журналистка написала, но дело было сделано. Установка поставлена. Не сказать, что сильно отрицательная, но, скажем так, концерта люди ожидали с каким-то внутренним напряжением. И когда начало концерта, так же как в случае А. Розенбаума, несколько затягивалось, возмущенных возгласов было не в пример больше. А первые несколько песен певцу почти никто не аплодировал, так как все старались понять, под "фанеру" он поет или нет. То есть артисту было достаточно трудно продираться через отрицательную установку, что он, кстати, сделал блестяще. Потом уже все и аплодировали, и стоя пели и плясали.

Влияние установок сложно переоценить. Оно колоссальное. В подтверждение этого рассказу еще один широко известный анекдотический случай, который в действительности имел место.

Примечание

В одном городе в одно и то же время в разных местах должны были состояться два выступления: лекция знаменитого академика на научно-популярную тему и встреча со знаменитым клоуном. (Кстати, надо заметить, что академик был почти полностью лишен чувства юмора. Это замечание будет понятно в дальнейшем.) Как часто бывает, организаторы все перепутали, и в то место, где должен был выступать академик, привезли клоуна, а к тем зрителям, которые пришли на встречу с клоуном, привезли академика. Клоун честно старался всех рассмешить, но у него ничего не получалось: не улыбался никто, а некоторые даже что-то конспектировали. Зато на лекции академика царило веселье. Сам же академик никак не мог понять, что же такого смешного в результатах проводимых им исследований. От отчаяния он даже написал на доске очень сложное дифференциальное уравнение второго порядка в частных производных, на что зал отреагировал взрывом хохота и бурными аплодисментами. Ученый сорвался на крик и стал кричать в зал что-то типа "Объясните же мне, что смешного я говорю?! Почему вы смеетесь?! Это же все очень серьезно! Я приехал с серьезной лекцией к серьезным людям!" В зале — хохот до слез: люди, загибаясь от хохота, шепчут друг другу. "Ну юморист, никогда такого не видел. Во дает!" Бедный ученый покинул зал в слезах под бурные аплодисменты. Зато кривлявшийся и так и эдак клоун так ничего и не добился: люди просто подумали, что он сегодня болен, и у него нервный тик. ...Вот что такое сила установки. Как говорится, комментарии излишни.

Зная все вышеописанное, опытные переговорщики, используют любую возможность, чтобы создать перед переговорами положительную установку на себя. Простой пример. Если вашему клиенту перед тем, как вы его посетите, кто-то расскажет о вас что-то хорошее, вероятность того, что переговоры пройдут успешно, значительно повышается. Вернее, часто это даже переговорами сложно назвать: просто приятная беседа. Обратное тоже верно: если установка на вас будет в силу тех или иных причин отрицательной, про любые логические доводы на переговорах можно забыть: они не помогут.

Установки у людей можно устанавливать, извиняюсь за тавтологию. Так, к примеру, часто некоторые люди специально устанавливают у людей на себя отрицательные установки. Причин может быть много. В том городе, где я вырос, в середине восьмидесятых годов было просто опасно ходить даже по центральным улицам в силу происходящих разборок между различными подростковыми группировками. Часто после этих разборок на асфальте оставалось лежать по несколько трупов. И многие специально "шили на себя" ампулы отморозков, по сути такими не являясь. Понятно почему. Кому охота с бешеным связываться. Точно такой же сценарий поведения выбирают себе некоторые новобранцы в армии для того, чтобы избежать дальнейших издевательств со стороны старослужащих. И надо сказать, у многих это неплохо получается.

Анекдот в тему

Едет деревенский мужик на повозке, вдруг в него врывается крутой джип. Всё летит в кювет: лошадь в одну сторону, мужик в другую, телега в третью. Мужик повезло, жив остался, даже никаких повреждений нет. А лошадь в агонии мечется, стонет. Лежит мужик и думает: интересно, сколько мне этот новый русский из джипа заплатит за лошадь, телегу и увечья? Лежит, ждет. Меж тем новый русский вылез из своего джипа, подошел к лошади, вытащил пистолет и одним выстрелом в голову прекратил ее мучения. После этого подходит к мужику и спрашивает: "А как вы себя чувствуете?" Мужик о деньгах и думать забыл, на колени встал и лопочет "Спасибо, мил человек, очень хорошо себя чувствую".

Небольшое отступление в сторону типологии личностей. Точнее типологии хулиганов разных мастей. Такие "отморозки-распальцовщики" много где наличествуют. И в вузах, и на предприятиях, не исключено, что и в своей конторе пару-тройку таких найдете. Научитесь их распознавать и не бойтесь. Прочитав этот раздел, вы уже наверняка поняли, что они сами специально генерируют такого рода отрицательные установки на себя. ...Чтобы их не трогали. Потому что боятся они много чего в глубине души, а нередко не в глубине, а на поверхности. И им нужно, чтобы про них говорили, "Мол, Петр Петрович наш бешеный, вчера в очередной раз взбесился и натурально загрыз соседского кобеля ни с того ни с сего. Ага, вместе с хозяином. Сначала кобеля, потом хозяина. Кобель скончался на месте, хозяин до сих пор в реанимации". Потому что агрессор такого сорта почти всегда представляет собой редкостного труса. И любая агрессия против него, более сильная, чем им демонстрируемая, меняет таких людей до неузнаваемости. Не один раз видел, как только что брызгающие слюнями и угрожающие "всех порешить" люди, почувствовав, что их самих порешить могут, становились такими милыми и обходительными... Поверьте, действительно серьезные люди, которые могут серьезно вдарить (неважно, кулаком или словом), просто так ни на кого не бросаются. А наоборот производят впечатление вполне добродушных и мирных людей.

Примечание

Если вам когда-либо посчастливится познакомиться с профессиональным снайпером, вы удивитесь, что у них, как правило, очень добродушные лица и они никогда не размениваются на эмоции.

Точно так же, действительно от природы взрывные люди, как правило, вздорность свою и бешенность скрывают, как только могут. И иногда скорее предпочтут получить "по фейсу" или убежать от драки, чем в нее ввязаться. Потому что знают: "сорвет с катушек", таких дел натворить могут...

И еще об отрицательной стороне установок. Установка многих руководителей на то, что их подчиненные ворюги, не хотят работать, а только зарплату хотят получать, действительно отбивает у подчиненных охоту работать. Во многих организациях, в которых мне доводилось бывать, из-за этого разваливались целые коллективы.

Убеждение с игрой на некоторых слабостях

Многим из нас присущи те или иные слабости, играя на которых, нами можно сравнительно легко управлять. Рассмотрим некоторые из них.

Неуверенность в себе

Неуверенному человеку сложно сказать "нет". Поэтому стратегия его убеждения, как правило, сводится к следующему приему: ему нужно убедительно продемонстрировать, что ничего страшного от того, что он согласится с вами, не произойдет. Дайте ему возможность не говорить "нет", и он никогда этого не скажет.

Медлительность

Для того чтобы медлительный человек согласился с вами, его нужно поставить в условия, в которых нужно быстро принимать решения и искать быстрый ответ. Обдумать свое решение он не успеет и будет вынужден согласиться с вами.

Анекдот в тему

- А что это там за толпа людей стоит?
- Это эстонцы бегут.

Тщеславность

Все мы в той или иной мере тщеславны. Поэтому под словом "тщеславность" будем здесь понимать тщеславность, скажем так, неприкрытую. Когда человек начинает любить себя совсем уж сильно. С такими людьми работать достаточно просто. Достаточно им слегка польстить, а потом еще слегка и еще... А потом коньячок хорошенький, или что он там любит. И все. Он — ваш. И можете управлять им как хотите. Хоть так же, как управляла придворная свита поздним Брежневым и Ельциным.

Примечание

Вообще, всем нам надо в этом смысле быть начеку. Потому что все мы к себе относимся, скажем так, очень неплохо. И это очень хорошо! Но на удовлетворении чувства собственной значимости можно легко попасться. Говорите, что вас это не касается? Вы лезть за километр без бинокля распознаете? А не надо километра. Можно подстроить все так, что вы как будто бы услышите, как

ваш сотрудник вас хвалит. За глаза. То есть в ваше отсутствие, а вы это услышали "совершенно случайно". Наверное, после такого вы все-таки будете относиться к этому человеку с положительной установкой. Говорите, даже на это не клюнете? Рад за вас. Или не рад. Не знаю. А вдруг не подстава, а человек и вправду хорошо к вам относится? А вы со своей подозрительностью таким образом всех распугать можете. Есть, конечно, люди, которых ничем не проведешь, которые просто чувствуют человека, и все тут. Глянут раз-другой и все: как просканировали. Но таких не много.

Азартность

Азартного человека очень легко сподвигнуть на какое-то дело. Надо только потакать его азарту и как бы "заряжаться" вместе с ним.

Невежество

Невежество присутствует у каждого из нас. Все мы невежественны в вопросах, лежащих достаточно далеко от нашей профессиональной деятельности. И, увы, этот факт кто только не использует себе во благо.

Эффект ореола или эффект обобщения

Для того чтобы было понятно, что подразумевается под этим эффектом, приведем простой пример. Очень нередко наши успехи или, что хуже, неудачи в какой-либо области деятельности пролонгируются на другие области. Вот это и есть эффект ореола. Если вы преуспели, скажем, в науке, то люди почему-то склонны думать, что вы обязательно должны преуспеть и в любой другой области, скажем, в бизнесе. Что, вообще говоря, совершенно не так, и в жизни полным-полно подтверждений этому. Совершенно не обязательно, что бывший успешный спортсмен станет не менее успешным бизнесменом. Более того, наблюдается скорее обратная тенденция. Так же неверно, что командир-военный сможет с таким же успехом командовать фирмой, с каким он командовал полком. Может и сможет, а может и нет. Те примеры, что я знаю, чаще говорят о том, что нет. И обратное, конечно, верно: глупо думать, что глава корпорации из тысячи человек сможет командовать армейской дивизией. Очень многие из нас попадают на этот эффект на выборах, когда голосуют за известных и добившихся успеха в той или иной деятельности лиц (актеров, военных, артистов, ученых и т. д.), считая, что в Думе они будут такими же профессионалами, какими являлись по своему предыдущему месту работы. Однако несмотря на многочисленные опровержения эффект работает. Потому как является следствием другого эффекта, суть которого заключается в том, что **люди очень не любят менять своих установок.**

Примечание

Понятно почему. Дело в том, что смена установки равносильна признанию в собственной ошибке, чего, естественно, многие не любят.

И, наконец, приведу один из самых распространенных примеров эффекта ореола, с которым все наверняка сталкивались по учебе в вузе. Для того чтобы стать отличником, совершенно не нужно в прямом смысле слова "на отлично" учиться все годы. Достаточно отлично закончить первый курс. Максимум — второй. А дальше оценки будут ставиться "автоматом". И обратное, увы, верно. Если вы сначала сдавали все на троечки, то потом выскочить из амплуа "троечника" крайне сложно. Посмотрит преподаватель зачетку, а там... И ставит то же самое независимо от знания вами предмета.

Примечание

Дорогие читатели, из тех, кто обучается в каком-либо учебном заведении! Если вы попались в такую ловушку и пытаетесь что-то изменить, сделайте одну простую вещь. Закрепите скрепкой предыдущие листки зачетки. Преподавателю будет неудобно при вас возиться со скрепкой, чтобы посмотреть, что вам там понаставили его коллеги, и ваши шансы на объективность заметно повысятся.

Эффект ореола может распространяться не только на одного человека, но и на ту или иную группу людей. Простой пример: очень часто одни и те же школьные учителя, учившие одновременно или с промежутком во времени двух детей из одной семьи, относятся к ним одинаково. То есть, если, к примеру, старший брат не знал английского, а младший потом попал к той же учительнице, то с немалой долей вероятности она ему автоматически будет ставить плохие оценки, даже если он знает язык лучше нее. (Увы, при нашем нынешнем низком уровне школьного образования нередко встречается, когда ученик знает предмет лучше учителя.) Точно так же, если вы работали раньше в компании, о которой шла плохая слава, против вас часто априори имеется отрицательная установка по принципу "все они там такие (плохие)".

Примечание

Однажды во время разработки сайта одной из компаний мы работали в паре с коллегами из другой студии: они делали дизайн, мы программирование. Коллеги с работой не справились. А ранее в разговоре с руководством компании они упоминали, что с нами знакомы. Так руководство компании сначала и нас отстранило от разработки программных блоков, хотя мы никаких поводов для такого решения не давали; более того, все они были уже практически готовы. Сработал эффект ореола: если эти не справились, то и их знакомые тоже не справятся. Хотя никаких логических или иных здравых соображений для таких выводов совершенно не было. Завершилось все нормально, мы и по сей день успешно сотрудничаем, а тогда я пошел к руководству и в достаточно жесткой

форме объяснил, в чем они не правы. Для начала обругал на чем свет стоит ту компанию, которая "завалила" проект, сказав, что страдаю по их вине уже не первый раз. В общем, разыграл жертву.

Эффект близости

Эффект близости по сути представляет собой продолжение эффекта ореола. Попросту говоря, суть эффекта близости заключается в том, что то, что находится рядом с вами, переносится на вас. Эффект этот часто используется всевозможными пиарщиками, политтехнологами и прочими. Помню, во время избирательной компании в Думу для того, чтобы снизить рейтинг фракции, возглавляемой Е. М. Примаковым, во многих газетах его фотографии помещали рядом с какими-то непрезентабельными снимками, к которым он, конечно, не имел никакого отношения. В одной газете чуть ниже фотографии Примакова была фотография кучи навоза, в другой — помойки, в третьей — расчлененного трупа. И эффект работал. И на телевидении делалось то же самое. После речей "нужного" кандидата шел какой-то умиротворяющий и жизнерадостный репортаж (к которому кандидат не имел никакого отношения, но в подсознании избирателей откладывалось, что он как бы к этому причастен), а после выступления "ненужных" — криминальная хроника. Думаю, смысл понятен.

Важность первого впечатления

Известно, что впечатление о человеке складывается в первые минуты общения с ним, а затем практически не меняется. Многочисленные эксперименты показали, что, к примеру, на собеседованиях (когда экзаменатор и экзаменуемый видят друг друга впервые) большинство экзаменаторов составляет свое мнение о человеке в первые 3—5 минут. А все другое время он это составленное мнение подтверждает: либо заваливает, если мнение плохое, либо, наоборот, помогает, если вы ему понравились. А некоторые исследователи приводят данные и того страшнее: 30—50% своего мнения о собеседнике люди составляют в **первую** минуту разговора. Поэтому на переговорах ли, на собеседовании особое внимание уделяйте именно первым минутам разговора. Потом можно чуть расслабиться.

Точность — вежливость королей

И долг всех людей. Этот афоризм, принадлежащий Людовику XIV, все знают, но, увы, не все соблюдают. А между тем сложно переоценить, насколько страдает наш имидж, когда мы опаздываем. Потому что ожидающий думает

про опаздывающих примерно так: либо он настолько неорганизован, что не может прийти вовремя, либо он демонстрирует свое наплевательское ко мне отношение. Согласитесь, оба случая достаточно неприятны. На занятость пенять не имеет смысла: заняты все. И, как правило, самые суперзанятые почти всегда приходят на встречи вовремя. А если вы все время, всегда и везде хронически опаздываете, обратитесь к разделу *главы 10 "Управление временем"*, так как, судя по всему, у вас имеет место быть хроническая непроработанность отношений со временем — болезнь, которой переболели многие, и от которой обязательно надо излечиваться.

Примечание

А уж если опаздываете куда-то, так предупреждайте об этом ваших партнеров. Не бойтесь в этом случае позвонить и сказать, что опаздываете, по каким причинам и в какое время будете. Это гораздо лучше того, когда ваш партнер ждет вас в неизвестности относительно того, когда вы приедете и приедете ли вообще. Кроме того, извинившись за опоздание во время телефонного разговора, вы можете уже не извиняться при встрече, что тоже неплохо, так как начинать встречу с извинений и оправданий это изначально ставить себя в невыгодную позицию. Но лучше, конечно, вообще не опаздывать. А если заранее знаете, что вероятность опоздания велика (встреча в другом городе, предыдущая ваша встреча может затянуться и т. д.), так предупредите об этом партнеров, указав при этом не точное время прибытия, а интервал. В конце концов, идите на хитрости. Если уж опаздываете минут на 20, так ...опоздайте еще на 10, заехав предварительно в какую-нибудь всеми позабытую улочку недалеко от места встречи и позвонив оттуда клиенту со словами извинения за опоздание и просьбой рассказать, как из этой самой улочки выбраться. Показав тем самым, что вы вроде приехали-то вовремя, но вот заплутали слегка. Это воспринимается гораздо менее болезненно, чем опоздание "просто так". И таких способов можно придумать много. И придумывайте. Только не допускайте того, чтобы о вас в связи с опозданием подумали в невыгодном свете.

Давайте делать паузы в словах

Как в песне А. Макаревича поется... Искусство делать паузы — очень важное умение для переговорщика. Тот, кто филигранно может "работать с паузами", выигрывает намного больше не обладающих этим искусством. К примеру, когда вы слышали условия сделки от вашего партнера, не торопитесь сказать "да", даже если вас эти условия вполне устраивают. Потому что, если вы это "да" выпалите, то клиент может подумать, что он продешевил. А вот если вы возьмете небольшую паузу, то это будет выглядеть гораздо лучше. Так как пауза как бы свидетельствует о ваших сомнениях, и раздумьях и создает впечатление, что предлагаемые условия — это минимум из того, на что вы можете согласиться.

Вместо резюме или минус эмоции

По сути, большинство рекомендаций этой главы предполагают бесстрастное отношение к работе с клиентом. Как единственно возможный вариант профессионального выполнения данного вида работы. Никаких эмоций. Вам важен только результат. Когда вы уже работаете над проектом, результат — завершение проекта и получение оплаты. Вы не должны испытывать к себе-седнику ни жалости, ни ненависти, ни человеческой привязанности. Беседа с заказчиком — это ваша работа, а вы — машина для ее выполнения. Чувства можно позволить себе потом. При выполнении работы можно позволить себе только два чувства: радости (за хорошо сделанную работу) и злости (на себя — в противном варианте).

Примечание

Чувства в бизнесе и вообще в любом деле вещь крайне опасная. Если хотите, можете себе позволить их дома, на отдыхе с друзьями, но не во время проведения переговоров. При переговорах вам важно максимально улавливать информацию и адекватно на нее реагировать. Чувства этому только мешают. Только в спокойной воде вы можете четко увидеть свое отражение, если на воде даже небольшое волнение — изображение "поплывет". Так и с эмоциями.

Это не значит, что, общаясь с клиентом, вы должны вести себя, словно робот и отвечать механическими фразами. Вы можете выглядеть сколь угодно эмоциональным. Но именно выглядеть, а не быть: эмоции нужно играть, а не испытывать на самом деле. А если испытываете... Ничего кроме вреда это не принесет. Во-первых, как уже говорилось чуть выше, можно пропустить важные детали. Во-вторых, если все воспринимать излишне эмоционально, можно нанести ощутимый вред своему здоровью. Потому что люди встречаются, мягко говоря, разные. А работать только с теми, с кем у вас стопроцентная психологическая совместимость, значит, завалить все дело. Ну и, в-третьих. Вспомните пример с ледоколом. Который монотонно раскалывает льдину. У этого ледокола, как у любого судна, есть капитан. Который, вообще говоря, когда шел работать капитаном на ледокол, знал, что он льдины колет, а не арбузы перевозит. И знал, что в некоторых ситуациях расколоть льдину непросто, а иногда и просто невозможно. А теперь представьте себе, что стоит капитан в рубке и увидел ту льдину, которую надо расколоть. И по этому поводу случился у него нервный срыв. Потому что в глубине души он надеялся, что все-таки будет на ледоколе перевозить арбузы. (Как многие из нас, которые ожидают, что клиенты почему-то априори должны относиться к нам хорошо и подписывать договора сразу, как только нас увидят.) Бегает, значит, этот капитан по рубке и кричит "Ах ты, падла, и откуда же ты тут взялась на мою седую голову!" И понимая, что его мечте об арбузах не сбыться, впадает капитан в полноценное буйство, хватается лом, бежит на нос и, размахивая ломом, выговаривает льдине все, что он о ней

думает. Решив, что этого недостаточно, бежит на камбуз за тарелками и начинает кидаться ими в льдину, в надежде, что она догадается, что ведет себя не очень хорошо, и саморасколется. Ну, или отплывет куда-нибудь, на худой конец, испугавшись капитанского буйства. Глуповато получается, не правда ли? А ведь нередко мы уподобляемся этому капитану, когда ведем переговоры: чуть что не по-нашему, сжимаемся как стальные пружины, глаза стекленеют, на губах пена и пошло... "Как вы смеете мне такое говорить?!", "Что вы себе позволяете?!", "Я никогда больше не переступлю порог этого кабинета!", "Вы просто хам, а не директор!"... Бесимся и хлопаем дверьми... А потом неподдельно удивляемся, чего это мы вдруг разорились? Кто виноват?! А никто. Мы виноваты. Потому что, пока по носу нашего корабля с ломиком бегаем, проклятия изрыгаем, кораблик наш в это время, оставшись без управления, куда-то врзается. И тонет. А мы даже в такой ситуации все бегаем по палубе, кричим, беснуемся вместо того, чтобы "бороться за живучесть вверенного нам плавсредства". "Гады все! Сволочи! Я давно знал, что в этом дрянном мире никому не нужен! И команда — полные отморозки! Глупцы! Подсказать не могли вовремя, что корабли тонут после того, как куда-то на полном ходу врзаются!" И вот уже наш кораблик под водой скрылся, а мы все прыгаем, в воде уже, из последних сил выныриваем, чтобы изрыгнуть свои последние проклятья этому миру.

...Вот что такое неконтролируемые эмоции. И излишне высокое самомнение. Эти двое, эмоции и самомнение, близкие родственники. Всегда вместе ходят. Потому что когда мы "в эмоциях", то видим только себя. Наше очень хорошее о себе мнение. Все остальное для нас в этот момент вторично. Более того, в основе эмоций лежит животное начало, а не разум. Поэтому доверять вести переговоры человеку, не умеющему контролировать свои эмоции, это все равно, что запустить на эти самые переговоры выжившего из ума дикобраза, который всех иглами утыкает и будет таков. ...Ведь чем мы по большому счету отличаемся от животных? Да ничем кроме наличия второй сигнальной системы. Способностью думать, то есть, отличаемся. Способностью контролировать свои эмоции, чего животные делать не могут. И некоторые люди с органическими поражениями мозга, олигофрены, к примеру, тоже не могут. Да, они живут на голых эмоциях. Солнышко выглянуло — радость, дождик пошел — в плач. Снова солнышко — кончил плакать и опять улыбается. А теперь небольшой пример стандартного диалога, который вы можете услышать в различных вариантах где угодно и когда угодно. Представьте себе: два человека. Неважно кто: начальник и подчиненный или партнеры на переговорах. Важно, что один как бы критикует, а второй реагирует.

— Должен сказать, что мне очень нравится с вами работать. (Солнышко светит.)

— Я так рад это слышать! (Радуетя.)

— Но в последнее время вы делаете много досадных ошибок! (Тучка налетела.)

— Кто?! Я?! Да как вы смеете такое говорить! Мне — дважды заслуженному, трижды лауреату, руководителю коллектива из 50 человек! (Страдания.)

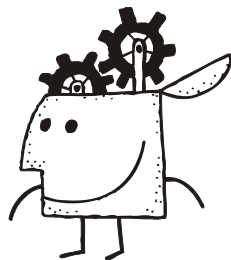
— Да, вы правы. Извините, я, наверное, ошибся. За всеми не уследишь. Наверное, это кто-то из сотрудников напортачил, а вы не уследили. (Снова солнышко.)

— И вы извините, что погорячился. Бывает. (Снова улыбается.)

Вот так мы разговариваем... И еще удивляемся, почему к нам деньги не идут. Понятно, почему... Потому что мы... Потому что мы к разуму обращаемся только тогда, когда уже сил на эмоции не остается. Где-нибудь на больничной койке. После перенесенного инфаркта, трех инсультов и пяти клинических смертей.

Поэтому давайте в переговорах, в бизнесе, да и вообще в жизни не делать ставку на эмоции. Давайте ставить на разум. И тогда все будет не просто хорошо. Все будет замечательно!

Глава 5



Проектирование

Любой проект состоит из трех крупных этапов — проектирования, разработки и тестирования. При создании качественного программного обеспечения основные усилия сосредотачиваются на тестировании системы, по сути, на завершающем этапе создания программного продукта. Однако качественной программа будет только в том случае, если усилия, направленные на повышение качества, будут затрачены не только в конце проекта, но и в начале, и в середине.

Ошибки, совершенные на этапе проектирования, слишком дороги в исправлении, так как на реализацию ошибочных положений затрачены уже значительные усилия. Согласно последним исследованиям, исправление архитектурной ошибки на этапе проектирования обходится в 10—100 раз дешевле, чем в конце работы над проектом. Если спроектировано и разработано приложение регистрации заказов, а заказчику, оказывается, была необходима биллинговая система, никакие тестирования не превратят одно приложение в другое. Исправлять код уже построенной системы также будет неудачным решением — так как приложение изначально совершенно для этого не предназначалось — самое правильное будет выкинуть все наработки и начать работу сначала... Только практически ни один из разработчиков не решится на такой шаг, а заказчик вряд ли согласится оплачивать дополнительное время разработки, если вообще захочет ждать.

Требования

Проектирование системы начинается с выработки требований. Разработчики совместно с заказчиком выясняют, какие проблемы должна решать система и какую функциональность предоставлять — это позволяет заказчику и разработчикам лучше понять друг друга и будущую систему. Часто разработчики не знакомы с предметной областью и им требуется в мельчайших подробностях узнать особенности бизнес-процессов перед моделированием их в

программе. Заказчик со своей стороны часто не знаком с особенностями разработки программного обеспечения и выдвигает требования, сильно усложняющие проект, но, в то же время, отсутствие которых не сильно отразится на работе. Начальный этап проектирования позволяет разработчикам и клиентам лучше понять друг друга.

Замечание

Термин "требования" часто называют "функциональной спецификацией" или "техническим заданием".

Кроме того, часто без формулировки требований невозможно назвать точную сумму и сроки разработки программы. Следует учитывать, что заказчик не является экспертом в области программного обеспечения. Он не разбирается в программе, которую хочет заказать, и не может знать, как она будет выглядеть. Поэтому требования к программе по мере разработки у него меняются всегда — это закон. Протесты и угрозы в сторону заказчика здесь бесполезны — это нужно принимать как неизбежность. Связано это в первую очередь с тем, что в отличие от других инженерных сооружений программа не материальна и у заказчика складывается чувство "легкости" внесения изменений. Когда заказчик имеет дело с домом, он понимает, что перестройка балкона, а тем более фундамента зачастую является сложной задачей, требующей больше усилий, чем построение дома. В случае программы это не очевидно зачастую даже для самих разработчиков.

Часто разработчики впадают в ярость при изменении требований в конце проекта. Их можно понять, внесение изменений после разработки приложения стоит в десятки и сотни раз дороже, чем при кодировании. Исследования IBM показали, что при разработке среднего проекта требования изменяются на 25%, на которые уходит 70—85% объема работ. Однако если вы замораживаете требования в самом начале, на самом деле вы не собираетесь реагировать на потребности заказчика.

В таких условиях выработка предварительных требований особенно важна — это позволяет не только разработчику понять, как следует разрабатывать программу и что в нее будет входить, но и предоставить заказчику документ с описанием функциональности программы и ее стоимости. В дальнейшем, когда у заказчика меняются требования к программе, этот документ поможет обоснованно объяснить, почему следует повысить стоимость разработки или увеличить сроки разработки.

Замечание

Клиента следует подготовить к тому, что значительное изменение функциональности во время разработки может повлечь увеличение стоимости и времени разработки, что назначенная цена является предварительной. Иначе такое объявление в конце проекта может стать неприятным сюрпризом. Следует учи-

тывать тот факт, что клиент вряд ли пойдет на увеличение стоимости (если это не внутрикорпоративный проект) и скорее согласится на увеличение сроков разработки. Самым лучшим выходом является предугадывание объемов работ, связанных с изменениями, и включение их в стоимость разработки.

Проектирование начинается с краткой формулировки задачи, которая описывает ее одним предложением. Если вы моделируете в программе маршруты движения автобусов города, то формулировка может выглядеть следующим образом: "Нужно смоделировать, как автобусы перевозят людей в городе, при этом необходимо минимизировать простой автобусов". Для корпоративного сайта такая формулировка может выглядеть следующим образом: "Необходимо создать сайт, через который компания будет предлагать свои услуги по сети Интернет". Для операционной системы краткая формулировка может выглядеть так: "Необходимо создать среду, позволяющую запускать программы и взаимодействовать с внешними и внутренними устройствами компьютера". Для любой задачи можно сформулировать краткий тезис, который будет полностью описывать ее; если вы не можете сформулировать такой тезис, скорее всего вы не до конца понимаете, что должна делать система.

Важной особенностью требований является то, что они должны формулировать проблему, а не предлагать способы ее решения. Например, формулировка "Необходимо обеспечить обратную связь с покупателями на сайте" является хорошей формулировкой проблемы. Однако тезис "Необходимо поставить на сайте Web-приложение «Гостевая книга»" — плохое определение проблемы, так как является, по сути, решением. Этого допускать нельзя, так как заказчику для обеспечения обратной связи может больше подойти форма отправки почтового сообщения с сайта или Web-форум. Навязывание своего представления решения проблемы с самого начала может привести к тому, что, договорившись о создании "гостевой книги", требования заказчика по мере создания приложения могут возражать: сначала нужно будет добавить регистрацию и авторизацию посетителей, потом смайлики, затем несколько разделов, потом линейное и ступенчатое представление ответов, и в результате окажется, что вы разрабатываете форум. При этом база данных будет спроектирована под гостевую книгу, и в договоре с заказчиком стоять сумма разработки гостевой книги. Явные требования помогают гарантировать, что функциональность системы определяется пользователем, а не разработчиком. Если явных требований нет, разработчик все равно вынужден будет их вырабатывать во время программирования. Явные требования позволяют не гадать, чего хочет пользователь.

После того как краткое описание сформулировано, его следует расшифровать примерно на двух листах с описанием функциональности. Именно здесь указывается, что будет входить в систему и какой сервис она будет предоставлять, а какой нет. Не следует бояться писать здесь, что в систему не будет входить — это позволит исключить двусмысленности, когда заказ-

чик считает, что разработка данного блока подразумевает авторизацию посетителей, а вы нет. Лучше явно написать, что "система не поддерживает авторизацию" — это позволит заострить на этом внимание заказчика. Если его это не устраивает, можно с самого начала предусмотреть в системе авторизацию, а не впихивать скелет в мясо после того, как система разработана.

Описание функциональности не должно включать в себя низкоуровневые детали, например, детализацию, что "форум" будет хранить информацию в файлах или в базе данных. Здесь описывается только функциональность, какие операции может совершать пользователь, какие отчеты предоставляет система, в каком виде: on-line-доступ или отправка по e-mail. По возможности следует избегать специальных терминов, они допустимы только в том случае, если разрабатывается или модифицируется инструментарий — компилятор, утилита генерации кода и т. п. В подробной расшифровке функциональности ни в коем случае не должно быть описания структуры классов, служебных файлов, используемых протоколов и т. п.

Архитектура проекта

После того как сформулированы требования к программному проекту, т. е. определено, как программа должна будет выглядеть и что с помощью нее можно будет сделать, приступают к созданию архитектуры. Архитектура — это высокоуровневый каркас, состоящий из деталей проекта. Основная цель архитектуры — разбить проект на части (компоненты), над которыми отдельные разработчики или команды могли бы работать независимо друг от друга. При создании архитектуры выделяют несколько абстрактных уровней:

- ☐ вся система;
- ☐ подсистемы;
- ☐ пакеты и классы, из которых состоит подсистема;
- ☐ методы и данные, из которых состоят пакеты и классы;
- ☐ вспомогательные методы.

Число абстрактных уровней зависит от типа приложения. В небольших приложениях можно оперировать сразу классами, в больших системах может понадобиться вводить несколько видов систем (например, несколько видов серверов). Подсистемы обычно реализуют достаточно большие части конечной программы, например:

- ☐ система взаимодействия с пользователем (реализованная как совокупность Web-страниц или окон);
- ☐ система администрирования (реализованная как совокупность Web-страниц или окон);

- доступ к данным (в файле или базе данных);
 - реализация бизнес-правил (вычисления)
- и т. д.

Каждый компонент должен выполнять одну задачу, если он выполняет несколько задач — он является кандидатом на деление. Важно, чтобы компоненты как можно меньше знали о внутреннем устройстве других компонентов — идеальными являются случаи, когда компонент можно взять из одного проекта и перенести в другой, при этом для его нормальной работы не требуется других компонентов. Достигается это за счет четких интерфейсов, в обход которых компоненты действовать не должны. Это означает, что если любые обращения к базе данных реализованы через специальный класс, все обращения в системе должны проходить через этот класс, как бы велико ни было искушение обратиться к базе данных напрямую. Изменения в структуре базы данных затронут только компонент, предназначенный для работы с базой данных. Если у вас помимо единого компонента по приложению разбросано несколько прямых обращений — придется проверять все компоненты, причем наверняка до конца исправить все запросы не удастся. Если вместо того, чтобы в обработчике диалогового окна вызывать методы класса, реализующего бизнес-правила, этот компонент читает данные непосредственно из элементов управления диалогового окна, изменение интерфейса или просто переименование элемента управления будет отражаться на всем коде, приводя к множеству трудноулаживаемых ошибок.

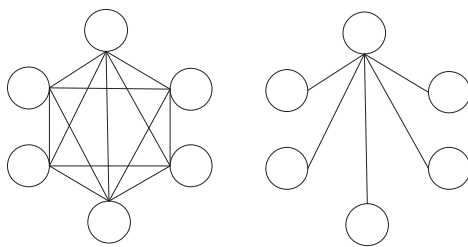


Рис 5.1. Чем меньше связей между подсистемами, тем лучше

На рис. 5.1 представлена система, разделенная на шесть подсистем. Чем меньше связей между отдельными подсистемами — тем надежнее система и тем проще ее сопровождать, так как общая сложность системы уменьшена. Когда каждая подсистема взаимодействует с другими подсистемами, программисту необходимо знать особенности их работы и как отразятся вносимые изменения на других частях системы. Когда же такое взаимодействие сведено к минимуму, программисту необходимо удерживать меньше информации в голове, и вероятность совершения ошибок уменьшается. Это вовсе

не означает, что система должна быть построена по принципу звезды, а не сети. Часто это просто невозможно из-за системных требований и слишком медленной работы, но чем меньше связей в системе, тем лучше.

Если система администрирования сайта реализована в виде отдельного каталога, то все административные функции должны выполняться только в этом каталоге, так как если понадобится защитить систему администрирования паролем, это будет проще сделать, когда вся она будет сосредоточена в одной директории, а не разбросана по всему Web-приложению.

Замечание

Ошибка в архитектуре программы обходится так же дорого, как и ошибка в требованиях. Изменение требований по мере создания программы часто влечет изменение архитектуры — опытный проектировщик старается создать запас, чтобы не позволить изменениям обрушить программу в шторм неуправляемой разработки.

Обычно архитектура определяет не только классы и их ответственность, но и время жизни объектов.

После того как подсистемы определены, следует разбить их классы или пакеты, если язык программирования не поддерживает классы. На данном этапе необходимо определить, какие методы будут использоваться для взаимодействия с другими классами — т. е. определяется открытый интерфейс классов.

После этого приступают к детальной проработке каждого класса, так как открытая реализация определена, разрабатываются закрытые методы. Впрочем, данный этап часто опускают и оставляют до этапа кодирования, чтобы не перегружать проектную документацию.

Архитектура не обязательно должна описывать все компоненты в мельчайших деталях. При разработке архитектуры следует придерживаться правила 80/20, согласно которому следует описывать 20% системы, которые будут описывать систему на 80%. Ни в коем случае архитектура не должна описывать низкоуровневые детали, т. е. при помощи каких алгоритмов реализовывать ту или иную функцию и какие члены должны входить в тот или иной класс. Задача архитектуры — разбить задачу на более или менее независимые части.

Спецификации

После того как архитектура программы готова и каждый из участников знает и понимает, какую часть работы он будет осуществлять, следует создать спецификации, в которых описываются требования к коду, которые должны выполнять разработчики, например, выбор имен для переменных,

констант, классов, расстановка фигурных скобок, использование сторонних библиотек. Спецификации являются рабочим инструментом конечного разработчика.

Замечание

Главной задачей архитектора проекта является слежение за выполнением архитектурных решений. У конечных разработчиков, особенно когда проект выходит на финишную прямую, возникает соблазн срезать углы и отойти от архитектурных требований.

Если в одном блоке текущее число заказов называется `total`, в другом `number`, а в третьем `orders`, то это не приведет ни к чему, кроме того, что программистам, сопровождающим код, придется каждый раз исследовать код для того, чтобы выяснить, как в данном блоке именуется переменная. Спецификации позволяют предупредить такую ситуацию и потребовать называть переменную в одном стиле. Ни один проект, в котором участвует более одного человека, не обходится без спецификации, формальной (на бумаге) или неформальной (устная договоренность).

Замечание

Даже при написании данной книги авторы используют спецификации, выработанные для авторов издательством "БХВ-Петербург", которые регламентируют заголовки глав и разделов, оформление текста и рисунков. Благодаря этому все книги одинаковы в оформлении.

Планирование времени

Время, которое отводится на проектирование системы, зависит от особенностей проекта. По данным МакКоннела проектирование среднего проекта поглощает 10—20% усилий и 20—30% процентов времени.

В любом случае не стоит зарываться в проектирование с головой. Следует помнить о правиле 20/80, согласно которому проектная документация должна описывать 20% системы, которая будет осуществлять 80% работы.

При нестабильных требованиях проектирование обычно рассматривают как отдельный проект, который позволит оценить время для оставшейся части проекта.

Пренебрежение проектированием обусловлено двумя причинами: неспособностью к проектированию разработчиков, которые сводят процесс проектирования к кодированию, и невыделением времени под этот процесс. Часто на проектирование совершенно не выделяется время, так как заказчиков/руководителей раздражает сам вид недельной или месячной задержки, в ходе которой программисты не пишут ни строчки кода, а бегают и ведут

переговоры с заказчиком. Слишком большой соблазн засадить "этих тунеядцев" за компьютеры сразу. Доказать, что проект в результате этого будет выпущен с задержкой и гораздо худшего качества, очень сложно, поэтому нередко часть команды приходится бросать на создание видимости работы, поручив им разработку инструментария, пусть даже не имеющего отношения к проекту.

Создание программного обеспечения является инженерной задачей, которая проходит через все этапы, в том числе и проектирование. Никто не отважится строить дом без предварительного планирования — в лучшем случае средства и время будут потрачены впустую, в худшем случае это создаст угрозу человеческим жизням. Точно так же и в программировании пренебрежение проектированием приводит к пустой трате средств и времени на разработку неработающей программы или программы, сопровождать которую невозможно из-за множества изъянов.

Замечание

Тщательное планирование подразумевает полную проработку возможности расширения программного обеспечения. Канализация и водопровод закладываются до постройки дома, а не после того, как он построен. Впрочем, в Российской Федерации принят другой подход, заключающийся в строительстве дома с последующим перерыванием окрестностей... Возможно, российские программисты, наблюдая эту картину, и пренебрегают проектированием.

При проектировании программного обеспечения существуют две крайности. В первом случае разработчики пренебрегают проектированием, скрывая свою неспособность к проектированию отговорками, что в небольшом проекте проектирование — лишь пустая трата времени. В конечном счете и большие, и небольшие плохо спроектированные программы приходится переписывать с нуля два, три, а то и большее число раз.

Во втором случае, стараясь избежать повторной разработки, разработчики затрачивают массу времени, пытаясь понять проект и выработать наиболее оптимальную архитектуру. Разрабатываются мельчайшие подробности и когда половина времени отводится на проектирование, проект все равно заканчивается неудачно или требует повторной разработки из-за многочисленных подводных камней, которые открылись лишь при кодировании и сборке проекта.

Следует помнить, что ряд задач просто не поддаются анализу на этапе проектирования. Во многих инженерных областях, например, в химической промышленности, часто прибегают к построению промежуточных заводов, чтобы правильно масштабировать крупное производство — перенос реакции с 1 грамма на несколько тысяч тонн — задача не тривиальная и часто не поддается моделированию. Точно так же как самостоятельная разработка

программного продукта и разработка его силами десятка, сотни или тысячи программистов требуют совершенно разных подходов.

Замечание

Такие задачи часто называют "грязными задачами", т. е. задачи, которые можно решить только путем частичного или полного решения. То есть задачу нужно решить один раз, чтобы получить представление о том, как ее решать, и еще один раз для того, чтобы решить правильно.

Как правило, хороший разработчик понимает, что перед ним стоит "грязная задача", и предпринимает действия, направленные на ее решение. Например, выделяет группу разработчиков, которые исследуют ситуацию, создавая промежуточный проект, основная задача которого состоит в проверке работоспособности идеи, лежащей в основе проекта. Учитывая это, следует планировать дополнительное время на такие исследования.

Как отмечают многие исследователи, оценка времени техническими специалистами, которые ранее не работали с текущей командой разработчиков хотя бы над двумя проектами, редко бывает адекватной. Зачастую эксперты, участвующие в оценке времени проекта, испытывают давление руководства, заказчиков, конкурентов и сильно занижают оценки. Надежных методик оценки времени проекта нет, так как основным рабочим инструментом являются люди. Если плохое настроение рабочего, занимающегося физическим трудом, может приводить к снижению производительности в два раза, то для программистов эта величина может достигать нескольких порядков. Настрой команды, политические игры, семейные неурядицы очень сложно прогнозировать, но это нужно делать, если требуется более или менее близкая к реальности оценка. Без анализа конкретного окружения можно говорить лишь о тенденциях. Например, чем больше команда и больший срок разработки программы, тем более вероятно, что срок сдачи будет сорван.

Сложность

Одной из причин неудачного завершения проектов является их сложность, которую зачастую невозможно охватить усилиями одного человека. Основная задача при проектировании программ состоит, по словам Дейкстры, в организации работы таким образом, чтобы можно было безопасно работать с их отдельными фрагментами. Попытка охватить проблему целиком может привести к излишнему усложнению процесса программирования, когда изменение одной части программы будет приводить к неконтролируемым изменениям в другой части программы. Несколько несложных фрагментов понять проще, чем один сложный. Важно не только разбивать проект на

несколько частей, но и чтобы эти части были слабо сопряжены друг с другом, т. е. изменение одной части не отражалось на другой или отражалось минимально.

Многочисленные исследования программных проектов показали, что плотность ошибок в программах почти не зависит от языка программирования и зависит только от квалификации программиста. Это означает, что чем меньше строк кода будет содержать программный проект, тем меньше будет число ошибок. Постоянное увеличение уровня абстракции языков программирования преследует именно эту цель — снижение числа строк в программных проектах и выражение операций на более высоком уровне абстракции.

Замечание

Причем, как показал в свое время Брукс, полностью автоматизировать процесс создания программного обеспечения никогда не удастся. Технологии создания программного обеспечения могут отмирать, но на смену им будут приходить другие, для управления которыми все равно потребуются программисты.

Другим способом снижения сложности является разделение сложного программного проекта на более простые части. Это является основной идеей для улучшения качества программного проекта. Реализация может принимать несколько форм.

1. Разбиение задачи на несколько взаимодействующих программ. Этот подход характерен для UNIX, где множество мелких программ, решающих каждая свою задачу, объединяются для решения одной большой задачи.
2. Использование параллельных программ, одновременный запуск нескольких однотипных процессов, выполняющих одну и ту же задачу. Данный подход особенно оправдан на многопроцессорных станциях и в кластерных системах.
3. Клиент-серверный подход, связан с разбиением программы на две глобальные части — сервер, принимающий запросы и отсылающий по ним ответы, и клиент (или множество клиентов), посылающий запрос серверу и получающий ответ. Использование клиент-серверного подхода целесообразно при распределенном характере приложения, когда требуется его выполнение на нескольких машинах локальной или глобальной сети. Также такой подход применим для приложений, предоставляющих одновременный доступ множества клиентов к одному ресурсу, например, в базах данных, сетевых проектах или графических серверах, вроде Window X.
4. Объектно-ориентированный подход. Программа часто представляет собой монолитное образование, борьба со сложностью в котором происходит за счет иерархической организации кода внутри проекта. Про-

граммист оперирует не низкоуровневыми деталями, а высокоуровневыми абстракциями.

5. Создание мини-языков. Сложность частично перекладывается на плечи пользователя. К мини-языкам обычно относят узкоспециализированные языки программирования, вроде SQL, регулярных выражений, макросов программных комплексов.

Для уменьшения сложности следует прибегать и к другим приемам, например, отдавать предпочтение текстовому формату по сравнению с бинарным, так как человек лучше воспринимает текстовую информацию и может воспользоваться эвристическим подходом для оптимизации или просто положиться на свою интуицию. Этот совет относится главным образом к файлам и протоколам. Конечно, использование бинарного формата приводит к увеличению скорости и уменьшению объема памяти. Однако не следует забывать о том, что требования к минимизации объема памяти и увеличению скорости были актуальными в прошлом веке. В настоящий момент компьютерное время и память стоят слишком дешево, а время программиста слишком дорого, и этот разрыв только увеличивается каждый год. Поэтому лучше сэкономить на времени разработки программы и на ее сопровождении другими программистами, чем на эфемерной выгоде от скорости выполнения программы и занимаемого файлами дискового пространства.

Замечание

Начинающие разработчики часто имеют следующий комплекс: пытаюсь доказать себе и окружающим, что они являются настоящими программистами, они намеренно усложняют код и используют "хитрые" алгоритмы. Результатом являются не доведенные до конца проекты, которые настолько усложняются, что ни один человек не способен понять принципы их работы. Чем экзотичнее система, тем сложнее будет другим разработчикам в ней разобраться; по возможности, следует применять стандартные средства и методики.

Даже кампания Microsoft, характеризующаяся традиционным использованием закрытых двоичных форматов, на момент написания книги приняла решение отказаться от использования двоичных форматов для хранения документов Word и Excel в виде бинарных пакетов и перейти на открытый формат хранения в виде XML-файлов.

Если необходимость сжатия файлов все же существует, разумно сжимать уже текстовые файлы, как это реализовано в OpenOffice — XML-файлы этого пакета могут быть подвергнуты сжатию при помощи стандартных UNIX-упаковщиков `gzip` или `zip`. Налицо классический UNIX-подход — каждое приложение выполняет одну задачу, а конечный результат достигается их комбинацией.

Сложность проекта многогранна, она складывается не только из сложности разработки, но и сложности сопровождения. Создать и отладить сложный

код еще можно, но добавление в него новых функций, причем зачастую разработчиками, которые не принимали участие в разработке, может стоить очень дорого и по затратам, и по времени, если при проектировании не учитывалась возможность его преобразования.

Часто в систему закладываются дополнительные возможности, например, расширения, которые позволяют легко модифицировать проект или повторно использовать код. Это является хорошей практикой, но следует помнить, что разработка такого кода требует больших затрат по времени и усилиям. Ничто не дается просто так — если код спроектирован с возможностями повторного использования кода, подключения внешних модулей или разделения дизайна и кода, в создание этих особенностей следует вложить усилия. Если потом данные возможности использоваться не будут — усилия и время пропадут даром.

Масштабирование

Все знают, что, просидев без сна за компьютером двое суток, человек способен своротить горы. Есть информация о том, что большие программные проекты разрабатываются со скоростью 300—1000 операторов в год на человека. Несмотря на то, что это отвечает на вопрос о затратах на проекты, в которых участвуют десятки тысяч программистов, такое положение вещей часто приводит в недоумение начинающих разработчиков. Каким образом, можно умудриться работать с такой черепашной скоростью?

Даже несмотря на то, что в эти десятки тысяч входит обслуживающий персонал, который вообще не производит код, низкая скорость связана с издержками взаимодействия между разработчиками.

Когда над программой работает один разработчик — вся информация находится в одной голове, мозг в которой работает со скоростью средней супер-ЭВМ. Разработчик ясно видит архитектуру и общую концепцию программы, пока прошло немного времени, он прекрасно осознает, в каком месте программы следует внести изменения. Однако когда над проектом начинают работать два разработчика встает проблема общения и взаимодействия — другой разработчик, чтобы также хорошо ориентироваться в коде, должен его изучить, т. е. потратить время. Знания, сосредоточенные в голове одного разработчика, не так просто передать другому разработчику, так как скорость обмена информацией между людьми чрезвычайно мала (иногда она снижается еще больше из-за неумения разработчика объяснить свою мысль). Именно поэтому в области создания программного обеспечения так популярны паттерны, которые позволяют одним словом объяснить идею, на объяснение которой может уйти час.

Замечание

Именно поэтому в последнее время приобретает популярность парное кодирование или кодирование в стиле XP, когда над кодом работают одновременно два человека — это позволяет максимально быстро распространять идеи и знание о коде среди большой команды.

При построении кластеров или многопроцессорных ЭВМ две машины в кластере или два процессора в машине работают с гораздо меньшей производительностью по сравнению с тем, когда они работают отдельно. Часть компьютерного времени должна тратиться на коммутацию с другими участниками вычислений.

Замечание

В кластерах и многопроцессорных ЭВМ рано или поздно наступает момент, когда введение в систему новых участников только снижает производительность всей системы.

Скорость обмена информацией между людьми гораздо меньше, чем у машин, так как для четкой передачи всей информации одному человеку необходимо составить документацию, а другим участникам разработки прочитать ее. Создание документации можно рассматривать как отдельный и независимый проект, и ведение собственной документации каждым участником проекта не всегда является приемлемым решением. Кроме того, зачастую оно лишнее, так как код сам является прекрасным способом документации. Однако для того, чтобы участники проекта понимали друг друга, нужно не только принимать спецификации по кодированию, но добиться того, чтобы каждый участник проекта выполнял эти требования. Таким образом, нам необходима контролирующая комиссия, которая будет проводить ревизию кода на предмет его соответствия спецификации. Такая комиссия может состоять либо из отдельно нанятых людей, либо осуществляться за счет времени самих разработчиков. То есть помимо того, что каждый разработчик будет расходовать время на разбор кода другого разработчика, потребуется дополнительное время либо на комиссии, либо на совещания.

Большие проекты, такие как программное обеспечение кораблей или операционные системы, часто невозможно пересобрать каждую минуту для того, чтобы посмотреть как внесенные изменения отразились на программе, так как компиляция релиза может занимать от нескольких часов до нескольких суток. Поэтому часто прибегают к системе тестов, которые эмулируют работу участка кода в реальном проекте. Таким образом, на плечи разработчиков ложится обязанность создания тестов и поддержания их актуальности (т. е. создание новых тестов по мере развития системы).

Тесты не могут полностью устранить ошибки, так как, не видя результатов, разработчик подвергается большому соблазну не тестировать свой код слиш-

ком тщательно. Поэтому код желательно подвергать ревизии на предмет поиска ошибок, для чего нужна дополнительная комиссия.

Таким образом, как только над программой будет работать группа из нескольких человек, перед руководителем встанет проблема либо снижения темпов разработки, либо увеличения штата.

Замечание

Добавление дополнительных разработчиков в начале, середине и конце проекта не является эквивалентными; по исследованию Брукса (Ф. Брукс. Мифический человек-месяц), временные затраты на обучение новых людей и введение их в курс дела отвлекают от разработки программистов, что приводит не к увеличению скорости, а в большинстве случаев к снижению скорости разработки. По его яркому выражению "увеличение числа разработчиков в конце проекта — это все равно, что тушить пламя бензином".

Осознав все сложности взаимодействия, разработчики часто принимают крайнее решение разрабатывать приложения в одиночку (или с группой ассистентов, разрабатывающей вспомогательный код). Однако этот путь редко заканчивается успешно. В первую очередь дело упирается во время, один разработчик или небольшая группа будут разрабатывать код большого приложения слишком долго, за это время проект устареет морально или конкуренты опередят и успеют захватить клиентов. Поэтому руководители проектов предпочитают работать с многотысячной командой, которая грубой силой создаст проект за приемлемые сроки, даже понимая, что управление огромной командой потребует гораздо более серьезных усилий и затрат, чем управление небольшой группой.

Другим подходом, позволяющим небольшой группе быстро продвинуться вперед в разработках, — являются деньги. Нет, дополнительная оплата не увеличит производительность (по крайней мере, надолго), но дополнительные средства позволяют купить готовый код, который зачастую будет много лучше, чем код, разработанный пусть высокопрофессиональной командой программистов, но не работавших ранее в этой области. Самое главное, код можно получить на руки сразу после оплаты и не потребуется ждать несколько месяцев, пока он будет разработан и отлажен.

Многие, воодушевленные тем, что могут быстро создавать программы, похожие на коммерческие продукты, решают, что смогут быстро войти на рынок в одиночку, не принимая во внимание, что программа — это еще не коммерческий продукт. Превращение программы стоимостью в 2500 долларов в коммерческий продукт оценивается в 25 000 долларов и соответствующие временные затраты. Если цель — выход на рынок, это следует учитывать и при планировании сроков сдачи, и при планировании бюджета.

Осознанное и случайное проектирование

Ни для кого не секрет, что программы редко проектируются осознанно. В большинстве случаев мы наблюдаем так называемое случайное проектирование. Причин этому несколько, и почти все из них в своей основе имеют пресловутый "человеческий фактор". Самая главная и основная заключается в том, что в большинстве коллективов просто нет профессиональных проектировщиков, которые бы могли посмотреть на программу не своими глазами, а глазами пользователей. Большинство же так называемых "проектировщиков" проектируют программы исходя сугубо из своих личных предпочтений: т. е. так, чтобы было легче разрабатывать, чтобы быстрее закончить проект, чтобы иметь возможность где-то по пути "блеснуть красивым кодом" и т. д. И хотя объем проектной документации на выходе может быть очень велик, реально выполнена только половина задачи: намечены пути разработки продукта, но "пользовательское проектирование" (а все программы пишутся для пользователей!), как правило, отсутствует вовсе. Поэтому профессионал всегда проектирует, исходя из задач, стоящих перед пользователем, а случайный проектировщик — исходя из задач, стоящих перед ним самим. Такое случайное проектирование может иметь и более, скажем так, низменные причины: замечания и недовольство коллег, страх перед "авторитетами" и т. д.

Не давать пользователю почувствовать себя дураком

Основная и самая главная задача осознанного проектирования, т. е. проектирования, исходя из задач, стоящих перед пользователем, — это не давать пользователю возможности почувствовать себя дураком. Если расклад иной — то такая ошибка в проектировании фатальна. Данное утверждение имеет глубокие психологические корни: ведь мотивы многих поступков людей лежат в удовлетворении чувства собственной значимости. А когда пользователь видит программу, которая у него это чувство отбирает... Понятно, что он с ней сделает. И такая программа не будет иметь широкого успеха у пользователей. Так мало того, что некоторые программисты не отдают себе в этом отчета, многие специально усложняют программу, мотивируя это примерно так: "Работа с программой — это вам не на унитазе сидеть. Здесь интеллект нужен". И при этом самодовольно посмеиваются, упиваясь чувством уже своей собственной значимости. Ясно, что подобный программный продукт не имеет никакого коммерческого будущего. Как, впрочем, и его создатель. Можно сказать более жестко: чем сложнее в использовании программный продукт, тем меньше умственного труда потратил на разработку этого продукта его создатель. Повторюсь, слова о том, что для того, чтобы пользоваться какой-то программой, нужно специальное образование — это

бред от компьютерных и околокомпьютерных "ботаников". Не меньшая глупость, когда в оправдание сложности программного продукта говорят, что "всему виной технологии". Мол, какие технологии, такие и программы. Всему виной всегда люди. И в программировании тоже. Потому что программы пишут люди, а не технологии. Просто очень часто программисты, инженеры и прочие не утруждают себя спуском со своих небес на грешные пользовательские земли и не делают вещи, понятные для пользователя, не являющегося специалистом в инженерных областях. Конечно, можно сделать какой-то "инженерно-программистский" шедевр, но если он не будет принят пользователями, то его место — в музее. В лучшем случае. В худшем — на помойке, как оно часто и бывает.

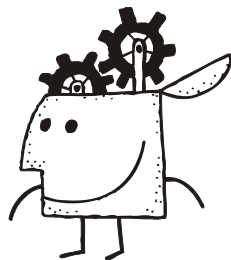
Осторожней с функциональностью!

Ее не должно быть слишком много! Функциональности в программном продукте должно быть ровно столько, сколько нужно, и не функцией больше! К сожалению, очень часто бывает, что функциональность программного продукта раздувается настолько, что, в конце концов, перестаешь понимать, для чего же изначально планировалось его создание. Разработчикам почему-то кажется, что если разработка новой функции мало чего стоит и после ее добавления программа не перестанет работать, то эту функцию обязательно надо добавить. В результате и получаются такие артефакты инженерной мысли, когда "вдруг, откуда ни возьмись" появляются всплывающие календари, выпрыгивающие пиктограммы и т. д. И это еще полбеды. К сожалению, сейчас многие программные продукты, особенно предназначенные для корпоративного использования, настолько сложны, что для того, чтобы их хоть чуть-чуть изучить, надо на пару недель запереться где-то в глухом ауле, принять монашеский образ жизни и заниматься только изучением этой программы.

Примечание

Хотя я всерьез подозреваю, что некоторые фирмы-разработчики идут на такое усложнение осознанно: чтобы кроме денег за программный продукт содрать (другого слова здесь не подберешь) еще деньги и за обучение пользования оным. Классический пример такого на российском софтверном рынке — пресловутая бухгалтерия 1С, где рынок обучения этим программным продуктам реально не намного меньше рынка продаж самого продукта. Смотреть на эту вакханалию и больно, и смешно. Общаясь с бухгалтерами, я им искренне сочувствую, так как они плачут от кучи функций (три четверти из которых ими никогда не используются) и ненужных действий (три четверти из которых программа могла бы сделать и сама). При этом некоторые, действительно нужные бухгалтерам функции в 1С отсутствуют, и банки вынуждены нанимать сторонних разработчиков, чтобы они написали какой-то промежуточный модуль, который будет делать что надо и с этой страхолюдиной взаимодействовать.

Глава 6



Если вы — руководитель

Вообще говоря, работа руководителя программистской организации ничем не отличается от работы руководителя любой другой организации: работа с договорами, встречи с партнерами и клиентами, стратегическое руководство, способы управления компанией — все эти виды деятельности одинаковы для руководителей любых видов организаций. Серьезное различие существует в другом. В масштабах руководства. То есть руководство командой разработчиков на уровне менеджера проекта и руководство компаний это две совершенно разные вещи, отличающиеся друг от друга точно так же, как стратегия отличается от тактики. В первом случае вы, как правило, сами участвуете в проектах и, часто не только в качестве руководителя, но и программиста. И на вас как на руководителе команды висит задача проектирования разработки, вы следите за тем, чтобы проект укладывался в срок и т. д. Задачи важные, но в масштабах компании сугубо тактические, даже если вы сами уполномочены вести переговоры с заказчиками. А вот руководитель компании, выше которого только звезды, должен решать совсем другие задачи. Стратегические. Куда должна развиваться компания, что она будет делать через два года, через пять лет? С какими партнерами работать? Какую очередную бяку подсунет наше государство и как демпфировать последствия воздействия этой бяки на деятельность организации? И так далее. Да и методы руководства отделом из пяти человек и компанией из ста человек различаются. Хотя главный принцип работы с коллективом — один и тот же. С него и начнем.

Две стратегии руководства

По большому счету существуют только два способа руководства: командный и управленческий. (Третий способ, когда в кресле руководителя сидит ни на что не влияющая мяля, не будем рассматривать как тривиальный.) Вот выбор стиля руководства и является самым важным. Все остальное, о чем мы будем говорить после этого, — это, пусть и немаловажные, но дополнения.

Из названия ясно, что при первом способе вы командуете: "Пойди сюда, сделай то, потом это, отчитайся сначала за то, потом за это ..." Примерно так. Как в армии, в общем. Ваши подчиненные при этом практически не имеют никакой самостоятельности, или очень небольшую. Виноватыми при невыполнении какой-то задачи, естественно, будет тот, кто не выполнил ваш приказ. Командный способ имеет ряд неоспоримых достоинств: ощущение собственной значимости, возможность не подыскивать нужные слова и не объясняться за то или иное решение, быстрота выполнения задач... И так далее в том же духе. Нередко руководители такого типа действуют по принципу "боится, значит, уважает". Боится-то, может и боится, тут спору нет. А насчет уважает... Это вряд ли.

Второй способ прямо противоположен первому. Если в первом случае вы говорите "Сделай это!", то во втором нужно сотруднику объяснить причину появления того или иного вашего приказа. Труднее? Намного труднее, чем при первом способе. Хотя бы потому, что объяснить другим, почему они должны это сделать, часто намного труднее, чем придумать то, что надо сделать. Ну и еще потому, что командный стиль управления нередко предполагает приказы бессмысленные с точки зрения дела. Как в той же армии: увидел, что бойцы сидят и курят, значит, надо отдать приказ им побегать двести метров туда и обратно. Раз десять-двадцать-тридцать. В общем, столько времени, сколько до ужина осталось. Зачем? А просто, чтоб без дела не сидели. И чтоб знали, кто тут командир. Согласитесь, что такие приказы при втором способе управления аргументировать трудно. Не будете же вы объяснять подчиненному, что отдали такой приказ, потому что это на вас просто блажь нашла. А можно (и даже нужно) вообще ничего не объяснять. Если работа правильно поставлена, то указание может быть выполнено еще до того, как вы его огласите. Заключили вы, к примеру, договор на создание программного обеспечения, поставили о нем в известность свою команду и ушли пообедать. А пришли — вам уже говорят, что и как нужно делать, кто это будет делать и к какому сроку это будет сделано.

Примечание

По мнению В. Тарасова, идеальный руководитель в этом смысле — вообще с виду бездействующий. Как кажется бездействующим человек, управляющий стремительно несущимся парусником по сравнению с гребцом на весельной лодке.

Красиво? Да, неплохо. Но таких результатов никогда не будет при командирском стиле управления. "От забора и до обеда?" Хорошо. Но только от забора и только до обеда. А после обеда — пошел-ка ты, гражданин начальник куда-нибудь... А если послать тебя пока не получается, ну что-ж... Подожди. Пока я на твое место не встану, и будешь ты тогда от ужина и до ужина... Вот так вот, как правило, и бывает, когда преобладает командный

стиль. В принципе, повторяюсь, командовать намного легче. Но, правда, тогда уж и удивляться не надо некоторым вещам. К примеру, тому, что вы в отпуск уехать не можете из-за опасения, что пока вас нет, все развалится. И правильно опасаетесь. Потому что если не развалится, то уж двигаться ничего точно не будет. Или тому, что в вашем коллективе, если его можно так назвать, процветают наущничество и сплетни. А может тому, что вас предадут. Вы-то считали, что вас уважают, в смысле боятся, а оказывается, что подчиненные спят и видят, как бы от вас сбежать и забыть, как вас звали. А вы потом говорите, мол, предали... Да не предали. Это все равно, что сказать "стадо овец предало пастуха". Уж и кормил их, и поил, и хлыстиком хлестал, и гулять выводил, и в загон загонял, когда темнело. И все время ведь послушно ходили стадом, а потом взяли и ушли, твари неблагодарные. Предатели, а не овцы. ...Проведем такой простой опыт. Возьмите стакан с водой и поршень, который по объему должен быть меньше стакана. И давите этим поршнем на воду. Куда она денется? Правильно, вытечет. Можно создать идеальные условия, чтобы диаметр поршня четко равнялся диаметру стакана. И снова давите. Что будет? Правильно — при определенной силе давления поршня, превышающей критическую, стакан лопнет. И вода, опять же, вытечет. Вас результат не удивляет? Вряд ли. Детский опыт, скажете. Результат очевиден и предсказуем. Потому что вода стремится освободиться от давления и занять свободный объем. И не только вода. Что угодно. Теперь понимаете, почему при командном стиле управления удивляться тому, что ваши сотрудники разбегаются, или мечтают о том, чтобы вас разорвать где-нибудь в темном углу, не приходится. Детский опыт-то. И результат очевиден и предсказуем.

Поясню еще раз различие первого и второго способов на отвлеченном примере. Представьте, что вы — капитан. В первом случае, командном, вы должны будете стоять на мостике все время плавания. Потому что от команды от вашей толку никакого, так как она может выполнять только ваши приказы, а, значит, требуется ваше постоянное присутствие. Уходить с мостика действительно никуда не рекомендуется, хотя бы потому, что в ваше отсутствие корабль могут специально посадить на мель или еще что-нибудь такое сотворить. Чтоб вам тоже жизнь медом не казалась. Во втором случае вы можете вообще не выходить на мостик. Вернее, поначалу, пока всех своих первых-вторых-третьих помощников судовождению учите, вы только на мостике и торчите. Вначале действительно придется немало потрудиться, в отличие от командного стиля управления. Зато потом можете вообще из каюты не вылезать, потому что все сделают и без вас. А если действительно понадобится — позовут. Потому что ваша команда это действительно команда. А не табун.

Кстати, когда вы управляете, а не командуете, то всегда виноваты только вы. По определению. И искать причины того, что что-то делается не так, как вам хочется, надо только в себе. Приказ не выполнен? Значит, непонятно

сформулировали. Или не создали условий для его выполнения. Или отдали этот приказ тому, кто его выполнить в принципе не способен. Бухгалтерия что-то не то сделала? Значит, наняли не того бухгалтера. Главбух предал и сдал налоговой с потрохами? Значит, в людях плохо разбираетесь и не сумели в своем главбухе распознать червоточинку. Старейший сотрудник от рук отбил? Делать ничего не хочет, а зарплату требует, чтоб не меньше вашей? Тоже вы виноваты. Потому что не сумели вовремя распознать, что сотрудник зажрался, а, скорее всего, еще и сами создали ему такие условия, при которых он возомнил о себе невесть что.

Резюме

Ключевое слово командного стиля руководства — "заставить сделать", а ключевое слово управленческого стиля — "убедить сделать".

Способы управления

У многих существуют такие опасения: мол, если я не буду командовать, никто ничего не будет делать, так как все подумают, что наступила долгожданная свобода. В ответ на такую точку зрения надо заметить две вещи. Во-первых, не надо путать управление с анархией. Во-вторых, ваш способ руководства — это еще и хороший критерий проверки ваших сотрудников на способность думать. Если некоторые из них могут что-то делать "строго по приказу" (читай — из-под палки), значит, скорее всего, думать самостоятельно они не могут или давно разучились. Принимать решения и отвечать за них — тоже. Вам нужны такие сотрудники? Возвращаясь к первому замечанию, стоит добавить, что управление не подразумевает, скажем, мягкотелости. Совершенно нет. Управление как раз может быть жестким. Бывают случаи, когда добро должно быть с кулаками. Ведь даже на войне многие управляли, а не командовали.

Примечание

Приведу несколько примеров жесткого управления. Начнем с самого "мягкого". В младенчестве чтобы отучить меня совать пальцы в розетку, родители, как только я подносил палец к розетке, незаметно кололи его иголкой. Выработав у меня тем самым отрицательный условный рефлекс на такого рода нехорошие поступки. Это — управление. А вот если бы они говорили в приказном тоне: "Пальцы в розетку совать запрещено!", то вряд ли бы это подействовало. Вот когда на доходчивом примере разъяснено, почему запрещено (потому что больно и неприятно) — тогда да, тогда понятно. Кстати, просто так в приказном тоне что-либо запрещать — дело вообще неблагоприятное, потому что действует всем известный принцип запретного плода. Другой пример. Пятнадцатилетний сын моих знакомых страдал, скажем так, повышенной тягой к риску. К примеру, он очень любил перебежать перед машинами и поездами, так, чтобы они его едва не сшибали. Я не знаю, есть ли

у психиатров точное название для этого состояния, но, штука крайне нехорошая, держащая всю семью в постоянном невротическом состоянии. (А надо сказать, что пару раз таким образом он спровоцировал полномасштабное ДТП, к счастью, без человеческих жертв). После работы с тремя психотерапевтами состояние только ухудшилось, а взаимные претензии друг к другу в семье только возросли. Как-то разговарившись с отцом этого парня и услышав всю эту историю, я выразил желание попытаться помочь. Пришел к этому парню, Саше, и стал с ним разговаривать. Сначала, естественно, отношение его ко мне было, мягко говоря, крайне настороженное. И несколько надменное. Мол, еще один "специалист по мне" приперся. Для начала сказал ему, что к его "увлечению" отношусь нормально, что он такой не один на земном шаре, и пообещал ему, что гарантирую успешные переговоры с родителями на предмет освобождения его из домашнего заточения (метод подстройки под партнера). Но при одном условии. Поскольку его увлечение связано с игрой со смертью, я сказал, что мне хотелось бы показать ему этого второго игрока. Вот тогда — после того, как он увидит, что такое смерть и как она может выглядеть, он может продолжать заниматься своим увлечением. Потому что это уже будет по-честному, а не так, как сейчас — в силу мимолетной безумной храбрости. И заодно объяснил ему, чем отличается безумная храбрость от осознанного риска. А пока, говорю, все, что ты делаешь — это всего лишь экстремальный онанизм, за который уважать человека как-то не получается. А уважения этот парень хотел, как и все мы. В общем, пошли мы с ним в морг. Вернее, для начала в анатомический музей мединститута. В принципе, ему хватило бы и его. То, что с ним там было, описывать не буду. Все то же, что бывает с теми, кто в первый раз увидел, что такое реальная, а не киношная смерть. Но моей садистской натуре этого показалось мало, и я его еще сводил в общегородской морг, который от анатомического театра отличается примерно так же, как корзина для мусора под столом у офисного работника отличается от городской мусорной свалки. Домой я его практически нес на руках, потому что все силы из него вышли. Сами понимаете с чем. Дальнейшее "лечение" описывать не буду, скажу лишь, что была трехдневная депрессия с отказом от приема пищи, а потом от прежнего увлечения не осталось и следа. Этот случай — тоже пример управления. А не командования, которое, как мы видели, ни к чему не привело. Ведь до этого его родители фактически им командовали: ты этого не должен делать и все тут. А почему не должен? Никто не объясняет. Объяснение в стиле "ты можешь погибнуть" для человека, ничего кроме боевиков не видевшего и выросшего на рассказах о придуманных подвигах "ребят с нашего двора", ничего не значит. Воззвания на тему "нам с отцом пришлось платить большие деньги за ремонт машин, которые из-за тебя попали в ДТП" опять же ни к чему не приводят, так как он еще не знает, как тяжело эти деньги могут доставаться. И так далее. Я же попытался не командовать, а управлять, т. е., ничего не требуя, просто показал, чем его увлечение, скорее всего, очень скоро закончится. А решение (вполне прогнозируемое) он принял самостоятельно. Причем, как видите, управление было достаточно жестким — никто ни с кем не сюсюкал. И опасения на предмет того, что при переходе от командного стиля к управленческому у вас в коллективе начнутся разброд и анархия, потому что вы считаете, что ваш коллектив требует жесткой руки — беспочвенны. Потому что, как видите, управление может быть и жестким, но все же это управление, а не командование. Какой способ предпочтительнее в вашем случае — решать вам.

Чем отличается руководство компанией от руководства командой

Тем же самым, чем стратегия отличается от тактики. И очень опасно перепутать эти "два руководства". Ну, нельзя руководить компанией точно так же, как вы руководили командой. Нередко приходится видеть примеры, к сожалению, печальные, когда при руководстве компанией руководитель путает стратегию с тактикой. Вроде бы и продукты создавались успешно и в срок, и зарплаты большие, и клиенты довольны... В общем налицо все внешние признаки успеха. А потом — бах! — и за какие-то полгода-год нет компании. Как? Почему? А потому что была перепутана стратегия с тактикой. Руководитель, серьезно вникая во все тактические задачи (на чем и как пишут, кто и что конкретно делает, по какой причине дворник Петр плетет интригу против уборщицы Люси), совершенно упускал из внимания стратегию. И, всецело будучи поглощенным тактикой, не увидел, что рынок чуть-чуть стал поворачиваться в другую сторону, что правительство подготавливало указ, который, вступив в силу, очень сильно "подрезал" деятельность компании, что конкуренты окольными путями подготавливали почву для увода большей части клиентуры... Хотя внешне все было очень даже прилично. ...Этот пример списан с реальной ситуации вокруг одной компании, руководителем которой был мой знакомый. Немаленькой компании с немаленьким оборотом. И руководитель был весьма высоколобый. Но... Занимался тактикой, когда уже надо было решать серьезные стратегические задачи, и вот, что из этого получилось.

Примечание

Точно так же многие программисты, став руководителями отделов или компаний, большую часть своего времени продолжают участвовать в проектах и выдавать код на-гора. И очень переживают, если выясняют, что кто-то из сотрудников знает больше языков программирования, чем они. "Какой позор для меня, как руководителя программистской компании! Я знаю 14 языков программирования, а мой сотрудник 15! Меня же уважать перестанут!". Увы, нередко такое бывает в тех или иных вариациях. Только надо понимать, что сотрудников больше волнуют вопросы собственной зарплаты, стабильности организации, престижа работы в данной организации, чем то, что вы знаете на один язык программирования меньше, чем они. Да хоть вообще все языки забудьте! Крайний, конечно, вариант, но если в вашей организации будут достойные зарплаты, работать в ней будет престижно, и все будут понимать, что существовать она будет очень долго, если не вечно, — все ваши сотрудники на всех углах будут кричать, какой вы гениальный программист, даже если из всех языков, которые вы когда-то знали, ныне вы помните только русский разговорный. Потому что умные ваши сотрудники прекрасно понимают, что не можете вы, руководя компанией, писать код. Потому что заботы у вас уже совершенно другие. Потому что грамотное управление — это, по сути, новая профессия. Которой вам надо овладевать. Причем, зачастую, "с колес". А вот если вы этой

новой профессией не овладеете, и в вашей компании будут низкие зарплаты, а работа с рассвета до заката, и все будут опасаться, как бы ваша фирма не закрылась по истечении очередного проекта, — тогда да. Учите этот пятнадцатый язык, потому что в этом случае против вас будет играть любая мелочь.

Вы можете мне возразить: мол, дисквалифицируюсь, потеряю профессию. Согласен — неприятно. И не нужно. Нельзя терять свои профессии. Но надо также четко понимать, что если уж сели в кресло руководителя (в конце концов, это было ваше решение, вряд ли вас принуждали к этому под дулом пистолета), то, скорее всего, на руководство будет тратиться львиная доля вашего времени и сил.

Но о тактике тоже нельзя забывать

Ни в коем случае. Стратегия и тактика, как две сестры-близняшки, должны идти рука об руку. Сильный крен в ту или иную сторону опасен. О том, чем опасен крен в сторону тактики, мы уже поговорили, теперь о том, что происходит, если всецело сосредоточиться на стратегии. А происходит вот что. Все вы, к примеру, правильно просчитали, поняли, что и как надо сделать, чтобы ваша компания не знала горя в ближайшие лет десять, и, решив, что пришла пора начать делать, а... А выяснилось, что делать не с кем. Потому что проводимая вами кадровая политика (тактическая задача) не была нацелена на выполнение задачи стратегической. Попросту говоря, сложно делать большие дела, когда сотрудники коллектива не умеют принимать самостоятельные решения и отвечать за эти решения, когда в коллективе одни интриги, когда сотрудники не понимают поставленной перед ними задачи, когда у вас нет необходимого оборудования, которое нужно было приобрести заранее, когда выясняется, что нужно было некоторых сотрудников уже давно послать... На курсы повышения квалификации. Таким образом, что у нас получается: тактика — это фундамент, на котором строится стратегия. А тут, как в строительстве: невозможно строить здание, если не заложен фундамент (не проработана тактика). Точно так же, если всю жизнь строить только фундамент (заниматься только тактикой), здание тоже никогда не будет построено.

Не складывайте все яйца в одну корзину

В смысле не занимайтесь только одним видом деятельности. ...Сложный для меня раздел. Посоветовать-то просто, а сделать в реальности гораздо сложнее. И все-таки: ищите, все время ищите новые виды деятельности. Потому

что то, что сегодня пользуется спросом, завтра может оказаться никому не нужно. Так не одна компания разорилась. И наоборот: то, что сегодня убыточно, завтра может начать приносить неплохой доход. Многие мудрые руководители, несмотря на то, что сегодня их фирма процветает, "вешают на себя" убыточные производства. И развивают их. Потому что думают о завтрашнем дне. И знают, что если сегодня в это производство вложиться, то завтра оно может начать приносить прибыль. А может и не начать. Поэтому лучше вкладываться в несколько новых проектов. Вероятность того, что какие-то из них в будущем "выстрелят", в этом случае значительно повышается. Делаете вы, к примеру, сайты? Прекрасно. Научитесь создавать презентации. Освойте промышленный дизайн. Неплохо писать код и не только для сайтов. И наоборот: если вы занимаетесь разработкой прикладного программного обеспечения, к примеру, бухгалтерской направленности, будет очень неплохо, если вы освоите программирование для Web. И так далее. А если позволяют средства и фантазия, то будет совсем неплохо, если вы вложите в вид деятельности, совершенно не связанный с вашей основной деятельностью. В нашем случае — с программированием. А откроете, к примеру, магазинчик или постройте пару зданий, комнаты в которых начнете сдавать в аренду под офисы... В этом случае все происходящее будет восприниматься значительно проще. По крайней мере, вас не хватит инфаркт после того, как вы узнаете, что основной бизнес вашей компании, скажем, разработка сайтов, накрывается медным тазом, потому что наше любимое правительство, в очередной раз душевно расстроившись, сделало разработку сайтов лицензируемым видом деятельности, и издало негласный указ, что лицензии выдать только пяти-шести крупным столичным компаниям. (Надеюсь, конечно, что такого никогда не будет, и это лишь моя разыгравшаяся фантазия, но если будет, то совершенно не удивлюсь. Бывало и покруче. В смысле — абсурднее.)

...Это правило работает не только в бизнесе, но и в жизни вообще. Старайтесь не посвящать себя только одному виду деятельности. Я видел очень много людей, которые, будучи не по своей воле выкинуты из своей любимой профессии, очень долго по этому поводу пребывали в депрессии, а некоторые так и не смогли оправиться. Потому что думали, что этого никогда не может случиться. А зря думали. Если уж такую серьезную и большую организацию, как СССР, угораздило развалиться, то что говорить о более мелких. Кроме того, совершенно не обязательно, что развалится организация, в которой вы работаете. Вас могут из нее просто — на просто "уйти". Говорите, не могут? Что вы такой заслуженный, всеми уважаемый суперспец? Зря говорите. Могут, еще как могут. Именно таких в первую очередь и уходят.

Примечание

Мне посчастливилось быть чуть-чуть знакомым с одним легендарным генералом. Несколько мимолетных, ни к чему не обязывающих "поездных" встреч, как я их называю... Когда его "ушли" из той организации, которой он отдал свою жизнь, это был уже известный человек, Герой России. Я помню, как рассказывал ему о своих первых успехах в научной деятельности. Генерал слушал меня с интересом, потому что результаты действительно были интересными (из того разряда, когда то, что сделано, можно объяснить "на пальцах" первому встречному). Потом спросил, какие у меня отношения на работе с вышестоящими. А отношения, надо сказать, были не очень... Не плохие, не хорошие — относительные такие отношения. Тут генерал сказал то, что мне тогда показалось кощунством: "Поищите себе запасной вид деятельности, мало ли что... Программированием еще увлеклись? Чудесно! Перестаньте увлекаться, займитесь серьезно. Пойдет хорошо — возьмите да фирму откройте. В любом случае не помешает". Я возмущился, сказав, что собираюсь заниматься только наукой, что никто меня тронуть не посмеет, ведь у меня такие серьезные результаты, много научных работ, — даже больше, чем у тех, кто старше меня на пятнадцать лет, что я уже готов кандидатскую написать... Он на меня очень грустно посмотрел, вероятно, восхищаясь внутренне моей непроходимой наивностью, и сказал примерно так: "И все-таки подумайте над тем, что я сказал. Серьезно подумайте. Вот если бы у вас всех этих работ не было, и результаты были бы самыми обычными, — я бы вам ничего этого не говорил. У вас все было бы нормально и обычно. Как у всех. А вот когда много работ, результаты, звания, награды — тут надо очень зорко по сторонам смотреть". Спасибо, генерал. Я не пропустил мимо ушей ваши слова, лично вами выстраданные, как я сейчас уже знаю. И, сначала играясь, а потом все серьезнее и серьезнее, занялся программированием (вместе со своим товарищем и соавтором всех книг, который наград имеет столько же, сколько и я, и которому тоже помешали защититься и остаться в науке академические кости). Получилось сделать студию. Написать первую книгу. А потом создать отдел в крупной связистской компании. А сделав все это, я предельно четко понял, что ждет меня в науке. И еще понял, что от наших чиновников можно ожидать всего, чего угодно. Даже того, чего от людей нельзя ожидать в принципе. И уже не переживал, когда меня гоняли по высоким кабинетам, заставляли по несколько часов ждать в приемных кабинетах, в которых потом большие дяди, упиваясь собственным величием, учили меня, как надо жить и с кем дружить. Не переживал, потому что знал, что это — для меня не конец всей жизни, а лишь конец одного из ее этапов. За который мне не стыдно, и который я прошел достойно. Не переживал, потому что знал, что впереди ждет другой жизненный этап, не менее интересный. В общем, этот инцидент прошел с меня, как с гуся вода. А для многих подобные инциденты стали очень серьезным жизненным испытанием. Для некоторых — последним. Лично знаю девушку, которую по собственной блажи научрук не допустил до защиты, после чего она сошла с ума. Знал очень умного парня, которому для того, чтобы выманить деньги из его небедных родителей, написали отрицательный отзыв, и который после этого покончил жизнь самоубийством. Знал очень успешного научного сотрудника, который заболел раком через месяц после того, как выродок-завкафедрой его уволил для того, чтобы освободить ставку для балбеса-сына своего друга. Знал многих капитанов и военных, которые после того, как уходили с флота или из армии, через очень небольшой промежуток времени заодно уходили и из жизни. Много таких примеров, увы, и

вы наверняка знаете. Поэтому, насколько возможно, искренне вам, дорогой читатель, желаю, чтобы про вас никто никогда такого не знал.

Не надо меня понимать слишком уж буквально, в том смысле, что получается, надо не жить и работать, а все время только "готовиться к войне". Нет, я не про это. Я про то, что надо знать, что "война, в принципе, может случиться". И подготовить у себя в голове запасные варианты, которые до поры до времени законсервировать. Как бомбоубежища. Не понадобятся — прекрасно! А вот если понадобятся, а их не будет — вот это уже опасно. Для жизни опасно. В прямом смысле слова. Помните, в *главе 4* был у нас бомж, кандидат наук? Это как раз пример того, что может стать с человеком, если у него "нет запасного аэродрома". Да что я об этом говорю... У каждого из вас наверняка отыщется не один пример того, когда приказ об увольнении фактически становился для человека приказом об отправке на тот свет. Много примеров. Слишком много...

Дело в том, что когда человек может делать только что-то одно, он очень уязвим. И им можно легко командовать. Он как неподвижная мишень — в которую стреляй, сколько влезет, с какого-нибудь раза да попадешь. А вот если мишень движется — тут можно и промахнуться. Кроме того, когда ваше руководство (соучредители, партнеры, чиновники, которым вы подконтрольны) будет знать, что вы в случае ухода не помрете, а вот у них дела могут пойти значительно хуже — они будут очень долго чесать голову, прежде чем решатся предпринять в отношении вас активные действия. Ну и, наконец, когда имеете в голове запасные варианты, и когда процентов 5—7% своего времени отводите на то, чтобы их прорабатывать, жить на самом деле спокойнее. Хотя с первого взгляда может показаться, что наоборот. Но подсознание-то ваше умное прекрасно знает, что ни от чего нельзя быть застрахованным, и тревожится в отличие от вас (в смысле — вашего сознания) о том, что на случай "вдруг чего" нет в запасе ничего. И посылает вашему сознанию тревожные сигналы. А вот когда в запасе много чего есть, то жить вам на самом деле станет спокойнее. И проработка этих запасных вариантов превратится для вас в забавную игру. Результатами которой, с немалой вероятностью, и не придется воспользоваться. Ну, уж если придется, — то вы себя 100 раз похвалите за предусмотрительность. Это как в любом государстве, которое наращивает свою военную мощь даже тогда, когда на него никто не собирается нападать. А когда соберутся, то могут и передумать, после того, как военную мощь взглядом окинут. А если и не передумают, то у государства с хорошим военным потенциалом есть много шансов отдубасить супостата и выйти из схватки победителем. А вот если военная мощь не наращивалась, если все жили в спокойствии и уверенности, что с ними такого произойти не может, что никто никогда не решится... Вот в этом случае шансов выиграть войну нет никаких. Да и нападут на такое государство с 90% вероятностью. Потому что это сильных обходят стороной, а слабых — бьют.

Главный психолог коллектива

Очень важное правило: главный психолог коллектива — всегда руководитель. Он и только он. Даже если у него в подчинении специальный психологический отдел, возглавляемый суперпуперпсихотерапевтом всех времен и народов. И второе важное правило, касающееся психологии, заключается в том, что она есть. Да, именно так. А многие считают, что ее нет, что это — чушь, придуманная западными империалистами, и вполне можно обойтись и без нее. Нельзя. Никак нельзя. Потому что работаем мы все с людьми, а значит, если хотим достичь желаемого, должны разбираться в психологии, хотим мы этого или нет. Очень часто кризис многих компаний происходит именно из-за того, что в расчет не принималась психология коллектива.

Люди живут для себя

Не стоит впадать в наивность и считать, что люди приходят в фирму (команду, институт, на производство и т. д.), чтобы укреплять благосостояние фирмы. Люди приходят укреплять свое благосостояние. Опытные управленцы это положение прекрасно знают. И делают так, чтобы работник понял, что укрепление фирмы важно для укрепления его благосостояния.

Престижность

Точно так же нельзя забывать, что людям очень важно ощущать, так сказать, "престижность". То есть если ваша фирма серьезная, с ней и с ее руководителем считаются, иначе говоря, если в фирме престижно работать — для многих это не самый последний довод в пользу того, чтобы честно служить фирме.

О престижности

Еще немного о том, почему важна престижность. Все мы в той или иной мере озабочены чувством собственной значимости, о чем очень хорошо и подробно написал Д. Карнеги в своих известных книгах. Поэтому наше Я противится тому, чтобы о нем плохо думали даже в косвенном смысле. К примеру, многим неприятно, когда кто-то плохо отзывается о наших друзьях. (На подсознательном уровне возникают примерно такие соображения: "Ведь это МОИ друзья. Их выбрал Я. А если он плохо о них отзывается, значит, он не доверяет МОЕМУ выбору".) По этому же механизму идентификации решается вопрос престижности работы. Если человек работает в престижной фирме, то его Я есть чем себя потешить: "Я работаю в престижной и развивающейся компании, МЕНЯ взяли сюда работать, значит, МОИ навыки и умения стоят немало".

Как превратить группу в коллектив

Очень важная задача руководителя — превращение той группы, которой он руководит, в более-менее нормальный коллектив.

Примечание

О том, чем отличается группа от коллектива, мы говорили в разделе, посвященном социальной психологии.

Мне кажется, что превращение группы в коллектив и есть та основная задача руководителя, которую никто кроме него в организации выполнить не сможет. И сделать это может только руководитель, так как он и только он является главным психологом коллектива.

Примечание

И действительно — многие задачи руководитель в той или иной степени может перепоручить другим. (Заключать договора можно позволить талантливому менеджеру, сделав его заодно замдиректора, вести финансы — хорошему главному бухгалтеру или финансовому директору и т. д.) Создание коллектива — одна из тех немногих основополагающих задач, которую нельзя перепоручить никому. И правда: даже если в организации есть талантливый психолог, который мог бы это сделать, у него это не получится только потому, что он не руководитель: во-первых, подчиненные в организации осознанно или неосознанно подстраиваются под психологический строй руководителя и, во-вторых, для создания коллектива руководитель должен принимать ряд ответственных решений, которые находятся только в его компетенции.

Для начала вспомним определение активной группы.

Активная группа — группа, занимающаяся деятельностью, которая ведет к достижению основной задачи группы.

А коллективом может стать только активная группа, т. е. группа, вся деятельность которой направлена на достижение своей основной задачи: в больнице лечат, на заводе выпускают продукцию, в строительной организации строят и т. д. Причем все время должны внедряться новые формы работы, чтобы сотрудники чувствовали интерес, могли найти свое место в группе, имели шанс пойти на повышение и т. д. То есть при выполнении своей основной задачи группа должна постоянно развиваться. Тогда она будет оставаться коллективом. Если же в деле застой, т. е. все делается по старинке, новые методы не внедряются, то в коллективе скоро будут возникать многочисленные алкогольно-сексуальные и культурно-развлекательные группы при одновременном уменьшении численности учебно-карьеристской группы (потому что карьеристы не могут не развиваться и будут искать себе новые места работы, где они смогут двигаться вперед). И в результате протекания этих процессов неизбежно наступает распад коллектива.

Важно!

Распределение благ в коллективе должно происходить только в соответствии со степенью трудового участия.

Если руководитель хочет руководить коллективом, а не сбродом, у него не должно быть "любимчиков" в работе. Все должно быть по следующему принципу: сделал работу — получи блага, сделал много — получи много, не сделал ничего — ничего и не получишь. И здесь исключений быть практически не должно. Конечно, если сотрудник, скажем, пахал на фирму полгода без выходных и проходных, а потом у него, скажем, заболела жена, и он месяц работал вполсилы, а вы ему за этот месяц зарплату урежете вполонину... Это неправильно. Это самодурством называется. Здесь никакого исключения нет, так как сотрудник, работая без усталости продолжительный промежуток времени, вправе рассчитывать на то, что в сложной ситуации вы его без средств к существованию не оставите. Это как раз те самые блага, на которые он вправе рассчитывать, работая не покладая рук. Продолжая эту мысль, нужно сказать, что огромным значением при создании коллектива является отношение к столкнувшимся с неприятностями сотрудникам. Если руководитель не помогает попавшим в беду работникам, то последние никогда не будут жертвовать своими интересами ради интересов коллектива. И, соответственно, наоборот.

Примечание

Этот принцип особенно хорошо пропагандируют американцы. Да, да — я именно про это всем известное изречение "Если наш человек попал за рубежом в беду, то даже для спасения одного человека мы пошлем войска". Такое отношение к людям очень важно. Причем оно даже важнее для вполне благополучных сотрудников коллектива, чем для тех, у кого неприятности. Потому что людям очень важно знать, что если с ними случится что-то нехорошее, то их в беде не бросят.

Другой важной обязанностью руководителя коллектива является, пользуясь термином Э. Берна, защита большой наружной границы. То есть все внешние дела, переговоры, выезды на симпозиумы, важные встречи — это компетенция руководителя.

Примечание

Я знаю не один пример, когда руководители уклоняются от важных переговоров, посылая на них своих сотрудников. Которые очень нередко дело проваливают. Результат: руководитель злится на сотрудников, а последние чувствуют себя уязвленными. Хотя злиться руководителю надо только на себя: он прекрасно осведомлен, что некоторые переговоры проваливаются по определению, если на них приезжает не первое лицо. Конечно, при таком подходе коллектив вряд ли сложится.

Конечно, вы можете передоверять некоторые из "внешних" дел своим сотрудникам, но только в том случае, если уверены, что они с ними справятся. А если речь идет о "пробивании" или "выбивании" чего-либо (материальной базы, кредита и т. д.), то в любом случае лучше всего этим заниматься самому. Потому что здесь высока вероятность того, что у вашего, пусть и очень доверенного, сотрудника это может не получиться.

О психологическом вампиризме руководителей

Очень нередко руководители впадают в один из видов психологического вампиризма, суть которого заключается в следующем. Они посылают своего сотрудника на переговоры, точно зная, что он их провалит (их провал либо предопределен, независимо от того, кто ведет эти переговоры, либо для того, чтобы они были успешными, их обязательно должно вести первое лицо). После того, как у подчиненного, естественно, ничего не получается, наступает один из двух вариантов. Либо, вариант первый, подчиненного ругают на чем свет стоит, играя в игру "Какой ты плохой". Вампиризм здесь в том, что руководители точно знают, что переговоры (договор, сделка и т. д.) будут провалены, и удовлетворяют свое вампирское нутро, ругая на все лады своего подчиненного. Либо играют не в игру "Какой ты плохой" (не ругают), а в игру "Какой я хороший". То есть руководитель картинно вздыхает, говорит "Ну что ж с вами поделаешь" и идет улаживать дела. У него все получается, он приезжает довольный собой, играя в игру "Учитесь, сынки, как работать надо". Здесь вампиризм в том, что руководитель, выступает в роли этакого избавителя, который показал сотрудникам, как надо действовать в сложной ситуации (при этом осознанно ее спровоцировав, чтобы потом с блеском разрешить). Очевидно, что при таких руководителях-вампирах коллектив не сложится.

Поручайте своим сотрудникам только те дела, которые они могут выполнить

Очень часто описанные выше нехорошие ситуации (недовольство подчиненными и уязвленность последних) случаются не потому, что руководители — вампиры, а просто из-за того, что они поручают своим подчиненным дела, которые те не способны выполнить! Поэтому, повторяюсь еще раз, так как это важно: не поручайте своим сотрудникам таких дел, которые, как вы знаете (или чувствуете), они не способны выполнить! Даже если они этого просят. И дело завалите, и сотрудника можете потерять.

Примечание

Подобные вещи допустимы только в отдельных воспитательных случаях. К примеру, один тренер по рукопашному бою выставил против своего одаренного ученика, побеждавшего на всех соревнованиях в своей весовой категории, заведомо более сильного и опытного партнера, в схватке с которым ученику было не выиграть и полпраунда. Мораль сей басни такова: тренер хотел пока-

зять ученику, начавшему заболевать, скажем так, звездной болезнью, что нет предела совершенству. В общем, сбил спесь. Ученик все понял правильно, хотя сначала, конечно, расстроился.

Подобный прием (из примечания выше) в данном случае вполне оправдан. Но! Тренер дал ученику неразрешимую задачу в **его области профессиональной деятельности**! То есть он просто поднял планку в его развитии на большую высоту, да и то, зная, что ученик эту высоту взять в состоянии. Переводя на наш программистский язык, если вы дадите своему сотруднику задание написать более сложную программу, чем те, которые он писал до того, — в этом ничего страшного нет. Даже если он с этим заданием не справится. Если сотрудник честолюбивый — он возьмет эту новую для себя высоту. И вы получите более ценного специалиста. Ну, а если, по каким-то причинам, у него это не получится, то вы будете знать, где потолок данного работника, и не будете давать ему заданий не по плечу. А вот если вы будете давать сотруднику сложно разрешимые задачи, находящиеся вне его интересов, то и результат будет вполне очевидным. И не надо после этого ругать бухгалтера за то, что он не написал макрос, а программиста за то, что он не квалифицированно говорил с налоговым инспектором.

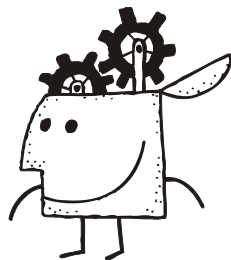
О критике, или метод кнута и пряника

Да-да. Он самый, не одну тысячу лет назад изобретенный и ничем не заменимый и по сию пору. И вряд ли когда будет ему альтернатива. Потому что в основе лежит фундаментальный постулат сравнения — все познается в сравнении. Тепло и холод, жизнь и смерть... Известный способ Пеле — водить потенциальных самоубийц и глубоких депрессивников, разочаровавшихся в жизни, в морг — из этого же числа. Чтобы, посмотрев на смерть, люди поняли, что жить стоит. Бесхитростный и гениальный способ, очень многим помогший. Знаю по себе, когда приходилось бывать в этом малораздостном заведении, по выходе из него первая возникающая мысль "Как хорошо, что вокруг все живые". Плохой и добрый следователь — тоже отсюда же. Замечали, наверное, в фильмах про мафиози, когда пытаются человека, чтобы он чего-то сказал, или сделал, или наоборот не сказал и не сделал, делают примерно так. Избивают, скажем, испытуемого, несколько человек в грязных робах, с нематыми волосами, плохо пахнущие и сквернословящие, кричащие "Бей его, гада, еще бей"... А после того, как закончится раунд избивания, подходит с иголки одетый человек в галстук (в фильмах, как правило, главный мафиози), приятно пахнущий и правильно говорящий: "Ну что ж вы, Василь Петрович, так неразумно себя ведете?" И так далее. И многие, даже не самые слабые духом, ломаются. И в реальной жизни тоже, не только в фильмах. Потому что — плохой и добрый следователь. Потому что работает этот прием, и все время будет работать, так как основан

на фундаментальных законах даже не человеческой психологии, а жизни. Потому что после злых, издевающихся дядек этому опрятному гражданину хочется все рассказать, даже понимая, что этого делать нельзя. Потому что для того, чтобы методу этому противостоять, нужна очень неслабая психика, годы тренировок нужны. А если не в лоб действовать, а с подходом, с использованием многочисленных вариаций, то и на сильную психику средство найдется. Метод кнута и пряника из той же серии методов противопоставления: **если поругали подчиненного, после этого обязательно похвалите.**

Теперь о том, как ругать. Вернее, критиковать. **Во-первых**, всегда с глазу на глаз и никогда при подчиненных. Потому что ваша цель — не покрасоваться перед обществом, а объяснить человеку, что он сделал неправильно. И если проступок серьезный, то и объяснить можно жестко. И наказать. А вот хвалить лучше при всех, если есть такая возможность. **Во-вторых**, критикуйте **поступок, а не человека**. "Сашка! Дурак! Ну кто так делает! Так делают только глупцы! И в кого ты такой уродился", — это критика человека. И тот, кто так критикует, не слишком далеко ушел в своем умственном развитии от этого Сашки. Который дурак. Да, Сашка, может, и совершил глупый поступок. О чем ему можно и нужно сказать. Но — критиковать нужно именно поступок и только его, а не человека. (Хотя бы потому, что ни мне, ни вам и никому никто не давал права судить других людей в целом. А вот судить о поступке мы можем в меру своей профессиональной компетенции). И нельзя характеристику поступка переносить на характеристику личности по принципу "Ты совершил глупость, значит, ты глупец!" А как надо? А хоть вот так, к примеру: "Саша! Ну как ты мог сделать такую глупость! Ты же кандидат наук! Что с тобой произошло в тот момент, когда ты это делал?" Тут, как говорится, и волки сыты, и овцы целы. И поступок вы по достоинству оценили, и человека не оскорбили. **В-третьих**, за один раз ругайте только за один проступок. Если хотите, чтобы человек что-то усвоил. **В-четвертых**, помните, что наказание должно быть адекватно преступлению. И в прямом, и в обратном смысле. С одной стороны, конечно, не стоит лишать человека месячной зарплаты за то, что он проспал и опоздал на работу. Но с другой стороны, если работник серьезно провинился, и фирма из-за него понесла материальные и моральные убытки, то говорить ему "Ну, ты так, пожалуйста, больше не делай" — тоже не нужно. Здесь уже нужен кнут, а не пряник. Кроме того, неплохо четко определиться с теми проступками, за совершение которых автоматически следует увольнение. (Только не забудьте этот "список проступков" довести до сотрудников). Но если уж такой список обнародовали, поблажек никому не делайте.

Глава 7



Стратегический маркетинг — программисту, или как найти свою нишу на рынке software

Создание программного обеспечения, как и любая другая область, часто обременяется мифами и традициями. Однако, в отличие от других областей, программное обеспечение выполняется на компьютерах, которые демонстрируют совершенно парадоксальную тенденцию — их качество и быстродействие растут, а цена падает уже на протяжении многих лет. С момента создания первых ЭВМ принципы продажи программного обеспечения сменились три раза.

В начале развития компьютерной области вычислительные машины стоили миллионы долларов и потребители могли позволить себе заказать изготовление программного обеспечения стоимостью в десятки тысяч долларов. В настоящий момент большие и дорогие ЭВМ так же имеют место быть, но доля их от общего парка машин очень мала. Благодаря взрывообразному распространению персональных компьютеров основной задачей программистов стало удовлетворение потребностей массового рынка.

В современных условиях, когда компьютеры стоят несколько сотен долларов, пользователь не может позволить себе купить программу в десятки тысяч долларов и предпочитает покупать коробочные варианты, которые в силу того, что продаются массово, имеют низкую стоимость. Кроме того, при переходе к персональным компьютерам обычным пользователям было психологически проще купить коробку и готовую программу — они покупали в коробках другие товары: пищу, одежду, бытовую технику, поэтому появление коробочных вариантов пришлось очень по душе свежее испеченной армии пользователей, которые не привыкли заказывать и ожидать выполнение заказа в течение долгих месяцев. Делать ставку на работу по заказу можно только в узкоспециализированных областях, для которых маловероятно, что появится новая команда разработчиков. Если работа ведется на массовом

рынке, следует выходить с готовым продуктом, который можно показать покупателю и продать его немедленно. Современные потребители в настоящий момент более склонны к покупке готового программного обеспечения, чем вкладывать деньги в разработку программного обеспечения на заказ.

Другим следствием стремительного развития вычислительных мощностей стало резкое снижение стоимости компьютеров. В настоящее время почти невозможно встретить понятие "компьютерное время", которое еще десять лет назад являлось предметом продаж, аренды и имело вполне определенную стоимость. Не секрет, что при разработке операционной системы Windows 95 основным требованием была возможность запуска ее на компьютерах с 8 Мбайт оперативной памяти, что и привело к значительному снижению качества первого релиза. До недавнего времени максимальный объем оперативной памяти в компьютерах достигал 2 Гбайт. Причем таким объемом снабжались лишь очень загруженные серверы. На момент написания книги 2 Гбайт равны по стоимости средней месячной заработной плате в Российской Федерации, а серверы снабжаются 12 Гбайт оперативной памяти.

Затраты, связанные с оптимизацией кода по скорости или по объему занимаемой оперативной памяти, не окупаются. Время программиста, затраченное на создание и последующее сопровождение такого кода, стоит дороже, чем компьютерное время и необходимый объем памяти. Учитывая, что в большинстве случаев ведется разработка массового продукта, нацеленного на конкретного потребителя, стоимость одного часа работы программиста возрастает многократно. Вместо того, чтобы усложнять код никому не нужной оптимизацией по скорости и объему, которая вредит как читабельности программы, так и устойчивости ее работы, лучше направить основные усилия на пользовательский интерфейс, поиск ошибок и создание как можно более простого в сопровождении кода.

С появлением сети Интернет распространение программ вошло в третью фазу. Пользователи к моменту увеличения популярности Интернета осознали, что программы это не коробка, а исполняемый код. Стоимость же коробочного варианта определяется упаковкой, печатью руководства пользователя, складских расходов и наценки распространителей. Всех этих лишних расходов можно избежать при покупке программы через Интернет. Поэтому коробочные варианты становятся вспомогательным способом распространения программы, выполняющим больше имиджевые функции. Распространение программ через Интернет уже привело к значительному снижению стоимости программного обеспечения с нескольких сотен долларов до нескольких десятков. Ярким примером использования распространения программ через Интернет являются демо-версии программ или полностью рабочие shareware-программы с ограниченным временем использования. Еще одной распространенной практикой являются on-line-сервисы — пользова-

тель может испробовать работу программы и сравнить ее с другим программным обеспечением. Созданием таких сервисов занимаются разработчики антивирусного программного обеспечения и электронных переводчиков — пользователь может проверить качество программы прямо на сайте, не загружая демо-версии.

Программное обеспечение является практически чистой мыслью, но, в отличие от книг, созданные один раз, они не требуют усилий по их дальнейшему копированию. Все усилия направляются на устранение недочетов и защиту от несанкционированного копирования. Невозможность должным образом защитить программу является часто фундаментальным свойством, связанным с тем, что программа должна выполняться при определенных условиях на машине пользователя. Единственным способом оградить ее от несанкционированного использования является ее полная неработоспособность. Поэтому разработчики программного обеспечения часто бесплатно распространяют свои программы, извлекая выгоду из побочных видов деятельности.

Многим кажется невероятно глупым отдавать бесплатно код в свободное пользование. Однако это не совсем так. Действительно, свободное распространение программного обеспечения было неоправданным в то время, когда число пользователей компьютеров было невелико и ограничивалось обслуживающим персоналом огромных мэйнфреймов, работающих на корпорации, которые покупали любое программное обеспечение за любые деньги, если только появлялся намек, что оно им нужно. Современные реалии таковы, что рынок программного обеспечения состоит из миллионов пользователей небольших персональных компьютеров, и как только выпускается продукт по недоступной цене, он будет либо воспроизведен пусть с меньшей функциональностью и низшим качеством, либо украден, либо просто не получит распространения. Жизнеспособность идеи бесплатного программного обеспечения демонстрирует как свободная операционная система Linux, разрабатываемая энтузиастами, так и бесплатные коммерческие проекты вроде реляционной базы данных MySQL. Зачастую гораздо выгоднее привлекать армию пользователей и продавать им услуги по сопровождению, показывать им рекламу, навязывать им стандарты, чем требовать с них деньги.

Далее в главе описываются различные компании, которые успешно работают на современном рынке и осуществляют успешную маркетинговую политику в мире миллионов персональных компьютеров, владельцы которых не спешат расстаться со своими небольшими деньгами.

MySQL

На сегодняшний день MySQL, наряду с СУБД Oracle, является одной из самых быстрых баз данных, успешно работает под управлением свыше 20 различных операционных систем и предоставляет интерфейс для взаимодействия с множеством языков программирования: C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, Tcl и др. В последней версии MySQL была достигнута функциональность, доступная ранее только в коммерческих базах данных. Чем же живут сотрудники корпорации MySQL AB? Дело в том, что база данных MySQL с открытым кодом распространяется по двойной лицензии: GPL, требующей, чтобы все разработки с применением кода, распространяющегося по GPL-лицензии, также распространялись под GPL-лицензии (что недопустимо для закрытого коммерческого использования), и коммерческой лицензии, которая требует платы. Следует отметить, что рынок реляционных баз данных составляет миллиарды (именно миллиарды) долларов в год и влезть на него новичку не так просто. Поэтому код и дистрибутивы этой базы данных распространяются свободно. Это привлекает массу пользователей, не способных платить неподъемные цены за лицензионные копии коммерческих СУБД. С другой стороны, открытый код привлекает массу программистов, которые имеют возможность внести в код дополнения, которые им необходимы. Таким образом, убивается сразу два зайца — набирается аудитория пользователей и практически бесплатно производится тестирование и устранение ошибок в коде. Ориентация на открытый код позволила MySQL стать стандартом де-факто в качестве серверной базы данных на Apache-хостингах (70% от всего хост-парка мира), а с другой стороны добиться функциональности, которая приближается к функциональности MS SQL — флагманской СУБД корпорации Microsoft.

Microsoft

Корпорация Microsoft становилась во времена войны стандартов, поэтому большинство ее усилий направлено на создание максимально несовместимых с остальным миром стандартов.

Замечание

Чего только стоит использование обратного слэша в качестве разделителей путей \, во всем компьютерном мире для этого принят прямой слеш /, а обратный слэш используется исключительно для экранирования спецсимволов. Результатом такого решения является ежедневная игра на нервах миллионов программистов и пользователей программ, портированных из других операционных систем.

Такое поведение и удачная маркетинговая политика привели к полному доминированию на персональных компьютерах. Теперь, когда наступило полное доминирование, дальнейшее продолжение такой политики на фоне возрастания популярности свободного программного обеспечения приведет только к оттоку пользователей на альтернативные операционные системы и программные продукты. Понимая это, Microsoft меняет политику с удушения конкурентов на устранение критических замечаний по поводу ее закрытости, сложности разработки программного обеспечения и непоследовательности.

На момент написания книги Microsoft объявила о переходе на открытые XML-стандарты для офисных программ Excel и Word. Понимая, что вытеснить данные пакеты уже не сможет никто, принято решение, значительно облегчающее жизнь программистам, которые смогут гораздо быстрее и качественнее создавать программы, взаимодействующие с Excel и Word, — это еще более усилит позиции Microsoft на десктопных системах.

Перенос графики на уровень ядра привел к высокой скорости обработки графики в операционной системе Windows по сравнению с конкурирующей системой Window X. Однако невозможность выгрузить графику является серьезным аргументом против использования данной системы в качестве сервера. Еще одним аргументом против использования Windows в качестве серверной платформы является необычайно высокая уязвимость данной операционной системы для "червей" и вирусов. Причиной этого является не столько широкая распространенность Windows, сколько проектные ошибки. Большая часть кода, унаследованного от ранних систем, игнорирует группы привилегий пользователей. Операционная система вынуждена разрешать это для обратной совместимости. Отсутствуют средства управления безопасностью сообщений между GUI-клиентами, а их добавление также нарушило бы обратную совместимость. В результате показатели использования Windows в качестве серверной операционной системы не превышают на сегодняшний момент 20% (сервер IIS), в то время как Web-сервер Apache, используемый главным образом под UNIX, имеет показатель в 70%.

Сбором статистики по использованию Web-серверов занимается компания NetCraft. Просмотреть статистику использования серверов с августа 1995 года по сегодняшний момент можно по адресу http://news.netcraft.com/archives/web_server_survey.html. Пик использования сервера IIS от Microsoft с 2001 по 2003 год, который выбивается из общей картины, связан с агрессивной маркетинговой политикой, направленной на продавливание использования своей операционной системы в качестве серверной платформы. Это привело к незначительному временному снижению использования Apache и практически полному прекращению использования серверов Sun.

Sun

Корпорация Sun является столь же легендарной, как и Microsoft, и специализируется в области серверного оборудования, поставляя готовые серверы, т. е. не просто "железо", как Intel или операционную систему, как Microsoft, а все сразу. Более того, корпорация Sun является местом возникновения языка программирования Java, который благодаря серии грубых технических и маркетинговых просчетов Sun практически потерпел неудачу. На кривых компании NetCraft доля серверов, поддерживающих Java, составляет 2%, и доля эта неуклонно продолжает снижаться. И это несмотря на то, что Java является легендарным языком программирования, в котором были реализованы передовые технологии, с успехом применяемые в других языках программирования. В последние годы Sun также объявляет о снижении объемов продаж серверного оборудования.

На момент написания книги корпорация Sun объявила об открытии исходных кодов коммерческой UNIX-системы Solaris 10. Это действие, вероятно, призвано решить несколько задач. Прежде всего, позволит продавать Sun серверы по более низкой цене — теперь покупатель может использовать сервер с бесплатной операционной системой. Открытый код легендарной операционной системы (настоящий UNIX) позволит привлечь энтузиастов, которые помогут устранять неизбежные в коде ошибки и разрабатывать новое программное обеспечение, которое привлечет дополнительных пользователей. Кроме того, это позволит возродить угасший интерес к технологии Java.

Adobe

Ярким примером спланированного захвата рынка является PDF-формат, разработанный компанией Adobe. Основная цель данного формата — платформо-независимое представление документа — документ выглядит одинаково на любой платформе, будь то UNIX, Windows, Macintosh или любая другая операционная система. Практически вся техническая документация на программное обеспечение и научная документация поставляется в PDF-формате. Коммерческие электронные библиотеки используют для представления документов именно PDF-формат, который заменил HTML-представление.

Успех и всеобщее признание PDF-формата связаны в первую очередь с тем, что компания Adobe распространяет бесплатно программу, позволяющую просматривать PDF-документы — Acrobat Reader. Поэтому любой пользователь, встретивший PDF-документ, может прочитать его, загрузив Acrobat Reader с официального сайта. При этом PDF-формат оправдал надежды

пользователей, документы можно оформлять с превосходным качеством и подвергать их при необходимости защите от копирования.

Однако Acrobat Reader не позволяет создавать PDF-документы, для этого предназначена другая программа — Acrobat, которая является платной. Таким образом, Adobe сделала свой формат стандартом де-факто, вынуждая производителей программного обеспечения и информационной продукции для кроссплатформенного использования покупать их редактор Acrobat.

WinAmp

Данный MP3-проигрыватель изначально был небольшой shareware-программой стоимостью 10 долларов. Спроектированная таким образом, чтобы иметь множество "шкур", программа привлекла многих пользователей, которые могли разрабатывать свои собственные "шкуры", чувствуя свою сопричастность к созданию программы. Разработки пользователей стали выкладываться на официальном сайте WinAmp (<http://www.winamp.com>), который в результате получил огромную популярность и стал приносить значительные дивиденды в виде дохода от рекламы. Владельцы WinAmp решили сделать свой продукт полностью бесплатным, что привлекло еще больше пользователей и сделало их сайт еще более популярным.

Opera

Интернет-браузер Opera во многом использует опыт MP3-проигрывателя WinAmp. В нем также имеется большое число "шкур", которые создают пользователи и выкладывают на официальном сайте. До недавнего времени браузер распространялся бесплатно, но во время использования показывал рекламный баннер, который загружается с <http://www.opera.com>. Пользователь мог заплатить 20\$ за регистрацию браузера, и окно исчезало, или продолжал использовать незарегистрированную версию и просматривал рекламу. Таким образом, владельцы программы имели возможность получать доход сразу по двум каналам, в том числе и от пользователей, которые в принципе никогда не платят за программы. Если оплата интернет-трафика временная, а не мегабайтная, они вряд ли будут прибегать к незаконным способам устранения рекламного баннера просто в силу лени. Такой вид программного обеспечения часто называют Adware. Использование баннеров характерно для сетевых программ, таких как интернет-браузеры, менеджеры загрузок, программы мгновенного обмена сообщениями.

Замечание

Осенью 2005 года владельцы браузера Орега объявили свой продукт бесплатным и получают прибыль только от рекламы со своего сайта и продажи лицензий на свой браузер для мобильных телефонов и КПК. Орега считается одним из лучших и быстрых браузеров для мобильной платформы.

Yandex

Поисковая система <http://www.yandex.ru> является самой популярной поисковой системой RuNet'a. Создание подобных систем требует значительных капиталовложений и зачастую убыточно. Однако несколько лет назад проект Yandex вышел на окупаемость. Это связано с размещением платной рекламы на страницах поисковой системы и платных ссылок, когда за определенную сумму по определенным запросам ссылка на сайт размещается в первых строчках. В результате компания получает прибыль от рекламодателей, конечные же пользователи работают с базой данных Yandex совершенно бесплатно и вряд ли стали бы оплачивать услуги поисковой системы, когда им доступны несколько конкурирующих бесплатных поисковых систем.

Электронные платежные системы

Одной из сложностей, с которыми столкнулись разработчики программного обеспечения, распространяющие его по низкой стоимости в Интернете, являются микроплатежи. Банкам неудобно возиться с мелкими платежами; кроме того, комиссия по ним зачастую превышает сам платеж. Именно поэтому вскоре после расцвета коммерческой деятельности в Интернете появились электронные платежные системы вроде WebMoney, E-gold и т. п. Данные системы позволяют передавать суммы в несколько копеек, при этом с каждого перевода взимается комиссия, которая и составляет доход владельцев платежной системы. Взамен пользователям гарантируется защита их средств, которая гораздо выше, чем при работе с кредитными карточками.

Вместо заключения

Таким образом, характеризуя наиболее удачные проекты, существующие на сегодняшний день, можно выделить следующие важные моменты:

- ориентация на массового пользователя с невысокими доходами, а не на небольшие группы заказчиков, обладающих большими средствами;

- ❑ ориентация на открытый код и бесплатное распространение программной продукции и извлечение прибыли из побочного вида деятельности (рекламы, поддержки, обучения);
- ❑ ориентация на Интернет — мгновенное получение заказанного товара и поддержка по нему привлекает большое число покупателей и становится стандартом де-факто.

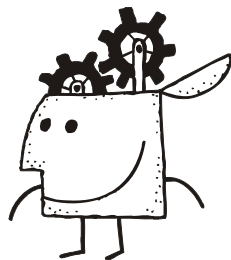
Мир вокруг быстро меняется, область программирования имеет дело с беспрецедентной ситуацией, когда происходит взрывообразное развитие "железа". В таких условиях изменение экономических условий происходит крайне быстро. Читатели должны уметь ориентироваться в ведущих игроках компьютерного рынка и уметь отслеживать события, способные повлиять на его развитие так же, как брокеры биржи отслеживают события на мировой арене и прогнозируют цены на нефть.

Замечание

Цены на нефть, рост экономики и прогнозируемую инфляцию, кстати, тоже неплохо отслеживать — именно это определяет, будут у ваших потенциальных покупателей деньги на программное обеспечение или нет. Возможно, вам выгоднее сосредоточиться на привлечении аудитории бесплатным проектом.

Необходимо отслеживать решения, принимаемые ведущими производителями "железа" и программного обеспечения, а также судьбу программных продуктов, которыми вы пользуетесь — не продан ли какой-либо из них сторонней фирме или корпорации, не поглощена ли фирма-производитель более крупной фирмой.

Глава 8



Проектирование пользовательского интерфейса

Главным императивом разработки интерфейсов программных продуктов является их незаметность. Программа должна позволять выполнять работу, не привлекая к себе внимание. Вспомните, как раздражает вас реклама по телевидению, вы смотрите передачу, и вдруг вас отрывают от цели и привлекают ваше внимание совершенно к другому — у многих это вызывает дискомфорт и приводит к переключению на другой канал или даже выключению телевизора. Точно так же и деталь программы, приковывающая внимание пользователя и отвлекающая от работы, вызывает раздражение. Это не является особенностью программы или ее изюминкой, как часто тешат себя разработчики, любясь своим умением программировать — это ошибка проектирования программного интерфейса. Она не приведет к тому, что пользователи будут забрасывать вас требованиями исправить ошибку (хотя это тоже возможно), но они сменяют инструмент при первой возможности. Если интерфейс будет четким и ортогональным, а программа незаменимой вещью, в привязанности к которой пользователи не будут отдавать себе отчета, — за нее будут цепляться до последнего.

Для программистов наиболее близкой к этому аналогией является язык C и операционная система UNIX, которые пережили не один десяток аппаратных архитектур, операционных систем и языков программирования. Это достигается их жесткой ортогональностью и минимализмом. Идея оказалась настолько живучей и плодотворной, что энтузиасты разработали несколько вариантов UNIX-подобных систем с нуля, просто чтобы создать что-то подобное. Язык C и его объектно-ориентированное развитие C++ являются стандартами промышленного программирования, которым в ближайшее время не предвидится замены.

Как в проектировании, так и в интерфейсе создание двух альтернативных решений проблемы в рамках одной задачи может привести к дискомфортной работе пользователей и значительным затруднениям при сопровождении программ. Улучшив одну часть интерфейса, велик соблазн забыть про

другую; кроме того, неортогональный интерфейс будет здорово смущать пользователя и затруднять его обучение программе. В результате будет наблюдаться отход от главного императива разработки интерфейса "Программа не должна быть заметной".

Компьютер — это средство, а не цель

Многие программисты настолько увлечены компьютером, что отказываются понять и принять, что в восприятии других людей — это серенькая жужжащая коробочка, которая стоит определенных денег и без нее, к сожалению, никак нельзя обойтись. Если бы она не решала проблем пользователей, они тут же выкинули бы компьютер, а на освободившееся место поставили вазу с цветами, ну или навалили бы в беспорядке книги, в любом случае место, занимаемое компьютерами, было бы выгодно использовано. Для программиста такое поведение кажется бредом, тем не менее, это открывает для пользователя множество преимуществ: не болят глаза и остальные части тела от неподвижного сидения за компьютером, нет излучения, про которое производители мониторов говорят, что его нет, но им никто не верит, не нужно запоминать массу технической информации... Список можно продолжать.

Со времени первых открытий человечества все изобретения были направлены на решение проблем человечества, и люди тут же забывали о предметах цивилизации, как только изобретались более совершенные орудия.

Вряд ли сейчас можно найти человека, который пользуется солнечными часами, так как они обладают перед наручными следующими недостатками:

- ❑ стационарность — их нельзя перемещать; чтобы узнать время, нужно постоянно подходить к ним, в то время как наручные часы можно всегда носить с собой;
- ❑ ненадежность — определить время в отсутствие прямого солнечного света невозможно, т. е. солнечные часы работают менее надежно, чем наручные.

Возможно, это покажется надуманным примером... Выйдите в деловой центр вашего города и понаблюдайте молодежь в возрасте от 16 до 30 лет — сколько из них имеют наручные часы? Три процента. С массовым распространением сотового телефона, который стал неотъемлемым атрибутом любого современного человека, наручные часы стали менее удобными, так как обладают рядом недостатков по сравнению с сотовым телефоном, который тоже показывает время:

- ❑ наручные часы не светятся в темноте (в большинстве случаев флуоресцирующей подсветки хватает максимум на 2 часа), в то время как сотовые телефоны имеют подсветку;

- ☐ сотовый телефон все равно нужно носить с собой, зачем два источника времени?
- ☐ при загаре на месте от браслета остается белый ободок;
- ☐ не все часы водонепроницаемые, и попадание под дождь может привести к их поломке;
- ☐ это старомодный способ определения времени, все будут по сотовому время определять, а я как старый пень/старая дева (нужное подчеркнуть) по часам?

Список можно продолжать... Главная идея заключается в том, что большинству людей не нужны солнечные, наручные часы и сотовые телефоны — им нужно решение проблем. Конечно, всегда останутся энтузиасты, которые будут с упоением возиться с сотовым телефоном, носить наручные часы, даже если будет объявлено, что завтра произойдет коллапс Вселенной и по следствию из общей теории относительности время перестанет существовать, и даже те, кто будет продолжать лелеять у себя на даче солнечные часы. Но такие группы будут всегда оставаться в меньшинстве, как и сообщество программистов по сравнению с сообществом пользователей. Программистов, которые смотрят на программу с точки зрения компилятора, всегда будет меньше, чем пользователей, которые смотрят на программу с точки зрения человека. Только умение поглядеть на программу с точки зрения пользователя позволит программисту создавать изящные и востребованные программы.

Если программа вызывает восхищение у программиста, но не работает, ни один здравомыслящий человек не назовет программу программой, а программиста — программистом.

Если программа работает, но вызывает отвращение у пользователей (сложно использовать, требуется совершать большое число монотонных операций), — программа содержит ошибки в пользовательском интерфейсе. Ошибки интерфейса следует рассматривать как ошибки кодирования, так как пользователи — это не увлеченные программисты, которые простят неудобство работы за красивый код. Они просто не будут пользоваться программой, как ни один здравомыслящий человек не будет носить сломанные наручные часы.

Списывать жалобы пользователей на их недостаточную квалификацию не стоит — так как зачастую пользователи больше времени проводят за программой. Если программист работает с пользовательским интерфейсом только при тестировании программных блоков, то пользователи выполняют работу, причем в гораздо большем объеме. Кроме того, цели пользователей отличаются от целей программистов. Пользователи стремятся выполнить работу, а программист создать программный продукт, целью

которого будет выполнение работы. Поэтому в плане внесения изменений в пользовательский интерфейс последнее слово должно быть за конечными пользователями.

Виды интерфейсов

В настоящий момент в мире широко используются два вида интерфейса программного обеспечения:

- ☐ интерфейс командной строки;
- ☐ графический интерфейс.

Замечание

В литературе графический интерфейс пользователя часто называют GUI (Graphical User Interface), в то время как интерфейс командной строки часто называют CLI (Command Line Interface).

Исторически интерфейс командной строки появился раньше графического интерфейса и ассоциируется с операционной системой UNIX. Это связано с тем, что компьютер предназначен для обработки символьной информации, поэтому использование операций с короткими емкими командами было более предпочтительно на первых медленных и неповоротливых компьютерах. С увеличением вычислительной мощности компьютеров появилась возможность использования альтернативных способов ввода, например, с использованием дружественного графического интерфейса.

Замечание

Компьютеры не позволяют в настоящий момент осуществлять взаимодействие с помощью речи, но в полной мере используют звуковое сопровождение. Также не получает широкого распространения комбинация визуального восприятия, речи, слуха и тактильных ощущений (виртуальная реальность), хотя сомневаться в том, что при возрастании мощности компьютеров человечество предпочтет более удобный для себя способ общения, не приходится. Так как пользователям нужны не сами компьютеры или интерфейсы — им нужно решение их проблем, и если их решение при помощи альтернативных средств осуществлять удобнее — они выберут именно их.

Принцип интерфейса командной строки заключается в следующем: пользователь набирает имя программы в специальной среде, называемой оболочкой (shell), и передает программе параметры, которые влияют на ее работу. Пользователь может комбинировать программы, перенаправляя результат работы одной программы на вход другой программы при помощи конвейера. Часто повторяющиеся или сложные операции, состоящие из вызова

многочисленных программ и их параметров, можно оформлять в виде скриптов, которые сами могут включаться далее в другие скрипты.

Командная строка и программирование неотделимы друг от друга, взаимодействие с компьютером сводится к постоянному программированию его действий. Главными недостатками такого подхода являются:

- необходимость изучения сложного и объемного материала по командам и их параметрам;
- длительный набор на клавиатуре;
- высокая квалификация пользователя и, как следствие, длительное время его обучения.

К достоинствам командной строки обычно относят:

- низкую потребность в ресурсах;
- высокую скорость и продуктивность работы;
- возможность использования ранее разработанных программ повторно для решения самых неожиданных задач, для которых программы изначально не предназначались;
- хорошая масштабируемость, операции легко автоматизировать.

Достоинства и недостатки определяются традициями разработки программ в стиле командной строки.

Графический интерфейс пользователя реализует модель рабочего стола — т. е. информация на компьютере представляется пользователю не в виде набора дисков, директорий и файлов, а в виде модели графического стола, в котором расположены папки и документы. Основной упор делается не на программы, как в интерфейсе командной строки, а на документы, при выборе которых программы-обработчики вызываются автоматически. Сами документы часто представляются в WYSIWYG-стиле (What You See Is What You Get — "что видишь, то и получишь") и одинаково выглядят как на экране компьютера, так и на листе бумаги после распечатки, в противовес интерфейсу командной строки, где предпочтение отдается разметке документа и отображению всех подробностей форматирования. Большинство управляющих функций переносятся на мышь, которая все более и более становится похожа на универсальный управляющий манипулятор, обрастая колесиками и дополнительными кнопками. Для того чтобы уменьшить количество перемещений руки с мыши на клавиатуру, клавиатурные комбинации либо вообще не используются, либо используются мало, либо создаются таким образом, чтобы их можно было осуществлять одной левой рукой (<Ctrl>+<S>, <Ctrl>+<Z>, <Ctrl>+<Shift> и т. д.).

К недостаткам графического интерфейса пользователя относят:

- ❑ непрозрачность, сам графический интерфейс часто не обеспечивает решения всех задач, так как его философия поощряет сокрытие сложных деталей, но решение их необходимо и требует вскрытие интерфейса и углубление в детали реализации;
- ❑ сложная масштабируемость и автоматизация. Скрипты не характерны для графического интерфейса, так как нарушают целостность концепции. Графические программы стараются создавать монолитными, возможности использования их в скриптах или других программах не предусматривается. Это приводит к тому, что код дублируется в каждом новом приложении. Каждая программа превращается в универсальную глыбу в противовес UNIX-системам, где стараются разрабатывать небольшие программы с ясными целями, а сложные задачи решать путем комбинации программ. Кроме этого, создавая монолитные универсальные программы, никак не взаимодействующие с другими программами, программист обрекает пользователя на многотысячекратное щелканье мышью, даже если пользователь вполне компетентен, чтобы решить задачу путем программирования;
- ❑ сложность реализации. Создавать программное обеспечение с графическим пользовательским интерфейсом много сложнее, чем программы с интерфейсом командной строки. Это приводит к более длительным срокам разработки программ и к большему числу ошибок в конечном продукте.

К достоинствам графического пользовательского интерфейса относят:

- ❑ простоту использования компьютера конечным пользователем;
- ❑ простоту обучения взаимодействию как с компьютерами, так и с новыми программами;
- ❑ привлекательность для пользователя;
- ❑ привлечение нетехнических пользователей, в результате чего и была совершена компьютерная революция в 90-х годах прошлого века.

Графический интерфейс пользователя был впервые реализован для компьютера Macintosh, а затем в операционной системе Windows. UNIX-подобные операционные системы также обладают своим графическим интерфейсом, реализованным в системе X Window. Наиболее выдержанной в стиле графического пользовательского интерфейса является система Macintosh, которая даже не предоставляет пользователю интерфейса командной строки далее следует операционная система Windows, которая реализует графику на уровне ядра, как и Macintosh, но предоставляет также и режим командной строки, который в последних операционных системах даже стал развиваться. UNIX-подобные операционные системы могут снабжаться системой

X Window по желанию пользователей. X Window представляет собой надстройку над операционной системой как ранняя версия Windows 3.x была надстройкой над DOS. Однако более продумана с конструкторской точки зрения. X Window — это сервер, выполняющий операции по работе с графикой, к которому обращаются клиентские программы, когда требуется отрисовать элементы графического интерфейса на мониторе. Это позволяет развивать независимо сам сервер и его графическую часть и соответствует идее UNIX о поддержке механизма, а не политики. Данный подход, диаметрально противоположен принятому в Windows и Macintosh, которые навязывают пользовательский интерфейс, непосредственно встраивая его в систему.

Выбор операционной системы, ее критика и восхваление определяются задачами пользователя. Когда графический интерфейс пользователя встроен в операционную систему и загружается даже в том случае, если компьютер используется в качестве сервера и вообще не имеет монитора, это будет вызывать раздражение у пользователя, для которого важна бесперебойная работа сервера. Графический балласт в памяти будет занимать столь необходимую серверу оперативную память, и невозможность отключения столь ресурсоемкого сервиса, является nonsensом для серверных операционных систем. Однако конечный пользователь будет скорее раздражен зависаниями системы и медленной работой графики, если она реализована в виде надстройки над операционной системой. Так как ему гораздо важнее стабильная работа графической оболочки (что достигается встраиванием ее в операционную систему), чем возможность отключения графического интерфейса своей машины.

Командная строка

Как правило, программы с CLI-интерфейсом разрабатываются либо для использования в других программах, либо для операционных систем, в которых использование командной строки является традиционным, например, в UNIX.

При создании программ с интерфейсом командной строки нет такого размаха и разнообразия, как в случае графического интерфейса. Однако это не значит, что интерфейс не должен быть тщательно проработан и согласован. Пользователи и программисты, применяющие программу в скриптах, не должны испытывать дискомфорт при работе с командной строкой.

Учитывая, что пользователями командной строки, как правило, являются либо программисты, либо крайне ортодоксальные в интерфейсе пользователи UNIX-подобных операционных систем, следует учитывать традиции командной строки, которые оттачивались на протяжении более чем 30 лет.

Так как интерфейс командной строки продолжает использоваться и развиваться только в UNIX-подобных средах, дальнейшее описание и рекомендации будут касаться в основном UNIX-модели командной строки.

Основным способом воздействия пользователя на режимы работы программы в командной строке является передача ей ключей и параметров. В листинге 8.1 приводится пример вызова программы `mysql` с передачей ей ключа `-u` и параметра `root`, обозначающего имя пользователя.

Листинг 8.1. Передача ключей и параметров программе

```
mysql -u root
```

Параметр может представлять собой ключ или информацию (например, имя файла), которая следует, как правило, за ключом. Существуют два вида ключей:

- традиционные ключи;
- ключи в стиле GNU.

Традиционные ключи состоят из одной буквы с предшествующим дефисом. Причем ключи можно комбинировать, так, если программа принимает ключ `-a` и ключ `-b`, то она, как правило, допускает объединение ключей в последовательность `-ab` или `-ba`.

Использование таких коротких ключей связано с тем, что UNIX в начале своего развития использовался на машинах с медленными телетайпами — чем короче был ввод, тем быстрее можно было работать с программой.

В GNU-стиле вместо однобуквенных ключей используется полная расшифровка параметра, которому предшествуют два дефиса (`--`). Так, традиционный запрос на вывод версии программы может выглядеть как `--version`, в то время как в традиционном стиле для этого бы использовался параметр `-v`. Данный стиль развивался значительно позднее, когда медленные компьютеры с телетайпами сменились быстрыми с клавиатурами. Однако главным стимулом перехода на полный режим явилась нехватка однобуквенных ключей для сложных программ. Ключи в GNU-стиле не допускают комбинации.

Современные программы, как правило, дублируют ключи, т. е. все ключи выполняются в GNU-стиле, а наиболее часто используемые дублируются дополнительно однобуквенными ключами. Однобуквенные и ряд GNU-параметров имеют традиционное использование (см. табл. 8.1), и назначение им других функций приведет к тому, что пользователи вряд ли смогут приспособиться к программе. При разработке пользовательского интерфейса его следует создавать предсказуемым. Таблица 8.1 как раз отражает ожи-

дания пользователей командной строки. Придерживаясь традиционного обозначения параметров, можно облегчить процесс запоминания параметров и уменьшить срок обучения работе с программой, так как пользователь может иметь опыт работы с похожей программой, в которой используются те же самые параметры.

Замечание

Таблица 8.1 частично приводится по книге Э. Р. Реймонда "Искусство программирования для UNIX", которая является прекрасным пособием по философии UNIX-программ и рекомендуется к изучению каждому начинающему UNIX-разработчику.

Замечание

В описании однобуквенных ключей в скобках приводится их английская расшифровка, которая, как правило, используется в ключах в GNU-стиле. То есть если ключ `-c` используется в контексте создания файла, то он, как правило, имеет GNU-эквивалент `--create`.

Таблица 8.1. Традиционный смысл ключей командной строки

Ключ	Описание
-a	Означает слово "все" (all), как правило, не принимает параметров и используется совместно с программами, осуществляющими вывод в консоль. Часто образует пару с ключом <code>-d</code> для удаления
-b	Размер буфера или блока. Устанавливает размер буфера или размер блока
-c	1. Команда (command). Если программа является интерпретатором, то данный ключ позволяет передать строку с командой, которую следует интерпретировать. 2. Проверка (check). Проверяет корректность синтаксиса конфигурационного файла или файла ввода данных. 3. Создать (create). Сообщает программе о необходимости создания файла или записи в базе данных
-d	1. Удаление (delete). Удаление какой-либо информации. 2. Отладка (debug). Устанавливает уровень подробности отладочных сообщений
-D	Определить (define). Устанавливает значение макроопределения в интерпретаторе или компиляторе или в приложении с функциями макропроцессора. Например, для отладочной версии устанавливается, как правило, значение <code>DEBUG</code> , а для конечного выпуска <code>RELEASE</code>

Таблица 8.1 (продолжение)

Ключ	Описание
-e	1. Выполнить (execute). Программы, которые являются упаковщиками или могут использоваться как упаковщики, часто допускают этот ключ для определения программы, которой они передают управление. 2. Редактирование (edit). Программа, способная открыть ресурс либо в режиме только для чтения, либо в режиме редактирования, может принимать данный ключ для активизации режима редактирования
-f	1. Файл (file). Используется для указания файла входных (выходных) данных. Как правило, традиционно для этих целей используются символы перенаправления > и <, а ключ -f применяется тогда, когда возможностей символов > и < недостаточно. 2. Форсировать (force). Форсирует некоторые операции
-h	1. Заголовки (headers). Включает, подавляет или модифицирует заголовки табличных отчетов, сгенерированных программой. 2. Помощь (help). Выводит справочную информацию. Более характерен для консольных Windows-программ, чем для UNIX-программ, где документация хранится централизованно в виде man-страниц или info-страниц. 3. Хост (host). Данный ключ характерен для сетевых программ и указывает сетевое имя или IP-адрес хоста, с которым необходимо установить соединение
-i	Инициализировать (initialize). Устанавливает некоторый критический ресурс или базу данных, связанную с программой, в первоначальное состояние
-I	Включить (include). Добавляет имя файла или каталога в список просматриваемых приложением ресурсов. Данный параметр характерен для программ-обработчиков файлов, например, для компиляторов
-k	1. Хранить (keep). Подавляет обычное удаление какого-либо файла, сообщения или ресурса. 2. Уничтожить (kill). Аварийное и быстрое завершение
-l	1. Вывести список (list). При работе программы с массивом объектов, например, с файлами в каталоге или элементами базы данных данный параметр позволяет вывести их полный список. Там, где данные и так выводятся, параметр -l, как правило, означает более подробный список (long). 2. Загрузить (load). Позволяет загрузить объект, например, динамическую библиотеку. 3. Регистрация в системе (login). Позволяет передать имя пользователя

Таблица 8.1 (продолжение)

Ключ	Описание
-n	1. Число (number). Используется для передачи программе разнообразной числовой информации, например, диапазонов номеров страниц. 2. Нет (no). Подавляет вывод вопросов программы, на которые пользователь должен дать ответ "да" или "нет". При передаче данного параметра считается, что пользователь дает на все вопросы ответ "нет"
-o	Вывод (output). Используется для того, чтобы указать программе файл для вывода результатов работы
-p	Порт (port). Используется для указания номера порта для сетевых программ
-q	Подавление вывода (quiet). Подавляет отображение результатов или диагностической информации
-r	1. Рекурсия (recurse). Если программа осуществляет операции над файлами каталогов, данный параметр указывает необходимость рекурсивного спуска по дереву каталогов и применения операции к каждому файлу, в том числе и находящемуся во вложенной директории. 2. Реверс (reverse). Данный параметр предназначен для обозначения обратного действия программы
-s	1. Подавление вывода (silent). Подавление отображения результатов или диагностической информации. 2. Тема сообщения (subject). Используется в командах, которые отправляют или манипулируют почтовыми сообщениями или новостями
-t	Тег (tag). Передает программе строку, которая используется в качестве ключа поиска
-u	Пользователь (user). Передает программе имя пользователя или его числовой идентификатор
-v	1. Вывод подробных сведений (verbose). В противовес ключам -q и -s используется для вывода более подробных сведений или отладочной информации. 2. Версия (version). Отображает версию программы
-w	1. Ширина (width). Главным образом используется для определения ширины в форматах вывода. 2. Предупреждение (warning). Включает диагностические предупреждения или подавляет их
-x	Извлечь (extract). Передается программе для осуществления операции извлечения, например, если программа работает с файловыми архивами

Таблица 8.1 (окончание)

Ключ	Описание
-y	Да (yes). Подавляет вывод вопросов программы, на которые пользователь должен дать ответ "да" или "нет". При передаче данного параметра считается, что пользователь дает на все вопросы ответ "да"
-z	Разрешает сжатие (zip). Часто данный ключ используется в программах архивирования и резервного копирования

Длительный ввод данных с клавиатуры, как правило, утомителен, поэтому при разработке программы действия, выполняющиеся часто, стараются назначать программе по умолчанию, а редко выполняемым задачам назначать специальные параметры. С другой стороны, пользователи, предпочитающие командную строку, как правило, достаточно хорошо владеют десятипальцевым набором, поэтому управляющие клавиши не должны далеко отстоять от основного клавиатурного ряда, над которым постоянно находятся пальцы оператора.

Кроме того, должны быть способы быстрого исправления ошибок в командной строке. Так, командная строка в старых версиях операционной системы Windows и в DOS не позволяла исправлять набранный текст без удаления всего текста до позиции, где совершена ошибка. Для исправления необходимо было удалить часть команды, исправить ошибку, а затем набрать команду опять до конца. Такой подход требовал много времени по сравнению с современной командной строкой в UNIX-подобных операционных системах, реализуемых различными оболочками, например, bash, где при помощи клавиш со стрелками <←> и <→> можно передвигаться по всей командной строке и исправлять любой символ, не затрагивая другие части.

Для удобного использования командная строка должна иметь возможность вызова ранее введенных команд, которые, как правило, сохраняются в журнал и могут быть вызваны клавишами со стрелками <↑> и <↓>. Более того, современные оболочки позволяют облегчить набор команд, восстанавливая команду лишь по нескольким набранным символам. Для активизации данной функции, как правило, используется клавиша <Tab>. Если по введенным символам команда определяется однозначно, оболочка добавляет окончание команды. Если оболочка находит несколько подходящих вариантов, они выводятся в качестве подсказки, чтобы пользователь смог продолжить набор и уточнить, какую из команд он имел в виду.

Основным преимуществом программ, обладающих интерфейсом командной строки, перед графическими монолитными программами является возможность использования их в сценариях и вызовов из других программ. Этот

вопрос пересекается с проектированием программного обеспечения. Там, где потребности пользователя возрастают настолько, что работа с графическими программными проектами превращается в рутинную работу, напоминающую работу на конвейере, требуется автоматизация. Программы с интерфейсом командной строки легко поддаются включению в различные сценарии и вызовам из других программ, графические монстры тяжелее. Вам когда-нибудь приходилось создавать 200 документов Word, используя текст из вашей программы? Ничего сложного: через технологию OLE требуется вызывать Word и передать текст заранее написанным на Visual Basic for Application скриптом, который выполнит эту работу, или попытаться создать doc-файл по доступной обрывочной информации о внутреннем устройстве данного формата. Не стоит говорить, что программисты не связываются с такими сложностями и предпочитают просто не решать эту задачу, обрекая конечного пользователя на многочасовое копирование текста из одной программы в другую, со всеми связанными с этим ошибками. В мире командной строки такую задачу проще решить, передав в качестве аргумента необходимый текст. Программное взаимодействие с графическими многофункциональными пакетами много сложнее, чем с простой командной утилитой, выполняющей строго одну задачу. Это связано с тем, что при проектировании разработчики ориентируются на нетехнического пользователя и автоматически исключают простые способы взаимодействия, чтобы не смущать начинающих. В результате работа программиста с этой программой превращается в осаду хорошо укрепленной крепости, а конечный пользователь вынужден выполнять рутинную работу. Поэтому переход кампании Microsoft на открытый XML-формат файлов своего офисного пакета можно только приветствовать. Занимая практически монопольное положение на рынке, используя закрытый бинарный формат для файлов, кампания здорово замедляет развитие офисного программного обеспечения.

При разработке программ с интерфейсом командной строки следует помнить, что пользователи очень консервативны, а любые отклонения от общепринятых стандартов будут жестко критиковаться. Это связано с тем, что пользователи являются либо программистами, либо очень квалифицированными пользователями с большим опытом, не считающими в отличие от новичков, что если что-то работает необычно, то на это есть причины, ведомые только программисту.

Графический интерфейс пользователя

Графический интерфейс пользователя призван облегчить использование компьютера. Однако сложность никуда не денется, она либо будет заложена в программу, либо в интерфейс, либо перетечет еще куда-то. За простоту рабо-

ты пользователя, как правило, расплачиваются программисты, которые вынуждены разрабатывать очень сложные монолитные программы, практически не взаимодействующие с другими программами. Редко удается перенести всю сложность программы с интерфейса во внутреннюю структуру, чаще она остается замаскированной в том, что пользователь не может решить часть задач или решает их неэффективно, например, щелкая несколько тысяч раз мышью или осуществляя многочисленные операции копирования.

Главным императивом разработки программного обеспечения является **"незаметность"** программ. Следует помнить, что пользователь при помощи программы решает свои задачи, а не использует программу ради самой программы. Работая с программой, пользователь должен видеть перед собой только работу, не замечая самой программы, она не должна отвлекать его ничем. Вычурные яркие цвета, лишние детали, не имеющие отношения к работе, будут неизбежно привлекать внимание и раздражать пользователя, так как человек может эффективно работать только над одной задачей. Следует по возможности использовать системные шрифты, пользователь привык к ним, и замечает только текст, а не сами шрифты. Часто начинающие разработчики считают, что вычурный шрифт 50 кегля будет гораздо красивее смотреться по сравнению с убогим системным шрифтом. Возможно, при первом просмотре пользователь восхитится смелым дизайнерским решением, но при повседневной работе нестандартный шрифт будет только раздражать. Не следует забывать, что Microsoft были вложены миллионы долларов в разработку пользовательского интерфейса, и зачастую он очень хорошо проработан.

Не следует ограничиваться разработкой программного обеспечения под один монитор. Это касается как разрешения (очень часто приложения даже очень известных фирм "плывут" при нестандартных разрешениях), так и типов мониторов — приложение или сайт, которые прекрасно выглядят на CRT-мониторе, могут совершенно ужасно смотреться на LCD-мониторе. На рис. 8.1 приведено диалоговое окно смены дизайна утилиты ATI Catalyst Control Center. Разработчики приняли правильное решение, предоставив такое диалоговое окно, так как при частой работе красивый, стильный дизайн с множеством градиентов отвлекает внимание от основной задачи, и отвлекает настолько сильно, что автору потребовалось несколько минут, чтобы понять, что все кнопки и настройки в этой форме вышли за границы формы и сменить навязчивый дизайн не представляется возможным.

Замечание

Как и в любой области, любое правило может быть нарушено с целью более эффективного решения задачи. При возникновении критической ошибки, которая может привести, например, к потере данных пользователем, важно очень громко и настойчиво привлечь внимание пользователя выводом диало-

гового окна или надписи, выполненной ярким красным цветом, а не тихим помещением записи в журнал. Такое контрастное нарушение правила "незаметности" только усиливает общее хорошее впечатление от пользовательского интерфейса.

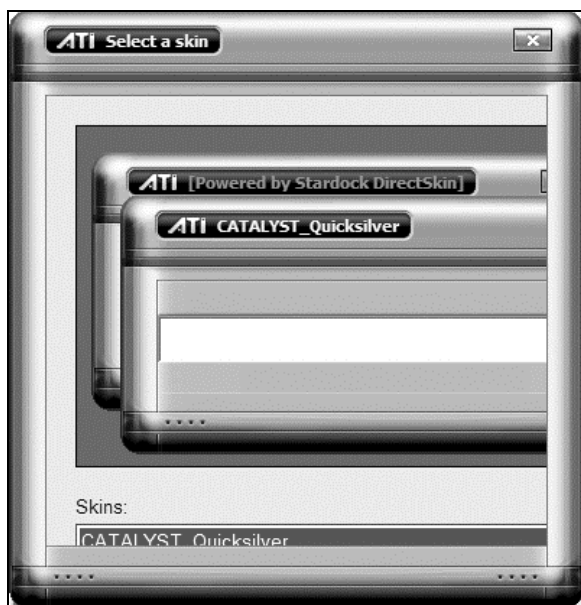


Рис. 8.1. Диалоговое окно ATI Catalyst Control Center

Основными достоинствами графического пользовательского интерфейса являются его **предсказуемость** и **интуитивность**. Обнаружив корзину на рабочем столе, пользователь может догадаться, для чего она нужна. Даже если он не догадается о ее назначении, после того, как ему будет продемонстрировано, как ее следует использовать, он вряд ли забудет, так как зрительная память развита лучше, чем символьная. Однако сами по себе графические интерфейсы не являются интуитивными, например, в двойном щелчке мышью никакой интуитивности нет. Интуитивность заключается в том, что после использования нескольких программ пользователь может быстро догадаться, как работать с совершенно новой программой, даже если он с ней до этого никогда не встречался.

Одной из основных причин массового перехода на графический интерфейс пользователя послужило то, что графический интерфейс позволяет оперировать не абстрактными командами, а пусть нарисованными, но объектами. Пользователь видит перед собой модель рабочего стола, на котором лежат объекты, способные перемещаться, почти как в реальной жизни. Причем

организовать расположение объектов можно так, как это требуется пользователю. В обычной жизни все вещи лежат на столе и в ящиках, и вы без труда можете их найти, когда они понадобятся. Сильной стороной графического интерфейса является свобода пользователя в организации виртуального рабочего стола — на нем может быть и безупречный порядок, и хаос. Каждый инструмент и документ лежит на своем месте.

Пожалуй, единственным недостатком при работе с графическим интерфейсом является обучение новых пользователей использованию мыши. Многие пользователи со стажем быстро забывают, как сложно научить новичка использовать мышь и производить операции перетаскивания предметов и двойной щелчок. Особенно трудно даются операции перемещения, так как они требуют одновременного нажатия кнопки и движения мыши. Поэтому в пользовательском интерфейсе стараются снизить число операций перетаскивания и двойного щелчка мыши. Это связано с тем, что при перемещении объектов с нажатой кнопкой, пользователь сосредоточен на двух операциях — удержании пальца и перемещении мыши. Эффективной работа становится только тогда, когда она становится автоматической и доводится до уровня рефлекса. То есть когда внимание полностью переключается на программу, а правая рука сама выполняет нужные действия. Точно так же, добиться большей эффективности при работе с клавиатурой можно только тогда, когда пальцы сами нажимают нужные клавиши. При наборе данного текста я просто формулирую мысль, а руки со скоростью 350 знаков в минуту вводят его — я даже не могу вспомнить, где какая клавиша расположена. Если при наборе текста я пытаюсь вспомнить расположение клавиш — ничего набрать уже не удастся. Сказывается автоматизм — набор становится просто рефлексорным.

Таким образом, пользовательский интерфейс должен быть по возможности **ортогональным** — одно действие — один ответ системы. Манипулирование клавишами клавиатуры при одновременно нажатой правой кнопке мыши не является удачной идеей, если только пользователь не может рефлексорно осуществлять такую операцию. Как правило, такой интерфейс характерен для игр, предназначенных для профессиональных игроков, у которых рефлексy отточены до мелочей. Если вы не ожидаете от своих пользователей такой прыти, лучше строить ортогональный интерфейс, который сводится к нажатию клавиш и перемещению мыши, причем эти действия по возможности не должны зависеть друг от друга.

Замечание

Нарушением ортогональности является сочетание бегущего текста со звуковым сопровождением в играх. Пользователь читает текст, при этом прислушивается к речи, в результате не понимает ни речи, ни текста — хорошей практикой является использование только одного канала передачи информации: либо зри-

тельного, либо слухового. Точно так же, анимация на заднем плане раздражает пользователя, так как он, читая текст, вынужден отвлекаться на видеоинформацию. Именно поэтому анимационная реклама так раздражает пользователей Интернета — она отвлекает сознание от текстовой информации.

При использовании графического интерфейса пользователя важную роль играет мышь, так как позволяет пользователю легко манипулировать графическими объектами. Часто пользователи осваивают очень хорошо либо клавиатуру, при длительном использовании командной строки, либо мышь, при длительном использовании пользовательского интерфейса. Последнее связано с тем, что переместить указатель мыши от одного элемента управления к другому проще, чем 100 раз нажать клавишу <Tab> для последовательного перехода через каждый элемент управления. Если вы создаете игру, ориентированную на начинающих пользователей, имеющих дело только с пользовательским графическим интерфейсом, постарайтесь не задействовать в ней клавиатурные комбинации и вообще клавиатуру. Подсознательно пользователи ее боятся — для них она представляется устройством в 50 раз сложнее мыши (по числу клавиш), поэтому игры, где используется только мышь, пользуются гораздо большей популярностью среди пользователей с малым опытом работы на компьютере.

Однако клавиатура в качестве управляющего инструмента также играет большую роль. Это связано с тем, что скорость работы с клавиатурой, как правило, больше, чем с мышью. Как упоминалось выше, быстрые клавиши в графическом пользовательском интерфейсе сосредоточены в районе левой руки (табл. 8.2), так как правая рука постоянно занята мышью. Для того чтобы снизить число необходимых движений правой руки от мыши к стандартным комбинациям, принятым для IBM, для копирования информации в буфер обмена <Ctrl>+<Ins> и извлечения ее из буфера обмена <Shift>+<Ins>, были добавлены дополнительные комбинации <Ctrl>+<C> и <Ctrl>+<V>.

Таблица 8.2. Быстрые клавиши, принятые в операционной системе Windows

Комбинация клавиш	Значение
<Ctrl>+<Z>	Отменить последнее действие или операцию
<Ctrl>+<X>	Вырезать отмеченный объект
<Ctrl>+<C>	Скопировать отмеченный объект в буфер обмена
<Ctrl>+<V>	Вставить скопированный объект из буфера обмена
<Ctrl>+<A>	Отметить все пиктограммы
<Ctrl>+<S>	Сохранить результаты работы
<Ctrl>+<F>	Вызвать диалог поиска

Именно поэтому в операционной системе Windows смена раскладки клавиатуры производится либо при помощи комбинации <Ctrl>+<Shift>, либо <Alt>+<Shift>. Такое расположение клавиш удобно, если пользователь постоянно использует мышь, но может быть неудобным, если оператор владеет десятипальцевым методом.

В Linux для смены раскладки клавиатуры может быть назначена любая комбинация клавиш, например, правая клавиша <Ctrl>, которую очень сложно задеть случайно. В то же время она позволяет сменить раскладку клавиатуры в одно касание, что очень важно при частой смене языка, например, при написании технической книги. Невозможность настройки собственных комбинаций быстрых клавиш необычайно раздражает.

Большое значение комбинации клавиш имеют при создании игр, где от игрока часто требуется хорошая реакция. Современные игры, такие как "Принц Персии", требуют управления двумя видами оружия (главный герой имеет по мечу в каждой руке) и достаточно сложную систему ведения боя и передвижения.

- Прыжок и управление левым мечом осуществляется клавишами <Пробел> и <E> соответственно.
- Управление движением осуществляется клавишами со стрелками.
- Форсаж для бега по стенам и управление правым мечом осуществляется мышью.

В результате, чтобы эффективно передвигаться и вести бой, игроку требуется три руки: в левой части клавиатуры, на управляющих стрелках и на мыши. К сожалению, большинство пользователей обладают слишком малым количеством рук, и становятся легкой добычей различных монстров и пропастей, преодолеть которые можно, только пробежав по стене, а затем, оттолкнувшись от нее, прыгнув. Обладающие всего лишь двумя руками пользователи помещают правую руку на мышь, левую на стрелки для бега по стене, а для прыжка с отталкиванием от стены затем вынуждены одновременно перенести правую руку на клавиши стрелок клавиатуры, левую руку на пробел, наблюдая падение незадачливого принца в пропасть. К счастью, создатели игр обладают большим опытом в разработке игр и в пользовательском интерфейсе. Обычно все комбинации клавиш поддаются переназначению — это является хорошей практикой, так как позволяет настроить пользовательский интерфейс так, как к нему привык пользователь.

Перенастройку следует использовать и в обычных программах, так как пользователи очень любят "играться" с настройками и считают себя причастными к созданию программы, когда они полностью переопределяют настройки по умолчанию. Разрешив пользователю поставить свое "клеймо" на

приложение, вы заставляете его воспринимать программу как свою собственную.

Однако настройке должно подвергаться не все и вся, чтобы пользователь не мог утонуть в них. Следует так организовывать работу программы, чтобы не нарушать принципа **последовательности**. Например, расположение кнопок часто является стандартом. Кнопки должны быть расположены на привычных местах. Обычное расположение кнопок **ОК** и **Отмена** изображено на рис. 8.2.



Рис. 8.2. Привычное для пользователя расположение кнопок

Если у пользователя действия доведены до рефлекса, то он даже не будет смотреть на названия кнопок, так как четко знает, что кнопка **ОК** всегда расположена слева, а **Отмена** справа. Если при разработке программ опустить данную особенность, ничего кроме дискомфорта при работе с программой добиться не получится. Согласованность должна быть не только с операционной системой, но и на уровне самого приложения. Если в одном месте программы применяется термин "Договор", то для обозначения этого понятия уже недопустимо использовать термин "Контракт" или "Документ". Если вы используете слово "Интернет" (с прописной буквы и не склоняя), то уже недопустимы варианты "интернетом" и "интернет". Обычно обозначение ключевых терминов принимается на уровне спецификаций всего проекта, которых придерживаются все разработчики.

Замечание

Преимущества принципа последовательности прекрасно характеризует цитата из книги Станислава Жаркова "Shareware: профессиональная разработка и продвижение программ": "Последовательный интерфейс понятен и очень легок для освоения, так как при его изучении пользователи не сталкиваются с сюрпризами, и даже те части интерфейса, которые он видит впервые, кажутся ему давно знакомыми. Если же в одних и тех же ситуациях программа реагирует по-разному, элементы управления в формах имеют различную компоновку, а текстовые сообщения удивляют все новыми и новыми терминами, то вся программа станет для пользователя одним сплошным сюрпризом".

Графический интерфейс предназначен для того, чтобы языком картинок передать информацию из тысячи слов. Однако плохая пиктограмма часто требует для своего пояснения куда большего количества слов. Часто работу по созданию графики лучше заказать профессиональной графической сту-

дии, чем создавать ее самостоятельно или переделывать уже готовую. Особенно это касается программистов, которым требуется примерно шесть часов, чтобы нарисовать убожество. Причем затраченное время никак не прибавляет опыта, чтобы нарисовать еще одно убожество, требуется еще шесть часов. Это связано с тем, что если работе с текстом нас всех учат с 1 по 11 класс, а чаще и далее, то работе с графикой учат 3 года, да и то кое-как. Поэтому ничего зазорного нет в том, чтобы обратиться к профессионалу.

Обратная связь

Длительные задержки без уведомления о них пользователя недопустимы — программа всегда должна информировать пользователя о состоянии своей работы: сколько времени прошло от начала процесса копирования файлов, записи CD-дорожки или MP3-файла и сколько процентов работы уже сделано. Кроме того, любое действие пользователя должно сопровождаться визуальным эффектом: нажатие кнопки, деактивация элементов управления, если их повторное нажатие не желательно, изменение цвета статусного окна и т. п. (рис. 8.3).

Замечание

Отображение полосы прогресса не является прерогативой только графического интерфейса. Организация отображения прогресса вполне доступна и из командной строки либо в виде изменяющихся процентов, либо в виде бегущей строки из символов =====. Более того, в Linux в этом случае принято в конце такой полосы чередовать символы \, |, / и —, создавая эффект вращающейся полоски для того, чтобы пользователь знал, что система работает. В графических средах, даже в мобильных устройствах вроде сотовых телефонов, традиционно для этого применяются вращающиеся песочные часы.

Замечание

Джеф Раскин установил эмпирическое правило: время отклика программы при диалоге с человеком может не превышать 0,7 секунд, люди почти не способны заметить более короткую задержку. Таким образом, уменьшение времени отклика программы при диалоге с пользователем до 0,3 секунд является пустой тратой ресурсов.

Выбор средства обратной связи обуславливается типом и важностью информации, которую необходимо сообщить пользователю. Критическую информацию обычно выводят в виде отдельного диалогового окна, которое блокирует работу приложения до тех пор, пока пользователь не прочтет его и не закроет (рис. 8.4).

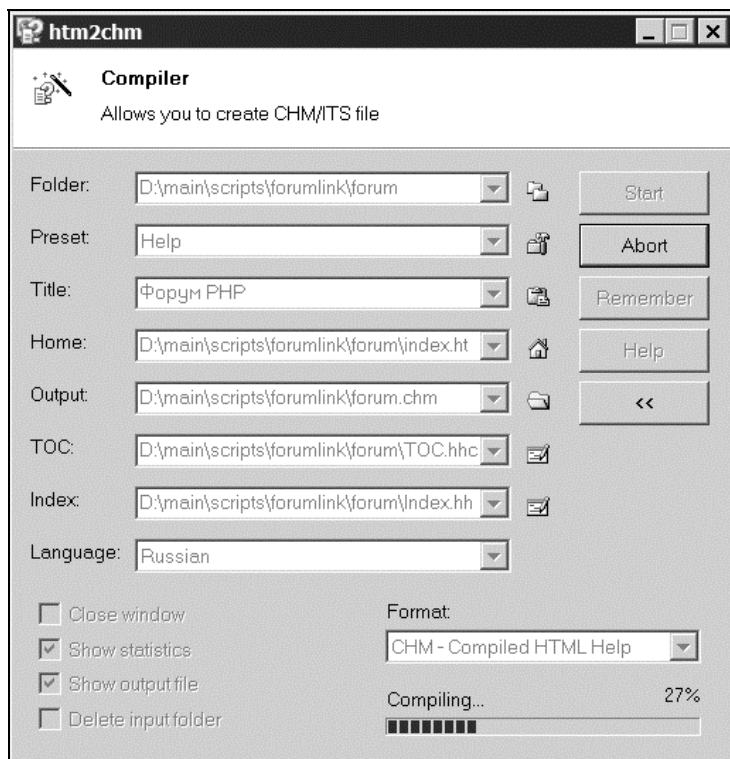


Рис. 8.3. Компилятор chm-файлов на момент компиляции блокирует все элементы управления кроме кнопки **Abort**. При этом в виде процентов и графического элемента управления отображается прогресс компиляции

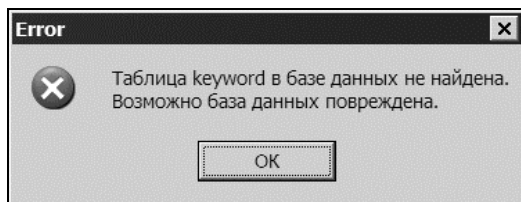


Рис. 8.4. Диалоговое окно

При этом незначительные ошибки обычно выводятся в статусном окне, информируя пользователя о состоянии дел только в том случае, если он заинтересуется этим. Показателен пример почтового клиента Outlook Express, который если не может осуществить загрузку почты из-за разрыва связи или проблем на почтовом сервере, выводит сообщение в диалоговом окне (рис. 8.5).



Рис. 8.5. Сообщение об ошибке
в статусном окне

Если пользователь загружает Outlook Express для того, чтобы просмотреть почту, полученную в предыдущих сеансах связи, не устанавливая соединения с интернет-провайдером, его наверняка раздражало бы постоянно выскакивающее сообщение о невозможности установить соединение.

С другой стороны, если пользователь запустил длительный процесс кодирования большого файла, подсчета числа символов в файле, компиляцию exe- или chm-файла, запись информации на CD-диск, в конце такого процесса результат должен быть выведен в виде отдельного диалогового окна (рис. 8.6).

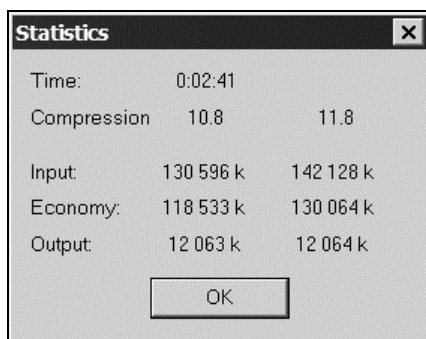


Рис. 8.6. Отображение статусной информации
в конце длительного процесса в виде отдельного диалогового окна

Появление нового диалогового окна "будит" пользователя после длительного процесса и сообщает дополнительные сведения, например, сколько времени было потрачено и какая работа сделана.

Часто для обратной связи с пользователем используют звук, что становится стандартом де-факто в программах с хорошо проработанным интерфейсом. Например, получение письма в упомянутом выше почтовом клиенте Outlook Express сопровождается звуковым эффектом, не позволяющим пользователю пропустить почтовое сообщение. Это дает возможность привлечь внимание пользователя, даже не сидящего за компьютером, а находящегося где-то поблизости.

Особую роль играет звук в играх, где каждый удар и выстрел должны сопровождаться ответным звуком. Причем звуки при попадании в различные предметы (стена, мусорный бак, противник) должны быть разными, это не

только вовлекает пользователя в игровой процесс, но и позволяет успешнее реагировать на быстро меняющуюся игровую ситуацию.

Важно помнить, что звуковое оповещение не должно быть единственным средством организации обратной связи, все звуковые события должны быть продублированы визуальными эффектами. Так, в Outlook Express приход письма сопровождается не только звуком, но и появлением значка "письма" в системном трее (рис. 8.7),



Рис. 8.7. В Outlook Express новое письмо сопровождается не только звуковым эффектом, но и появлением значка письма в системном трее

что не останется незамеченным для пользователей, которые привыкли отключать звук или не имеют звуковой карты на компьютере.

Эмуляция реальных предметов

Преимуществом графического интерфейса пользователя перед интерфейсом командной строки является то, что он моделирует реальность, с которой пользователь привык иметь дело с детства. Именно поэтому графические игры пользуются большой популярностью у новичков при освоении компьютера: помимо того, что игровой процесс очень увлекателен, он больше похож на реальный мир, чем все остальное в компьютере.

Замечание

Концепция эмуляции внешнего мира имеет глубокое значение, это относится не только к графическому представлению, но и к текстовой составляющей. Если описание к программе будет походить на скучную монографию, изобилующую специальными терминами, пользователь вряд ли будет делать дальнейшие попытки разобраться в том, как работает программа. С другой стороны, если описание и текст в самой программе будут содержать орфографические ошибки и сленговые словечки, это также вряд ли понравится серьезно настроенному пользователю.

Графический интерфейс позволяет нарисовать программы, выполняющие функции внешних предметов именно так, как они выглядят в реальности. Утилита связи сотового телефона с компьютером mobile PhoneTools выглядит точно так же, как сотовый телефон, копируя не только расположение клавиш и их форму, но и рельеф корпуса, и элементы дизайна (рис. 8.8).



Рис. 8.8. Утилита mobile PhoneTools

Причем программа принимает форму того телефона, который подключен к компьютеру, т. е. если вместо телефона Motorola C-350 будет подключен телефон Motorola V-500, программа автоматически примет форму V-500. Набор номера телефона на цифровом поле клавиатуры или мышью на клавиатуре программы приводит к набору телефона не только в программе, но и на реальном телефоне, а нажатие клавиши <Enter> аналогично нажатию кнопки дозвона на телефоне. Освоение программы не составляет труда — она полностью имитирует реальный объект, дополняя его виртуальными преимуществами вроде более удобного и более широкого окна для отправки SMS (скопированного, кстати, с интерфейса Outlook Express, к которому привыкло большинство пользователей Windows).

Примеров таких программ множество: проигрыватели компакт-дисков, калькуляторы, часы и т. п. Однако не всегда заимствование, представление из реального мира является удачной идеей. Неудачные решения связаны в первую очередь с тем, что компьютерный интерфейс часто является для опытных пользователей столь же привычным, как и реальные предметы, а с другой стороны, для некоторых операций, широко распространенных при работе с программами, просто не существует реальных аналогов.

В качестве примера нарушения традиций построения программ можно привести MP3-проигрыватель WinAmp. Моим главным затруднением при его использовании на первых порах была невозможность вызвать меню программы. Вывод главного меню при наведении мыши на иконку в правом нижнем углу и нажатии на правую кнопку не является традиционным решением для операционной системы Windows.



Рис. 8.9. MP3-проигрыватель WinAmp

Невозможность для некоторых операций аналогий с элементами окружающего мира можно также продемонстрировать на примере WinAmp. Клавиши управления треками (**Перемотка, Старт, Пауза, Стоп**) полностью дублируют обозначения клавиш в реальных магнитофонах, которые не менялись ни в катушечных, ни в кассетных, ни в CD-, ни в Flash-проигрывателях. Несколько поколений людей привыкли к такому обозначению и без труда осваивают данную панель WinAmp. Однако шестая клавиша, традиционно применяющаяся для выброса кассеты или лотка с CD-диском, в WinAmp приводит к появлению диалогового окна **Открыть файл**. Даже зная об этой особенности WinAmp, я не могу остановить свою руку от нажатия на эту клавишу, если мне необходимо извлечь CD- или DVD-диск из CD-ROM. Поэтому эмуляция реальных объектов не всегда бывает удачной, так как пользователи вынуждены тратить время на изучение нетрадиционного интерфейса.

Многословность программ

Одним из достоинств программ, начиная с появления первых компьютеров, была их немногословность. То есть требование минимальных действий (ввод с клавиатуры, движения мышью, движение глаз, время ожидания ответа, раздумий) от пользователя и минимально необходимого ответа программы.

Это связано не только с тем, что первые компьютеры были малопроизводительными, и требования минимизации были вызваны жесткой экономией ресурсов, а телетайпы, предназначенные для ввода и вывода информации, были настолько медленными, что распечатку длинного ответа можно было дожидаться дольше, чем самих вычислений. Немногословность актуальна и по сей день.

Это связано с тем, что программы, выдающие излишнюю информацию, часто не способны к конструктивному взаимодействию с пользователем, который в потоке информации не может найти результаты или важные сообщения. Беспольные сообщения отвлекают пользователя от решения бо-

лее важных задач. Кроме того, при использовании результатов такой много-словной программы другой программой снижается надежность их взаимодействия — разобраться в большом потоке данных и не совершить ошибку очень сложно. Но даже после того, как задача решена, никто не даст гарантии, что ошибка не вкралась в разбухший код анализатора. К правилам хорошего тона относится сообщение пользователю только важных результатов и предупреждений об ошибках. Это вовсе не значит, что следует пропускать другие сообщения и вообще их не фиксировать, но такие сообщения разумнее помещать в файл журнала, к которому пользователь может обратиться по мере надобности. Такой подход часто применяют в серверах, где сообщений бывает очень много, — сообщения помещаются в отдельные log-файлы. Если сервер обслуживает несколько сайтов и сообщений очень много, имеется возможность разбить log-файл на несколько, каждый из которых соответствует отдельному Web-сайту.

Другой подход, применяемый для скрытия излишних сообщений, это использование нескольких режимов работы приложения. Основной режим работы отличается лаконичностью, отладочный режим выводит многочисленные сообщения и подробности. Для переключения может использоваться либо отдельный параметр командной строки, как параметр `-v` программы-шлюза `fetchmail` в UNIX, выводящий подробную распечатку сеанса с сервером. Это может быть отдельный пункт меню, как в Word для просмотра непечатаемых символов, или пункт меню в Web-браузере для просмотра исходного HTML-кода. Создатели игр и сложных программных продуктов часто организуют возможность создания дампа системы, который можно отправить производителю и по которому он определит наличие ошибки в коде своего приложения. То есть вывод отладочной и вспомогательной информации должен протекать либо в особом режиме программы, либо вызываться пользователем осознанно. Программа не должна сопровождать свою работу выводом большого числа загадочных сообщений.

Как отделить важные сообщения от незначительных? Как правило, критерием выступает тот факт, знает пользователь о том, что ему сообщается или нет. Диалоговое сообщение "Программа запускается, вы согласны или нет?" является примером совершенно бесполезного сообщения, которое заставляет пользователя совершать лишнюю работу. Пользователь, запустивший программу, и так знает, что она должна быть запущена. Точно так же программа не должна выдавать сообщений "Уверены ли вы, что хотите задать программу?" — такие сообщения должны появляться только в том случае, если пользователь может потерять результаты своей работы. В любом случае если такие сообщения выводятся, должна иметься возможность их отключения. В противоположность пользовательскому интерфейсу, для которого применяется правило наименьшей неожиданности (интерфейс не должен удивлять пользователя), сообщения должны удовлетворять правилу наи-

большей неожиданности (должны выводиться только те сообщения, которые сообщают о серьезных отклонениях от нормы).

Замечание

На любое правило существует исключение, например, при включении сотового телефона запрос "Подтвердите включение телефона" нельзя однозначно назвать неудачным пользовательским интерфейсом. Кнопка включения может быть нажата случайно, когда телефон находится в сумке или кармане. Однако невозможность отказаться от этого запроса или создание такой структуры меню, в которой найти эту настройку невозможно, является однозначным признаком плохого пользовательского интерфейса.

В последнее время широко применяется технология сокрытия текстовой информации в редакторе — пользователю самому предоставляется выбрать, какие блоки следует скрывать, а какие должны быть развернуты. На рис. 8.10 представлено окно системы компьютерной алгебры Mathematica 4.0 которая является ярким представителем гибрида интерфейса командной строки и графического пользовательского интерфейса — пользователь получает гибкость мини-языка Mathematica 4.0 и в то же время может выбирать при помощи мыши спецсимволы вроде интегралов, квадратного корня и другие специфические математические символы. Система построена по архитектуре ядро/драйвер и допускает обращение к ядру из пользовательских сценариев командной строки.

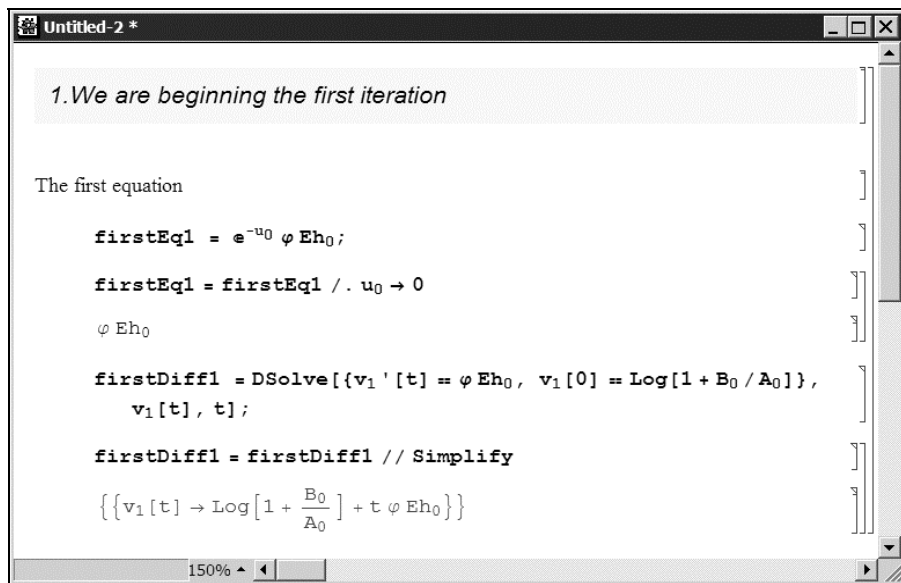


Рис. 8.10. Диалоговое окно системы Mathematica 4.0

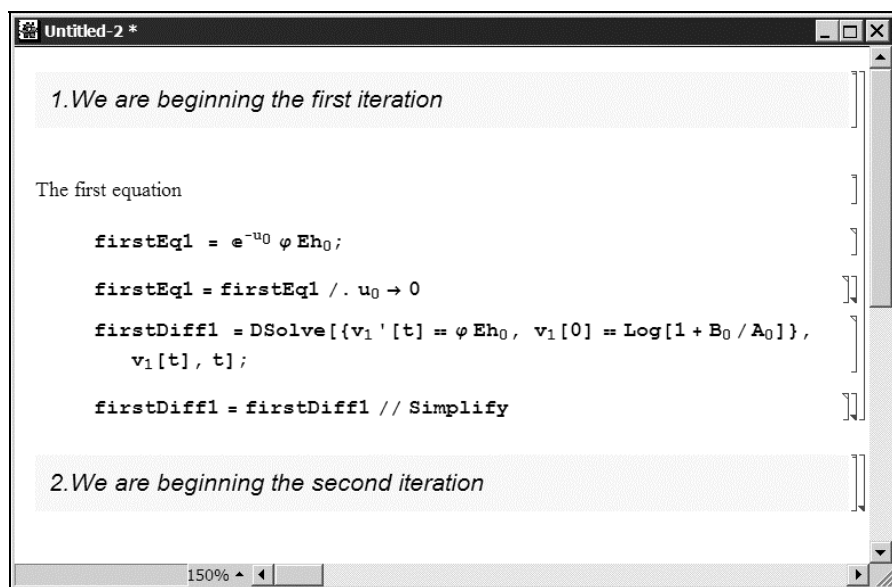


Рис. 8.11. Сворачивание подчиненного текста

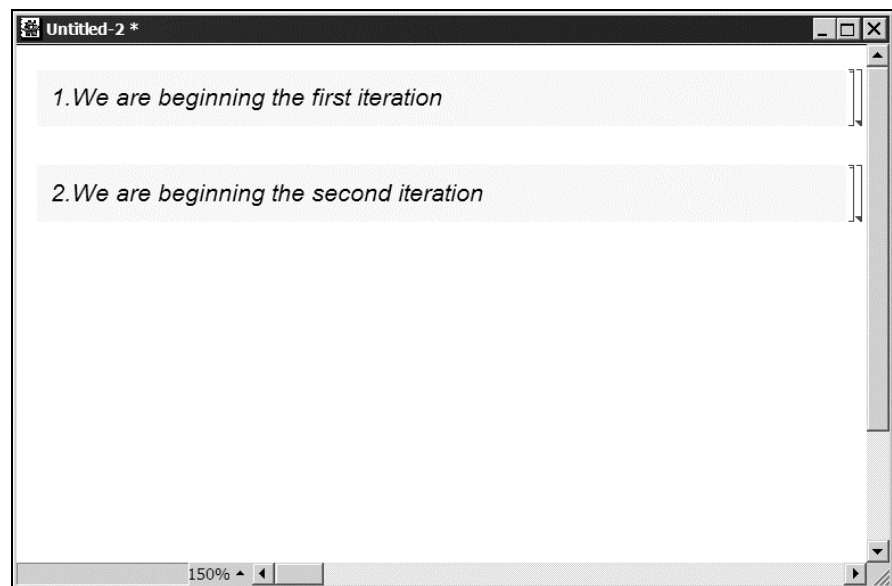


Рис. 8.12. Система сокрытия лишней информации позволяет экономить много пространства

Как видно из рис. 8.10, каждое математическое выражение, результат вычисления или просто сопровождающий текст заключены в квадратные скобки, которые имеют иерархию. Щелчок мышью по такой квадратной скобке приводит к сворачиванию текста (рис. 8.11). Иногда это может приводить к сокрытию большого объема промежуточных вычислений (рис. 8.12).

Пользователь сам выбирает, какую часть документа и в каком объеме он хочет скрыть. Такую же возможность предоставляет редактор среды Visual Studio .NET.

Психологический аспект

При построении пользовательского интерфейса необходимо принимать во внимание тот факт, что с программой будут работать люди. Поэтому при разработке программ следует учитывать некоторые психологические законы, игнорирование которых приводит к резкому снижению производительности оператора и его быстрой утомляемости.

Многословность программы отвлекает пользователя от работы, но совершенно молчать программа не может. Для совершения действий пользователь вынужден удерживать информацию, полученную в одной части программы, для применения в другой.

Статья Миллера 1956 года "The Magical Number Seven, Plus or Minus Two" является классическим когнитивным исследованием, выводом которого является то, что человек способен удерживать в краткосрочной памяти семь объектов плюс минус два.

Замечание

Со статьей "The Magical Number Seven, Plus or Minus Two" можно ознакомиться по адресу <http://www.well.com/user/smalin/miller.html>. Данное правило применяется повсеместно. Именно благодаря ему основной номер в телефонах состоит из семи цифр. Если необходимы дополнительные цифры — вводят код, оставляя при этом число основных цифр равным семи. Более того, телефон 89051235215 никогда не записывается без разделителей, скорее всего он будет представлен в виде 8-905-12-35-215, т. е. будет разбит на пять групп. По этой же причине очень редко встречаются домены выше пятого уровня.

Это означает, что для эффективной работы число элементов в меню не должно быть больше семи (в крайнем случае девяти). Если программа является консольной, то число часто применяемых параметров также не должно превышать семи штук. Наличие в панели инструментов более чем семи кнопок также является поводом для ее разбиения. При этом самих групп также не должно быть слишком много.

Другим важным аспектом в области разработки программного обеспечения является следование **принципу "золотого сечения"**, которое известно с глубокой древности и применяется в архитектуре и искусстве. "Золотое сечение" — это пропорция, наиболее комфортная для зрительного восприятия. Если окно или элемент программы удовлетворяют принципу "золотого сечения", то это способствует появлению ощущения красоты и гармонии.

Замечание

"Золотое сечение" очень широко распространено в природе: его можно найти в пропорциях растений, животных и даже человека. Принцип "золотого сечения" использовался в архитектуре с глубокой древности. Так, древнеегипетские пирамиды, начиная с четвертой династии фараонов, основаны на пропорциях "золотого сечения". Сам термин "золотое сечение" был введен Леонардо да Винчи.

Не вдаваясь в геометрические определения принципа "золотого сечения", можно сказать, что если меньшая сторона прямоугольника принимается за x , то большая сторона должна удовлетворять следующей формуле

$$y = \frac{\sqrt{5} - 1}{2} x.$$

Или, вычисляя постоянный множитель, $y \approx 0,618 \cdot x$. Прямоугольник с отношением сторон, удовлетворяющим принципу "золотого сечения", обладает интересной особенностью. Если вписать в него квадрат, то прямоугольник, полученный в остатке, тоже будет удовлетворять принципу "золотого сечения", и данный процесс можно продолжать бесконечно (рис. 8.13).

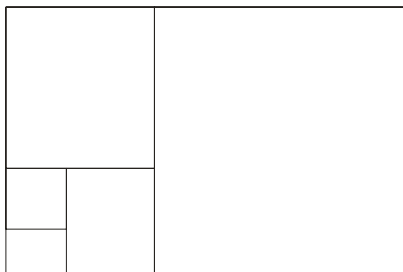


Рис. 8.13. Если прямоугольник удовлетворяет принципу "золотого сечения", то, вписав в него квадрат, в остатке получим новый прямоугольник, удовлетворяющий принципу "золотого сечения"

Многие разработчики предпочитают даже оставлять пустые места, лишь бы сохранить принцип золотого сечения всей формы (рис. 8.14).

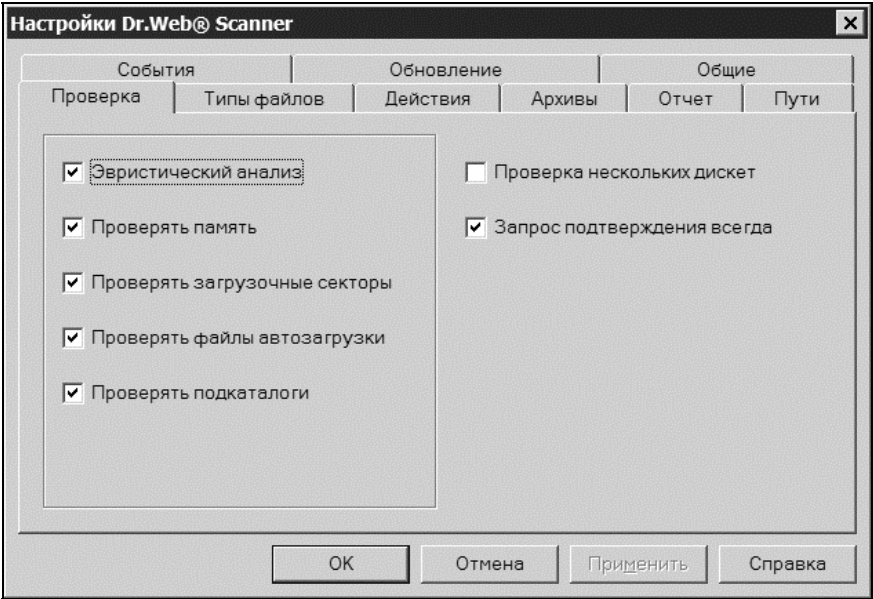


Рис. 8.14. Форма настроек, удовлетворяющая принципу "золотого сечения"

Распространение программы

При формировании дистрибутива конечной программы она, как правило, снабжается различными вспомогательными файлами, по которым пользователь может разобраться, как установить и использовать программу, по какому лицензионному соглашению она распространяется и где можно получить поддержку. Общепринятые названия сопроводительных файлов приводятся в табл. 8.3.

Замечание

Приведенные в табл. 8.3 файлы не являются обязательными и характерны в основном для UNIX-подобных операционных систем.

Таблица 8.3. Общепринятые названия сопроводительных файлов

Имя файла	Описание
README	Файл-путеводитель, который читают первым и который содержит описание программного продукта
INSTALL	Инструкции по конфигурированию, компиляции и установке

Таблица 8.3 (окончание)

Имя файла	Описание
AUTHORS	Перечень участников проекта (GNU-соглашение)
NEWS	Последние новости проекта
HISTORY	История проекта
CHANGES	Перечень значительных изменений между различными редакциями
COPYING	Лицензионные условия проекта
LICENSE	Лицензионные условия проекта
FAQ	Часто задаваемые вопросы по программному продукту и ответы на них

Для операционной системы Windows перечисленные в табл. 8.3 файлы снабжаются расширением txt, позволяющим запускать файл для просмотра в Блокноте. Сам дистрибутив упаковывается в архив, где помимо инсталлятора может содержаться краткое описание в файле read.txt и название и версия программы в файле file_id.diz. Последний файл часто автоматически извлекается из архива каталогизаторами, файловыми менеджерами и архиваторами. Наличие этого файла является хорошим тоном сборки дистрибутива. Windows-программы упаковываются в zip-архивы, являющиеся стандартом де-факто для этой операционной системы. В UNIX-подобной операционной системе дистрибутив программы упаковывается либо в так называемый тарбол, файл с расширением tar.gz, либо в RPM-пакет. Тарбол — это объединенные при помощи утилиты tar в один файл множества файлов, который подвергается сжатию при помощи утилиты gzip. Распаковка происходит в обратном порядке, сначала происходит разархивирование, а затем разбиение полученного файла на отдельные составляющие. Система RPM позволяет устанавливать программы в операционных системах, которые поддерживают базу данных RPM-пакетов, введенную компанией RedHat. Данная система позволяет устанавливать новые программы, обновлять уже установленные программы и выводить список RPM-пакетов, которые присутствуют в системе.

Замечание

В Linux и других операционных системах под GNU-лицензией программное обеспечение чаще распространяется в виде исходных кодов, так как GNU-лицензия требует обязательного предоставления исходных кодов к распространяемым программам.

Хороший разработчик — хороший пользователь

Создавать хорошие программы можно только в том случае, если разработчик понимает, что необходимо пользователю. Если вы не играете в игры, вы никогда не сможете создать игру, если вы не используете форум — ваше творение никогда не будет удобным, если вы не пользуетесь редакторами — созданным вами редактором вряд ли кто-нибудь будет пользоваться.

Примечание

Обратное правило не верно, если вы интенсивно используете чужие программы, это вовсе не значит, что вам удастся создать совершенный продукт.

Из этого следует, что своим продуктом в обязательном порядке следует пользоваться, только тогда можно спроектировать удобный (хотя бы для вас) интерфейс, и существует ненулевая вероятность, что интерфейс будет удобен и другим пользователям. Когда использовать программу нет возможности, следует наблюдать за пользователями. Пару лет назад я разрабатывал программу для телефонной справочной службы. Для того чтобы отточить ее интерфейс, пришлось наблюдать на первых порах за работой операторов. Лишь только отследив движения рук, можно было понять, какой интерфейс требуется. Разрабатывая программу на рабочей станции с разрешением 1280×1024 (вместо 1024×800), владея десятипальцевым методом (в отличие от операторов), без гарнитуры на голове и необходимости отвечать на вопрос клиента трудно учесть все детали пользовательского интерфейса. Операторам оказалось удобнее выбирать информацию из древовидных списков, чем набирать в поисковом окне, так как когда в трубке звучит голос клиента, а взгляд направлен не на монитор, а на клавиатуру, совершить ошибку гораздо проще. Перемещение ползунка прокрутки, оказалось, занимает чрезвычайно много времени, гораздо проще привязать к клавише <F1> функцию разворота информационного окна на полный экран. Вопросы, связанные с вариантами проезда из одного пункта в другой, удобнее оказалось решать при помощи отрисовки карты области, с возможностью быстрого просмотра проходящих через населенный пункт маршрутов автобусов. Таким образом, наблюдение за пользователями позволило быстро устранить конструктивные недостатки, которые в лабораторных условиях казались незначительными.

Если, разработав форум, вы предпочитаете использовать стандартный phpBB, аргументируя свой выбор тем, что он менее дырявый и более функциональный, — не следует возлагать большие надежды на то, что им кто-нибудь заинтересуется. Программный продукт только тогда является живым, когда его интенсивно использует сам разработчик.

Разрабатывая форум для нужд IT-студии SoftTime, мы постоянно использовали форум в работе. Сначала в течение года для внутрикорпоративного обсуждения, оттачивая детали форума. Затем в течение года на сайте IT-студии SoftTime для поддержки читателей наших книг по PHP. Затем код форума был выложен для свободного доступа (<http://www.softtime.ru/info/liteforum.php>). Каждый из этапов позволял улучшить форум: внутрикорпоративное использование предъявляло высокое требование к "легковесности" форума, так как многие члены нашей команды выходили в Интернет через диалап-доступ. Использование форума в качестве форума-поддержки позволило внести множество удобных нововведений, которые предложили участники. Выкладывание исходного кода форума в свободный доступ выявило множество уязвимостей, которые устранялись так быстро, как только было возможно, так как на сайте постоянно находилась рабочая версия и откладывание работы по устранению дыр грозило полным уничтожением сайта. В настоящий момент IT-студия SoftTime имеет отточенный форум, код которого сотрудники студии знают как свои пять пальцев (а следовательно, могут легко модифицировать по желанию клиента), а с другой стороны форум не вызывает раздражения у пользователей, так как над его интерфейсом проделана большая работа и она не прекращается и по сей день.

Часто нет возможности длительно тестировать приложения в "боевых условиях". Однако опыт использования чужих программ невозможно переоценить. Следует интенсивно использовать чужие программы, подмечать в них удачные моменты и реализовывать в своих программах. Например, я регулярно покупаю новые игры, хотя редко удается задержаться в игре больше, чем на 20 минут. В первую очередь в играх интересен интерфейс. Игры являются передовыми программами, над которыми работают программисты и дизайнеры экстра-класса, в их разработку вкладываются огромные средства и в них можно почерпнуть множество идей в плане интерфейса. Например, звуковые возможности, обучающие режимы или расположение быстрых клавиш — это то, что будет стандартом в завтрашних программных продуктах.

Заявления действующего программиста, что он не играл в игры с 1997 года, а все программы пишет в Блокноте и призывает раздутые редакторы и интегрированные среды, возможно, были бы по достоинству оценены в Средние века, как подвиги отречения и самоистязания. Однако в XXI веке, когда технологии меняются каждые два года, человек с такой позицией и его программы просто будут отброшены за ненадобностью.

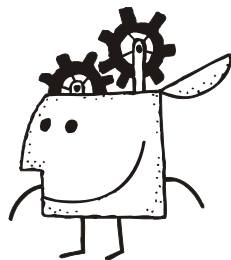
Необходимость исследования новинок касается не только программного обеспечения, но и различных "железных" нововведений. Даже если вы являетесь программистом экстра-класса, но ни разу не имели дело с USB, вам будет психологически трудно согласиться на разработку программного обеспечения, доступ к которому защищен ключом на Flash-карте. Точно также

вам будет тяжело согласиться добавить в программу функции автоматического резервного копирования на DVD-диск, если о DVD вы слышали только то, что он существует. Разработка простейшей программы распознавания образов может повергнуть вас в трепет, если вы не имели дело хотя бы с простейшей Web-камерой.

Это же касается регулярного просмотра новых книг, с которыми, слава богу, сейчас нет проблем. Однако посещение книжного магазина раз в год, приведет к тому, что в нужный момент необходимой книги может не оказаться. В результате чего вы вынуждены будете пользоваться английским мануалом и переводить его еще полгода (это в том случае, если английский мануал вообще существует в природе). Даже если вам неудобно посещать книжный магазин регулярно или вы живете в небольшом городе, привозят не все книги, но вам доступен Интернет, подпишитесь на рассылку крупного интернет-магазина, просто чтобы знать, какие книги имеются в продаже, чтобы в крайнем случае можно было достать нужную книгу, пусть даже дороже, а не тратить полгода на перевод английского мануала, обладая только троечными знаниями в немецком языке.

Интенсивное исследование нововведений предназначено для того, чтобы оставаться в струе программного обеспечения. Очень часто талантливые программисты уходят из этой области в связи с тем, что в какой-то момент они решают, что достигли вершины своей профессии и теперь могут почивать на лаврах. Такое в жизни бывает, но не в области программного обеспечения, пока здесь нет устоявшихся технологий, которые позволяют овладевшим ими не беспокоиться дальше о своем росте. Этому мешает уникальная ситуация, сложившаяся в области микроэлектроники, — производительность компьютеров неуклонно растет, а их стоимость падает.

Глава 9



Правила хорошего тона в программировании

Увеличение мощности вычислительной техники значительно изменило взгляд на программирование. После того как процессоры стали быстрее в тысячи раз, модули памяти объемнее в тысячу раз, а жесткие диски больше в десятки тысяч раз, главной ценностью при разработке программ стало время разработчика программы. Все усилия современных методик программирования направлены на уменьшение времени разработки программы, ее отладки и сопровождения. Оптимизация программ по скорости и занимаемому объему отошла на второй план. Одним из способов уменьшения сложности программного обеспечения является или разбивка на части, каждая из которых является простой.

Замечание

Единственная среда, где оптимизация скорости и объема, занимаемого программой, имеет значение — это мобильные устройства, но их динамичное развитие скорее всего снимет эти ограничения в ближайшее время.

Замечание

Изумительной книгой по "правилам хорошего тона программирования" является "Совершенный код" Стива Макконела.

Объявления переменных и функций

Переменные могут объявляться явно и неявно. Явное объявление переменных требует их декларации (C++, Pascal), при неявном объявлении (Visual Basic, PHP) переменную можно начинать использовать в выражении без

предварительных объявлений — компилятор сам позаботится об объявлении переменной. Однако неявное объявление переменных очень часто приводит к ошибкам отсутствия инициализации переменных. Разработчик часто забывает, что переменной следует присвоить значение и переменной присваивается значение 0 (в худшем случае вообще случайное значение). В больших проектах режим неявного объявления переменных отключается, а в спецификациях проекта требуют явного определения переменной и инициации ее начальным значением.

Замечание

Подавляющее большинство взломов Web-приложений, разработанных с использованием языка программирования PHP, основано на том, что переменные не получают начального значения.

Сам факт инициализации еще не страхует от совершения ошибок, так как часто на поиск места инициализации может уходить много времени.

Листинг 9.1. Пример неудачной инициализации переменной (C++)

```
// Объявление переменной
int CountContract;

// 200 строчек кода
...

// Инициализация переменной
CountContract = 10;

// Еще 200 строчек кода
...

// Использование переменной
for(int i = 0; i < CountContract; i++)
{
    ...
}

...
```

При таком подходе необходимо затратить значительные усилия на поиск строки инициализации переменной `CountContract`. С точки зрения читабельности программы гораздо более удачным решением является объявление и инициализация переменной либо в начале кода, либо перед непосредственным использованием.

Листинг 9.2. Объявление и инициализация переменной в начале кода (C++)

```
// Объявление переменной
int CountContract;
// Инициализация переменной
CountContract = 10;
// 400 строчек кода
...
// Использование переменной
for(int i = 0; i < CountContract; i++)
{
    ...
}
...
```

Листинг 9.3. Объявление и инициализация переменной перед использованием (C++)

```
...
// Объявление переменной
int CountContract;
// Инициализация переменной
CountContract = 10;
for(int i = 0; i < CountContract; i++)
{
    ...
}
...
```

Причем код в листинге 9.3 является более предпочтительным, чем код в листинге 9.2. Это связано с тем, что пока выполняются 400 строчек кода, разделяющих объявление и инициализацию переменной `CountContract` и место непосредственного использования переменной, значение `CountContract` может быть случайно переопределено.

Замечание

Некоторые языки программирования, например, Pascal, не позволяют объявлять переменные в произвольном месте программы. Если значение переменной не должно изменяться в ходе программы, ее следует объявить как константу — это предотвратит ее случайное изменение.

Еще одно правило, которым часто пренебрегают, это использование переменной с единственной целью — память в настоящий момент стоит гораздо дешевле, чем время разработчика, потраченное на разбор кода. Когда переменные используются для множества разнообразных задач, невозможно выдумать для них более или менее информативное название, переменная получает значение `temp`, `tmp` или что-то в этом духе. В результате общая читабельность программы падает. При повторном использовании массивов (особенно ассоциативных) возникает другая проблема — они часто оказываются засоренными предыдущими значениями. Разработчик чаще имеет дело с переменными, которые необязательно очищать — повторная инициализация уничтожит предыдущее значение — не отдавая себе отчет, разработчик при повторном использовании массива также часто забывает произвести его очистку, в результате чего появляются трудноулавливаемые ошибки. Использование переменной с одной целью означает не только отказ от повторного использования переменной в других задачах, но и запрет на использование переменной для нескольких целей. Например, часто положительное значение переменной может означать число товарных позиций, а отрицательное — код ошибки, или еще лучше — положительное — число байт в файле, а отрицательное — номер диска. Как только видно двойное использование переменной, необходимо тут же вводить вторую переменную, иначе ошибок будет трудно избежать, а читабельность программы будет снижена многократно. Даже если вам кажется очевидным правило использования переменных с двойным смыслом, такое использование переменной может оказаться непонятным для партнера по проекту или потребует дополнительного документирования, а затем и времени всех участников проекта на прочтение дополнительного объема документации.

Еще одним признаком небрежного кода является наличие так называемых мертвых переменных и функций — это конструкции, которые были разработаны, а затем исключены из использования в результате оптимизации кода. Программисты с большим воодушевлением пишут новый код и крайне неохотно стирают уже существующий. А вдруг пригодится!? Возможно, уничтожение старого кода сразу после внесения изменений является не совсем разумной идеей, но время от времени необходимо устраивать субботники по удалению мертвых функций и переменных, так как спустя некоторое время даже разработчику потребуются значительное время на то, чтобы определить, используется ли данная переменная или функция где-либо или нет. В результате объем кода будет возрастать и мешать в отладке и сопровождении программы.

Имена переменных и функций

Имена переменных и функций имеют огромное значение, так как чтение программы осуществляется не только машиной (которой, в общем, все рав-

но), но и человеком, который с именем связывает свое представление о предмете. Грубо говоря, если переменная называется "Кошка", то программист при работе с этой функцией невольно будет думать о "кошке", даже если эта функция обнуляет стек данных.

Замечание

Благодаря тому, что языки программирования создаются на английском языке, который не является родным для российских разработчиков, последние избавлены от целого класса ошибок, связанных с неоправданным ожиданием от конструкций языка действий, характерных для разговорного языка. Многие начинающие англоязычные разработчики ожидают, что цикл `while` прекратит работу сразу же, как будет выполнено условие цикла, в какой бы точке цикла не находилась программа.

Существует несколько стилей названия переменных:

- ☐ `var_bell` — стиль C: нижний регистр, знак подчеркивания;
- ☐ `VarBell` — стиль Pascal: каждая подстрока в названии начинается с прописной буквы;
- ☐ `varBell` — стиль Java: первая подстрока начинается со строчной (маленькой) буквы, все последующие — с прописной.

Не имеет значения, какой стиль будет выбран — главное, придерживаться в коде одного стиля. В больших проектах стиль именования переменных закрепляется в спецификациях, которые разрабатываются перед началом кодирования.

Совет

В программировании константы традиционно записываются в верхнем регистре `YANDEX_BOT`. Если вы хотите, чтобы другие программисты могли легко воспринимать ваш код, придерживайтесь этого правила.

Непродуманная система имен переменных и функций проекта является одной из серьезных ошибок программистов, причем не только начинающих. Называйте переменные и функции осмысленно — этим вы сэкономите много времени и себе, и вашему коллеге, который работает с вами в команде. От того, как вы назовете переменные и функции, зависит читабельность кода. Имя переменной (функции) должно отражать характер ее содержания (действия). Не используйте в названиях одному вам понятные сокращения — не пройдет и месяца, как вы сами забудете их смысл, не говоря уже о других людях, которые его не знали изначально. Крайне сложно отлаживать код, в котором переменные имеют имена `a1`, `dcr1_x15`, `nm`. Старайтесь, чтобы имена были не слишком короткими, но и не слишком длинными. Названия функций типа `MyNewFunctionForReadInformationFromForm` тоже не

способствуют улучшению читабельности кода. Ряд западных исследований показал, что отладка программы требует меньше всего усилий, если имена переменных состоят не больше чем из двух слов и 10—16 символов. Однако не всегда такое длинное имя является удачным, традиционно в качестве счетчиков циклов и индексов массивов используются односимвольные имена переменных (*i*, *j*, *k*, *n*), например, `employee[i]`.

Замечание

Данная традиция уходит корнями в FORTRAN, один из первых языков программирования. Если тип переменной не указывался явно, то переменные, чьи названия начинались с символов *I*, *J*, *K*, *L*, *M* и *N*, автоматически получали целочисленный тип `INTEGER`, те же переменные, которые начинались с других букв алфавита, получали по умолчанию вещественный тип `REAL`. Поэтому в циклах можно было использовать переменные *I*, *J*, *K*, *L*, *M* и *N*, не объявляя их специально.

Старайтесь эти слова отделять друг от друга символом подчеркивания или выделять заглавными буквами. Согласитесь, когда переменная имеет имя `user_address`, это удобнее чем просто `useraddress`. По возможности избегайте употребления числительных в именах. Сложно работать с кодом, в котором переменные имеют имена `str1`, `str2`, `str3` и т. д. Появление в коде переменных с такими именами говорит о необходимости введения массива.

Помните, что необдуманная система имен в том случае, если вы программируете один, добавит вам несколько десятков часов бессмысленной работы, а если вы работаете в команде, небрежное обращение с именами — просто элементарное неуважение к вашим коллегам и их труду.

Рассмотрим несколько примеров. Часто временные строки для хранения SQL называют `query` (запрос) — это очень удачное название, ассоциирующееся именно с SQL-запросом. Обычно на этом все и заканчивается. При появлении второго запроса вторая переменная получает имя `query1` — это порочная практика. Обычно запросы в рамках одного скрипта отличаются своим действием: один SQL-запрос может выполнять обновление (`UPDATE`), другой выборку (`SELECT`), поэтому переменные лучше называть с указанием действия оператора: `query_update` и `query_select` соответственно.

Листинг 9.4. Ошибочное именование переменных (PHP)

```
<?php
$query = "SELECT * FROM catalog";
$query1 = mysql_query($query);
while($result = mysql_fetch_array($query1))
{
```

```
/* Код обработки */  
}  
?>
```

Здесь дескриптор запроса, возвращаемый функцией `mysql_query()`, назван `query1`, это здорово запутывает как разработчика, так и читающего текст программы. Дескриптор не несет физического смысла — это лишь ключ к результату, поэтому его лучше называть сокращенным именем таблицы (листинг 9.5).

Листинг 9.5. Правильное именование переменных (PHP)

```
<?php  
$query = "SELECT * FROM catalog";  
$cat = mysql_query($query);  
while($catalog = mysql_fetch_array($cat))  
{  
    /* Код обработки */  
}  
?>
```

При именовании переменных булевых следует избегать ситуаций, когда значение переменной не будет сообщать разработчику ровным счетом ничего. Например, статусные переменные, которые принимают `true`, например, когда память выделена, и `false` в противном случае, часто называют ничего не значащими именами `flag` и `status`. Встретив в программе имя `flag` или `status`, невозможно определить при каком состоянии флага память выделена, а при каком сброшена. Поэтому такого рода переменные часто снабжаются префиксами `done` (сделано), `error` (ошибка), `ok` (Ok), `success` (успех), которые позволяют однозначно определить состояние программы. Часто используются стандартные префиксы для именования переменных. Самая известная из нотаций — венгерская нотация, которая широко применялась для программирования в Windows в самом начале развития этой операционной системы, когда программы создавались без современных библиотек и сред с использованием чистого API.

Замечание

Данная система является широко используемым внутренним стандартом корпорации Microsoft. Название "венгерская нотация" появилось из-за того, что ее разработал доктор Симонии родом из Венгрии.

Венгерская нотация использует стандартные префиксы перед именами переменных, это позволяет легко определять тип и назначение переменной в коде, не обращаясь к объявлению переменной. В табл. 9.1 приводится список префиксов и демонстрируется применение их в названиях переменных.

Таблица 9.1. Венгерская нотация

Префикс	Пример	Значение
b	bPoint	Размер переменной равен одному байту
c	cContract	Количество (записей, символов, объектов и т. д.)
cb	cbDocument	Размер в байтах
ch	chCursorPosition	Символ, обычно в ASCII
cw	cwDocument	Размер в словах
d	dPoint	Разница между двумя переменными
dw	dwPoint	Двойное слово (4 байта в 32-битной архитектуре)
dbl	dblPrice	Число с плавающей точкой двойной точности
f	fDoneConversion	Флаг (flag) для указания булева состояния
flt	fltPrice	Число с плавающей точкой одинарной точности
h	hWindow	Дескриптор (HANDLE)
i	iKey	Индекс массива
p	pDocument	Указатель
pv	pvAccess	Указатель на тип void (void *)
ppv	ppvAccess	Указатель на указатель (void **)
rg	rgContract	Массив (записей, объектов и т. д.)
sz	szTitle	Строка, которая заканчивается нулевым символом
w	wAddress	Машинное слово (2 байта в 32-битной архитектуре)
wn	wnMain	Окно

Вовсе не обязательно придерживаться стандартной нотации, можно разработать собственную, главное, чтобы все участники проекта знали о ней и использовали в работе.

Если в качестве префиксов имен применяются антонимы, следует использовать их последовательно. Так, если вы работаете с последовательностью сообщений, отсортированных по дате таким образом, что новые (new) сооб-

щения располагаются в начале (first), а самые старые (old) сообщения в конце (last). Не стоит для обозначения первого сообщения применять переменную `firstMessage`, а для последнего — `oldMessage`. Антонимы следует использовать в паре, т. е. называть эти две переменные либо `firstMessage` и `lastMessage`, либо `newMessage` и `oldMessage`. В табл. 9.2 приводится список часто используемых антонимов и их русские переводы.

Таблица 9.2. Список часто используемых антонимов

Антонимы	Русское значение
add/delete	добавить/удалить
begin/end	начало/конец
first/last	первый/последний
left/right	левый/правый
locked/unlocked	закрыто/открыто
min/max	минимальный/максимальный
next/previous	следующий/предыдущий
old/new	старый/новый
opened/closed	открытый/закрытый
visible/invisible	видимый/невидимый
show/hide	показать/скрыть
up/down	вверх/вниз

Применение общеизвестных имен снижает объем информации, которую необходимо удерживать разработчику в голове.

Одно из распространенных явлений — это создание двух переменных, незначительно отличающихся в написании. Например, `date_day` и `date_date` или `contract` и `contracts`. Когда программист работает с двумя такими переменными — ошибиться ничего не стоит. Еще одним бичом в проекте может быть использование переменной с орфографической ошибкой — такая переменная будет являться источником постоянных ошибок в коде. В связи с этим следует избегать использования слов, в которых вы допускаете ошибки, если вы не помните, сколько и каких букв должно быть в словах `access` или `referrer`, лучше не применяйте их, а найдите им эквиваленты.

Расстановка фигурных скобок и отступы

Неструктурированный код часто является источником ошибок начинающих программистов, так как они не знают, что код может быть упорядочен, после чего его отладка многократно облегчается.

Листинг 9.6. Пример неструктурированного кода (PHP)

```
<?php
$msg = ""; // сообщение
$error = false; // проверяем на ошибки
$file_nav = "../navigation_one.txt"; // файл
$file_fopen_nav = fopen($file_nav,"r"); // читаем его
$buff = fread($file_fopen_nav, 1000); // заносим в переменную
$exp_nav = explode("|", $buff); // затем в массив
$count_nav = count($exp_nav); // определяем кол-во элементов
for ($i = 0; $i < $count_nav; $i++) { // делаем, пока все не перечислим
if ($exp_nav[$i] == $name) {
// если раздел и введенная информация совпадает то:
$s++; // прибавляем к счетчику 1
$i1 = $i; // заносим первый элемент в переменную $i1
$i2 = $i1+1; // заносим следующий элемент в переменную $i1
}
$i++; // перепрыгиваем через один, так как потом идет ссылка
}
if ($s > 0) { // проверяем кол-во совпадений
$msg = "Раздел удачно удален!";
$error = false; // устанавливаем, что ошибок нет
} else {
$msg = "Данного раздела не существует!";
$error = true; // устанавливаем, что есть ошибки
}
if (!$error) { // если нет ошибок
$part_str = $exp_nav[$i1]."|"; // заносим раздел в переменную
// если вторая переменная $i2 равна кол-ву элементов, то значит,
// она последняя, соответственно тогда не надо вписывать в переменную
// знак "|"
if ($i2 == $count_nav) {
$url_str = $exp_nav[$i2];
```

```
} else {  
$url_str = $exp_nav[$i2]."|";  
}  
// заменяем  
$buff = str_replace($part_str, "", $buff);  
$buff = str_replace($url_str, "", $buff);  
// записываем  
$file_new = fopen($file_nav, "w");  
fputs($file_new, $buff);  
fclose($file_new); }  
>>
```

Код, приведенный в листинге 9.6, снабжен большим числом комментариев, но понять его логику трудно, так как составные операторы сливаются друг с другом и сложно различить, какой оператор зависит от другого оператора. Этот же код с правильно расставленными отступами гораздо проще анализировать (листинг 9.7).

Замечание

Когда мне необходимо проанализировать подобный код, я просто расставляю все отступы — сколько бы времени это не занимало, отладка неструктурированного кода все равно будет протекать дольше. Исследования показали, что программисты со стажем и начинающие отлаживают неструктурированный код с одинаковой скоростью, в то время как скорость отладки структурированного кода отличается в разы.

Листинг 9.7. Пример структурированного кода (PHP)

```
<?php  
$msg = ""; // сообщение  
$error = false; // проверяем на ошибки  
  
$file_nav = "../navigation_one.txt"; // файл  
$file_fopen_nav = fopen($file_nav, "r"); // читаем его  
$buff = fread($file_fopen_nav, 1000); // заносим в переменную  
$exp_nav = explode("|", $buff); // затем в массив  
$count_nav = count($exp_nav); // определяем кол-во элементов  
  
for ($i = 0; $i < $count_nav; $i++) { // делаем, пока все не перечислим  
    if ($exp_nav[$i] == $name) {
```

```
// если раздел и введенная информация совпадает то:
$s++; // прибавляем к счетчику 1
$i1 = $i; // заносим первый элемент в переменную $i1
$i2 = $i1+1; // заносим следующий элемент в переменную $i1
}
$i++; // перепрыгиваем через один, так как потом идет ссылка
}

if ($s > 0) { // проверяем кол-во совпадений
    $msg = "Раздел удачно удален!";
    $error = false; // устанавливаем, что ошибок нет
} else {
    $msg = "Данного раздела не существует!";
    $error = true; // устанавливаем, что есть ошибки
}

if (!$error) { // если нет ошибок
    $part_str = $exp_nav[$i1]."|"; // заносим раздел в переменную
    // если вторая переменная $i2 равна кол-ву элементов, то значит,
    // она последняя, соответственно тогда не надо вписывать в переменную
    // знак "|"
    if ($i2 == $count_nav) {
        $url_str = $exp_nav[$i2];
    } else {
        $url_str = $exp_nav[$i2]."|";
    }

    // заменяем
    $buff = str_replace($part_str, "", $buff);
    $buff = str_replace($url_str, "", $buff);
    // записываем
    $file_new = fopen($file_nav, "w");
    fputs($file_new, $buff);
    fclose($file_new);
}
?>
```

Существует несколько стилей расстановки фигурных скобок. Все они диктуются существующими стилями в других С-подобных языках программирования.

❑ *Рациональный стиль.*

Это один из наиболее распространенных стилей, т. к. им пользовались Керниген (Kernighan) и Ричи (Ritchie) — авторы языка С (в листинге 9.8).

Листинг 9.8. Рациональный стиль программирования

```
if(isPrint){  
    print("Hello, World!");  
}
```

Преимущество этого подхода заключается в экономии вертикального пространства, жизненно важного при отладке большого блока кода. Обратной стороной такого подхода является то, что может оказаться трудным найти символ {, спрятанный в конце строки. Этому стилю придерживаются и Java-программисты, как то предписывает корпорация Sun.

❑ *Стиль Алмена.*

Эрик Алмен (Eric Allman) написал утилиты BSD в этом стиле, поэтому данный стиль часто называют "стилем BSD" (листинг 9.9).

Листинг 9.9. Стиль Алмена

```
if(isPrint)  
{  
    print("Hello, World!");  
}
```

Аргументом в поддержку такого стиля является тот факт, что область видимости блочного оператора ясна и визуально ассоциируется с управляющим оператором.

❑ *Стиль Whitesmith.*

Данный стиль предписывает использование такой расстановки фигурных скобок, которая представлена в листинге 9.10.

Листинг 9.10. Стиль Whitesmith

```
if(isPrint)
{
    print("Hello, World!");
}
```

Этот стиль имеет преимущество в том, что скобки более тесно ассоциируются с кодом, который они включают и разграничивают, однако при визуальном просмотре текста отыскать скобки оказывается чуть более сложно.

Замечание

Данный стиль получил свое название по имени одного из первых компиляторов С. Долгое время документация к этому компилятору была единственным справочным пособием по С. Примеры кода были выполнены в стиле, приведенном в листинге 9.8.

❑ *Стиль GNU.*

Программисты GNU фонда Free Software Foundation используют следующий стиль расстановки фигурных скобок (листинг 9.11).

Листинг 9.11. Стиль GNU

```
if(isPrint)
{
    print("Hello world!");
}
```

Внутри любых управляющих конструкций операторы следует располагать с отступом на одинаковое число пробелов, например, для операторов `if-then-else` код должен выглядеть так, как показано в листинге 9.12.

Листинг 9.12. Стиль GNU для операторов if-then-else

```
isPrint = true;
if(isPrint)
{
    print("Переменная равна true");
}
```

```
else
{
    print("Переменная равна false");
}
```

Количество пробелов может быть любым, обычно используют 2, 4 или 8 пробелов. Старайтесь придерживаться этого правила некоторые программисты приходят в бешенство, когда это число не кратно 2. Наиболее оптимальным является использование 2 пробелов, т. к. при их большем числе, вложенные блоки становятся "растянутыми" и их сложно воспринимать.

Хорошей практикой является при организации составного блока сразу писать открывающую и закрывающую скобки {} — это позволяет избежать ошибок, связанных с тем, что, пока создается код составного блока, про закрывающую скобку можно забыть (листинг 9.13).

Листинг 9.13. Запись открывающей и закрывающей скобок одновременно (C++)

```
// Сначала создается код
for(i = 0; i < maxContracts; i++)
// Затем одновременно создаются открывающая и закрывающая скобки
for(i = 0; i < maxContracts; i++) {}
// После чего внутри их размещается код
for(i = 0; i < maxContracts; i++)
{
    // Код ...
}
```

Если в теле цикла много кода и вложенных операторов, хорошей практикой является после завершения цикла дублировать заголовок цикла, к которому относится закрывающая фигурная скобка (листинг 9.14).

Листинг 9.14. Запись открывающей и закрывающей скобок одновременно (C++)

```
// После чего внутри их размещается код
for(i = 0; i < maxContracts; i++)
{
    // Несколько сотен строк
    ...
} // for(i = 0; i < maxContracts; i++)
```


Правда, заголовок цикла, пусть даже в комментарии, может снижать читабельность кода, поэтому в комментарии может быть просто текстовый комментарий.

Одним из факторов, снижающих читабельность кода, является наличие большого числа вложенных операторов. В исследованиях утверждается, что немногие люди способны понять более трех вложенных операторов `if`. Рекомендуется не использовать более трех-четырёх уровней вложения. В листинге 9.15 приводится пример глубоко вложенного кода.

Листинг 9.15. Пример глубоко вложенного кода (C++)

```
if(isSetContractCondition)
{
    // 200 строк кода
    if(printContract != NULL)
    {
        // 200 строк кода
        if(SetupPage())
        {
            // 200 строк кода
            if(executePage)
            {
                // 200 строк кода
            }
        }
    }
}
```

Уменьшить число вложенных уровней в два раза можно за счет дублирования внешних условий для двух самых глубоко вложенных блоков `if` (листинг 9.16).

Листинг 9.16. Снижения уровня вложенности (C++)

```
if(isSetContractCondition)
{
    // 200 строк кода
    if(printContract != NULL)
    {
        // 200 строк кода
    }
}
```

```
    }  
}  
if(isSetContractCondition && printContract != NULL)  
{  
    if(SetupPage())  
    {  
        // 200 строк кода  
        if(executePage)  
        {  
            // 200 строк кода  
        }  
    }  
}
```

Такой подход, когда глубоко вложенные условия выносятся за пределы внешнего блока `if`, имеет недостаток, связанный с тем, что условия дублируются в нескольких местах программы и при последующем изменении программы потребуются перелопатить весь код, чтобы обнаружить все условия, подвергающиеся изменениям. Однако этот подход позволяет значительно увеличить читабельность кода.

Еще одним приемом уменьшения глубины вложенности является использование оператора `do` с постусловием. В операторе `do`, как и в любом цикле, можно использовать оператор `break`, чтобы немедленно покинуть составного оператора `{}`.

Листинг 9.17. Использование оператора `do` с постусловием (C++)

```
do  
{  
    if(!isSetContractCondition)  
    {  
        break; // Выходим из блока  
    }  
    // 200 строк кода  
  
    if(printContract == NULL)  
    {  
        break; // Выходим из блока  
    }  
}
```

```
// 200 строк кода

if(!SetupPage())
{
    break; // Выходим из блока
}

// 200 строк кода

if(!executePage)
{
    break; // Выходим из блока
}

// 200 строк кода
} while(false);
```

Как видно из листинга 9.17, все условия инвертированы — если условие каждого из блоков станет истинным — сработает оператор `break`, который заставит программу выйти из цикла и перейти на строку `while(false)`. Так как постусловие всегда ложно, то тело цикла `do` будет выполняться всегда один раз.

Замечание

В листинге 9.17 оператор `do` используется не по назначению, поэтому такая методика применима только в том случае, если разработчики, которые будут сопровождать ваш код, знакомы с таким приемом.

Если код выполняется в теле функции, то вместо оператора `break` лучше использовать досрочный выход из функции при помощи оператора `return`.

Листинг 9.18. Использование оператора `return` (C++)

```
void printContractCondition()
{
    if(!isSetContractCondition)
    {
        return; // Выходим из функции
    }

    // 200 строк кода
```

```
if(printContract == NULL)
{
    return; // Выходим из функции
}
// 200 строк кода

if(!SetupPage())
{
    return; // Выходим из функции
}
// 200 строк кода

if(!executePage)
{
    return; // Выходим из функции
}
// 200 строк кода
}
```

В ряде сред программирования, например, в PHP или Perl, можно также осуществить досрочный выход из основной программы при помощи функций `exit()` или `die()`.

Пробелы вокруг символов

Бинарные операторы следует обрамлять пробелами (листинг 9.19). Использование пробелов позволяет отделить операнды от операторов.

Листинг 9.19. Обрамление бинарных операторов пробелами (PHP)

```
<?php
// Неправильно
$a=$b+$c*$d;
// Правильно
$a = $b + $c * $d;
?>
```

Символ пробела ассоциируется с новым словом, поэтому формула читается не как непонятный набор символов, а как нечто осмысленное.

Комментарии

Часто код является самодокументируемым, т. е. не требует специального пояснения; так, в листинге 9.20 представлен код, выполняющий запрос к базе данных.

Листинг 9.20. Код, выполняющий запрос к базе данных (PHP)

```
<?php
$query = "SELECT name FROM tbl WHERE id = 1";
$dcr = mysql_query($query);
$arr = mysql_fetch_array($dcr);
echo $arr;
?>
```

Вряд ли порядок вызова функций и размещения строк кода будет перепутан, так как результат, возвращаемый каждым из операторов, является входным параметром для следующего оператора. В то же время порядок следования функций в листинге 9.21 легко перепутать, так как зависимость вызовов функций не очевидна из названий.

Листинг 9.21. Соккрытие логики программы в функциях

```
<?php
execute("SELECT name FROM tbl WHERE id = 1");
convert();
sendbrowser();
?>
```

Такие блоки необходимо снабжать комментариями — пройдет некоторое время, и вы забудете, что делал тот или иной программный блок. Вообще, принято комментировать код на английском языке или не комментировать вообще, т. к. для русского языка существует масса кодировок, да и вообще так исторически сложилось. Не обращайтесь на это внимания — код вы комментируете в первую очередь для себя, а не для других! А раз уж вы делаете это для себя, делайте это в удобной для вас кодировке.

Существует много видов комментариев, основные из которых представлены в табл. 9.3.

Таблица 9.3. Различные виды комментариев

Язык программирования	Вид комментария
C	/* */
C++	// /* */
Perl, PHP	# // /* */
Visual Basic	'
Fortran	!

Комментарии могут быть как однострочными (листинги 9.22 и 9.23), так и многострочными (листинг 9.24).

Листинг 9.22. Однострочный комментарий в стиле shell-скриптов

```
<?php
# Программный модуль index.php
echo "Hello world!";
?>
```

Листинг 9.23. Однострочный комментарий в стиле C++

```
<?php
// Программный модуль index.php
echo "Hello world!";
?>
```

Листинг 9.24. Многострочный комментарий в стиле C

```
<?php
/* Это многострочный комментарий в стиле C.
Он охватывает несколько строк – не допускается
вложенных комментариев
*/
```

```
echo "Hello world!";  
?>
```

К хорошему стилю программирования относится использование однострочных комментариев для короткого комментария, а многострочного — для комментария, охватывающего несколько строк. Не возбраняется использовать однострочные комментарии для большого текста, особенно в начале файла или важного блока кода (листинг 9.25).

Листинг 9.25. Использование однострочного комментария для большого текста

```
<?php  
////////////////////////////////////  
// Гостевая книга  
////////////////////////////////////  
?>
```

Как и при работе с отступами и фигурными скобками, основным требованием является необходимость придерживаться одного стиля во всех программных блоках.

При расстановке однострочных комментариев возможны два варианта: непосредственно перед выполняемым оператором (листинг 9.26) и после точки с запятой (листинг 9.27).

Листинг 9.26. Однострочный комментарий *перед* оператором

```
<?php  
// Вывод текстовой строки в окно браузера  
echo "Hello world!";  
?>
```

Листинг 9.27. Однострочный комментарий *после* точки с запятой

```
<?php  
echo "Hello world!"; // Вывод текстовой строки в окно браузера  
?>
```

Лучше придерживаться первого правила, т. к. во втором случае строка получается длинной и плохо воспринимается читающим. Единственным оправ-

данием использования такого комментария является комментирование закрывающейся скобки длинного программного блока, содержащего много вложенных блоков (листинг 9.28).

Листинг 9.28. Комментирование закрывающейся скобки длинного программного блока

```
<?php
    if($tot)
    {
        while($position = next($tot))
        {
            /* Очень длинный код,
               содержащий много
               вложенных блоков
               ...
            */
            if($flag)
            {
                echo "Ошибка";
                exit();
            }
        } // Конец while($position = next($tot))
    }
?>
```

Объектно-ориентированное программирование

Одним из способов снижения сложности программного обеспечения является применение объектно-ориентированного подхода, в котором программа разбивается на иерархию классов. Существуют базовые классы, от которых наследуются производные классы. Наследники могут использовать часть методов базовых классов, а базовые классы могут задавать поведение производных классов. Классы позволяют создавать объекты — структурные образования, которые включают и данные, и методы, обрабатывающие эти данные. Удобство их применения заключается в том, что задача переносится на более высокий абстрактный уровень. Программист вместо того, чтобы оперировать низкоуровневым представлением объекта, может работать в программе непосредственно с объектами, например, с договором или товарной

позицией, а не с набором из 120 разнообразных параметров. Кроме того, разработчик имеет дело лишь с интерфейсом, не вникая в сложности реализации класса.

Когда используется объектно-ориентированное программирование

Можно создавать объектно-ориентированные программы на языках, не поддерживающих элементы объектно-ориентированного программирования, именно так создавался Windows 95 на языке С, в котором отсутствует специальная поддержка объектов. В то же время с использованием языка с поддержкой объектно-ориентированного программирования можно создать нечитабельное месиво, ничего общего не имеющее с объектно-ориентированным программированием.

Замечание

Эрик С. Реймонд в книге "Искусство программирования для UNIX" следующим образом характеризует объектно-ориентированные языки: "Объектно-ориентированные языки поддерживают создание структур с большим количеством связующего кода и сложными уровнями. Это может оказаться полезным в случае, если предметная область является действительно сложной и требует множества абстракций, и вместе с тем такой подход может обернуться неприятностями, если программисты реализуют простые вещи сложными способами, просто потому, что им известны эти способы и они умеют ими пользоваться".

Другими словами, существует слишком большой соблазн использовать классы там, где не следует использовать наследование, когда в реальности иерархия объектов отсутствует, использовать полиморфизм, вообще не понимая, что это такое и в каких случаях его применение оправдано.

Часто решение использовать объектно-ориентированную технологию принимается в надежде, что это поможет компенсировать недостатки проектирования и понимания способов реализации поставленных задач.

Сделать код непрозрачным и нечитабельным при использовании объектно-ориентированного программирования так же легко, как и в его отсутствие, и даже еще проще. Сложную проблему можно разбить на несколько уровней, и это поможет решить проблему уровень за уровнем. Можно разбить и простую задачу на несколько уровней, но в результате вместо простой задачи у вас будет сложная задача (рис. 9.1).

Объектно-ориентированный подход — прекрасное инженерное решение, но принимать решение о его применении следует только в том случае, если вы твердо и аргументированно можете назвать выгоды его использования по сравнению с альтернативными методами разработки программного обеспечения.

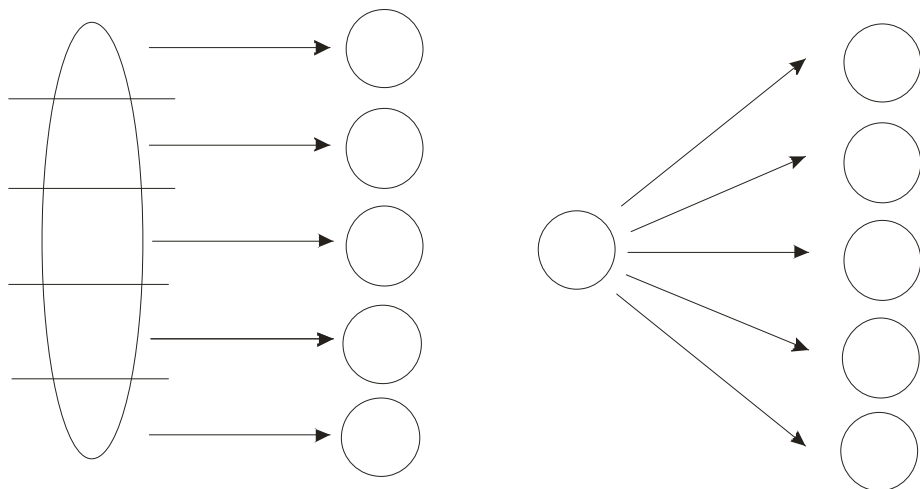


Рис. 9.1. Применение объектно-ориентированного подхода для сложной и простой задач

Типичной задачей использования объектно-ориентированного подхода является построение графической библиотеки или графического редактора, которые оперируют большим числом разнообразных объектов, между которыми имеется много общего. Например, в качестве базового класса можно выделить "графический примитив", представление которого можно сохранять на жесткий диск, выводить на экран монитора, перемещать по экрану при помощи мыши. Разумеется, сам по себе графический примитив отобразить нельзя — это абстрактное понятие и объектов данный класс в программе иметь не будет. Объекты будут только у классов-наследников, например, у класса "точки", который уже можно отобразить на экране компьютера и перемещать. Помимо абстрактного класса "графический примитив", точка будет обладать свойством цвета и координат на экране. Следующим по иерархии классом будет "графическая фигура", от которой будут унаследованы классы "круга", "прямоугольника", "линии" и т. п. От класса "прямоугольник" можно унаследовать класс "окно", от которого наследуется класс "кнопка", "текстовая область", "диалоговое окно" и т. д. Таким образом, выстраивается четкая иерархия сложного приложения, код, написанный один раз в базовом классе, уже никогда не повторяется, так как он автоматически включается в классы-наследники.

Однако существуют области, где применение объектно-ориентированного подхода не приемлемо. Например, это относится к задачам программирования для Интернета. Следует отметить, что Perl-разработчики практически не используют объектно-ориентированный подход. PHP-разработчики, если его и используют, то полученный код, как правило, не пользуется популяр-

ностью. Связано это в первую очередь с тем, что Интернет, Perl и PHP развивались в UNIX-среде, где предпочтение отдавалось небольшим программам с развитой интерфейсной частью. Для решения сложных задач использовалась комбинация таких программ, оформленная в виде shell-сценария. За годы существования UNIX и Интернета интерфейсы и философия создания программного обеспечения были отточены до мелочей. Вся сложность, какую только можно было убрать, была убрана. Использование классов может только привести ее.

Пусть стоит задача создания Web-форума. Применение объектно-ориентированной технологии здесь просто не оправдано, так как, с одной стороны, имеется база данных, которая как правило является реляционной и требует просто тупого копирования каждого элемента объекта в поля таблицы. С другой стороны, имеется протокол HTTP, с использованием которого построен Web. Данный протокол также не предназначен для работы с классами и требует линейного представления данных, желательно в текстовом, а не бинарном виде. Таким образом, вся объектная структура разбивается о глыбы базы данных и HTTP-протокола. Даже если трудности несовместимости SQL и HTTP с объектно-ориентированным подходом будут преодолены, например, за счет сериализации, т. е. сохранения структуры объекта в текстовую строку, следует признать, что в Web-форуме нет иерархии, и мы можем получить только беспорядочный набор классов, код в которых и без их использования нигде бы не дублировался. Какими же могут быть эти классы? Например, класс помещения в базу данных (листинг 9.29).

Листинг 9.29. Пример неудачного класса (PHP)

```
class put_database
{
    // Конструктор — устанавливает соединение с базой данных
    function __construct($dblocation = "localhost",
                        $dbuser = "root",
                        $dbpasswd = "")
    {
        $this->dblocation = $dblocation;
        $this->dbuser      = $dbuser;
        $this->dbpasswd    = $dbpasswd;
        $this->link = mysql_connect($dblocation, $dbuser, $dbpasswd);
        if (!$this->link) exit("К сожалению, не доступен сервер MySQL");
    }
    // Функция осуществляет выбор базы данных после установки соединения
    function select_db($dbname = "local")
```

```
{
    $this->dbname      = $dbname;
    if (!mysql_select_db($this->dbname, $this->link) )
    {
        exit("К сожалению, не доступна база данных");
    }
}

// Функция устанавливает SQL-запрос на выполнение
function set_sql($sql)
{
    $this->sql = $sql;
}

// Функция выполняет SQL-запрос
function execute()
{
    if(isset($this->sql)) return mysql_query($this->sql);
    else return null;
}

// Деструктор — закрывает соединение с базой данных
function __destruct()
{
    mysql_close($this->link);
}
}
```

Таким образом, для выполнения простейшего запроса необходимо выполнить следующий набор операторов (листинг 9.30).

Листинг 9.30. Выполнение SQL-запроса

```
$sql = new put_database();
$sql->select_db();
$sql->set_sql("SELECT VERSION()");
$sql->execute();
```

Без использования класса, данный код может выглядеть так как это представлено в листинге 9.31.

Листинг 9.31. Структурный подход

```
$dblocation = "localhost";  
$dbname = "local";  
$dbuser = "root";  
$dbpasswd = "";  
$link = mysql_connect($dblocation, $dbuser, $dbpasswd);  
if (!$link) exit("К сожалению, не доступен сервер MySQL");  
if (!mysql_select_db($dbname, $dbcnx)) exit "К сожалению, не доступна  
база данных";  
mysql_query("SELECT VERSION");
```

В отличие от листинга 9.30, программист, разбирая этот код, не встретит трудностей, так как все функции ему знакомы, и они имеют устоявшийся порядок вызова. Перед использованием класса он вместо того, чтобы сразу его использовать (именно для этого предназначены классы), будет вынужден изучать его, чтобы определить порядок вызова функций и запомнить его. Как видно, код, полученный в результате использования класса, получился более объемным (листинги 9.29 и 9.30) по сравнению с прямым вызовом функций API-интерфейса базы данных (листинг 9.31). Это связано с тем, что API-интерфейс создан максимально простым и удобным в работе. Дальнейшее его упрощение просто приводит к запутыванию и разбуханию кода.

Использование объектно-ориентированного подхода окупается как правило в больших задачах, где имеется большой соблазн дублировать код, а также необходима инкапсуляция сложных частей и построение иерархии. В небольших задачах, например, в Web-программировании (здесь, как правило, преобладают небольшие задачи) его использование не целесообразно.

Часто даже когда задача идеально подходит для использования объектно-ориентированного подхода, код в целом получается хуже, чем без его использования. Это связано с тем, что объектно-ориентированному подходу следует учиться... В первую очередь следует учиться выделять абстрактные понятия на высоком уровне, реализовывать класс "представления документа", класс "договора", класс "клиента", а не класс файла, в котором хранится договор, и класс C, основной задачей которого является преобразование объекта F в объект L. Даже после того, как объектно-ориентированная методология усвоена, считается, что это не приводит к ускорению первых пяти программных проектов. Это вовсе не значит, что объектно-ориентированное программирование бесполезно и не должно применяться; при правильном использовании данная методология позволяет многократно сократить издержки по созданию и сопровождению приложения.

Замечание

Преждевременная оптимизация (по скорости, использованию памяти и т. п.), по словам Д. Кнута, является "корнем всех зол". Многочисленные исследования показали, что программисты, как правило, не могут угадать "горячие" точки программы, так как внутренняя реализация современных компиляторов мало кому известна; кроме того, за последние годы она претерпела значительные изменения и практически все старые советы по оптимизации программы по скорости потеряли актуальность.

При создании объектно-ориентированного кода не следует гнаться за его скоростью и использовать в классах знания о внутренней реализации другого класса. Код будет либо быстрым, либо объектно-ориентированным. За лучшую структурированность и организацию кода посредством объектно-ориентированного подхода следует расплачиваться скоростью, и, наоборот, если увеличивается скорость кода — скорее всего где-то происходит нарушение инкапсуляции. Проверить классы на инкапсуляцию очень просто — возьмите любой класс своей программы и попробуйте применить его в другой программе. Если класс "договор" потребует использования класса "диалоговое окно" — код не является объектно-ориентированным, произошла разгерметизация класса (нарушение принципа инкапсуляции). Это означает только одно, что при проектировании было использовано знание о внутренней реализации другого класса — так получается быстрее, но код сложнее сопровождать и его практически невозможно применить повторно. Если ваш код именно такой — следует пересмотреть свое представление об объектно-ориентированном подходе.

Интерфейс класса

Если задача будет решаться с использованием объектно-ориентированного подхода, то первое, что необходимо выполнить при создании класса, — это разработка ясного интерфейса, который будет скрывать внутреннюю реализацию класса. Часто принцип сокрытия (инкапсуляции) нарушается, например, в листинге 9.29 необходимо помнить порядок вызовов методов, что требует от пользователя понимания внутреннего устройства класса — иногда легче вообще отказаться от использования классов, чем заставлять пользователя запоминать порядок вызова незнакомых ему методов.

Замечание

При разработке интерфейса зависимости от времени не всегда можно избежать, но по возможности к этому следует стремиться.

Интерфейс включает не только порядок вызова методов, но их название и расположение реализации. Рассмотрим класс договора (листинг 9.32).

Листинг 9.32. Класс договора (C++)

```
class Contract
{
    // Открытые члены и методы
public:
    // Конструкторы и деструкторы
    Contract();
    Contract(
        int Number,
        String name,
        String address,
        String phone,
        String condition
    );
    ~Contract();
    // Открытые члены класса
    String getName();
    String getAddress();
    String getPhone();
    String getCondition();
    String getSumm();
    String concludeTreaty();
    String getHTMLFormat();
    String getPlainFormat();
    ...
    // Закрытые члены и методы
private:
    ...
};
```

Как видно из листинга 9.32, класс начинается с открытых методов и членов, т. е. тех, с которыми пользователь класса будет иметь дело в первую очередь, затем следуют закрытые члены класса. Обратное расположение только запутает читающего и введет его в искушение разобраться в принципах работы класса, что является нарушением инкапсуляции (сокрытия внутренней реализации) и сводит преимущества использования класса на нет.

Разные языки программирования по-разному реализуют объектно-ориентированную модель и имеют отличающийся синтаксис конструкции класса.

Так, в C++ реализацию методов можно выносить за пределы класса, в других языках, таких как Java и PHP, реализация методов в обязательном порядке должны быть расположена в классе. Недостатком такого подхода является то, что пользователь класса не видит полного списка методов, поэтому рекомендуется в начале класса составлять комментарий с названием открытых членов и методов класса.

Замечание

На поддержание такого списка разработчику необходимо тратить дополнительные усилия, и возникает соблазн не поддерживать его в актуальном состоянии. В технологии Java данную проблему может помочь решить механизм `JavaDoc`, автоматически формирующий документацию на основании кода.

Методам класса следует давать однотипные названия, как правило, состоящие из глагола и существительного (`getName` — вернуть имя, `concludeTreaty` — заключить договор, `getHTMLFormat` — вернуть контракт в HTML-формате). Система именования рассматривается чуть ниже, но какой бы системы вы не придерживались, важно следовать ей на протяжении всего проекта. Если в пределах одного класса или нескольких классов имена методов будут принимать форму `get_name()`, `GetName()`, `getName()`, `Name()`, — разработчики будут тратить время на то, чтобы посмотреть название метода каждого класса, вместо того, чтобы запомнить одно название. По мере роста проекта на поиск (и на проверку путем компиляции и прогона приложения) будет затрачиваться все больше и больше времени, и разработчики будут испытывать соблазн положиться на свою интуицию, которая при таком разношерстном наименовании будет подводить. Как результат проект будет обрастать большим числом ошибок, найти которые не легко, т. к. с первого взгляда трудно сказать, правильно написан метод или нет.

То же относится к порядку аргументов: если дескриптор (файла, базы данных, строки) передается всегда вначале, все функции класса должны передавать его вначале. Ситуация, когда позиция параметра, общего для всех методов, кочует от метода к методу, приводит к тому же, что и в случае с разношерстными именами — для того, чтобы воспользоваться методом, необходимо будет обратиться к определению класса, чтобы узнать как расставлять параметры.

Договоренность о наименовании методов, членов, классов заключается в спецификациях в момент разработки проектной документации. Если вы не составляете документацию проекта, то по крайней мере не меняйте стиль наименования переменных посередине проекта.

Замечание

Отсутствие должного проектирования можно наблюдать во многих проектах и даже языках программирования. Как правило, этим страдают новые языки, на-

пример, Java и PHP. В PHP половина функций для работы с массивами имеет префикс `array_`, другая половина такого префикса не имеет. Строковые функции страдают другой напастью, параметр, через который функции передается текстовая строка, располагается в части функций первым, а в части функций последним. Запомнить порядок в первые несколько месяцев использования языка практически невозможно и приходится держать под рукой справочник.

Другой часто возникающей ошибкой при формировании интерфейса класса является несогласованный уровень абстракции. В листинге 9.32 все методы согласованы друг с другом и осуществляют те или иные операции с элементами договора. В листинге 9.33 демонстрируется объявление класса, в котором добавляются методы `NextContract()`, `PrevContract()`, `FirstContract()` и `CountContract()`, позволяющие работать с массивом договоров, реализуя модель итератора. Таким образом, один класс описывает и структуру самого договора, и структуру группы договоров.

Листинг 9.33. Несогласованный интерфейс класса договора (C++).

```
class Contract
{
    public:
        ...
        String getName();
        String getAddress();
        String getPhone();
        String getCondition();
        String getSumm();
        void concludeTreaty();
        String getHTMLFormat();
        String getPlainFormat();
        ...
        Contract NextContract();
        Contract PrevContract();
        Contract FirstContract();
        Contract CountContract();
        ...
        // Закрытые члены и методы
    private:
        ...
};
```

Таким образом, класс выполняет две разные задачи — такой ситуации следует избегать и разбивать класс на два независимых класса. Другим поводом для разделения класса на два независимых класса является ситуация, при которой половина методов работает с одной половиной данных, а другая половина с другой.

Еще одним критерием согласованного интерфейса является его ортогональность — под каждую задачу должен быть один метод, класс не должен предоставлять несколько путей решения одной задачи. Начинаящим программистам часто кажется, что чем разнообразнее интерфейс класса, тем он эффективнее. Ни к чему кроме путаницы и разношерстного кода это не приводит. Программисту, который далее модифицирует, сопровождает или отлаживает код, необходимо удерживать в голове несколько схем реализации задачи вместо одной. Исключение составляют методы, которые сильно отличаются по представлению результатов.

При реализации интерфейса рекомендуется не использовать в открытой части класса членов, так как это приводит к нарушению инкапсуляции и неудобно в плане расширения класса. Например, пусть класс `ArrayContract` имеет в своем составе член `count`, в котором хранится число договоров, загруженных в приложение из базы данных.

Листинг 9.34. Класс `ArrayContract` (C++)

```
class ArrayContract
{
    ...
    int count;
    ...
}
```

Даже зная, что здесь хранится число договоров, у пользователя класса возникает соблазн записать в член `count`, например, значение `-1`, для того, чтобы известить другую часть программы, что пользователь перешел в раздел ставок и больше не работает с договорами, после этого значению `-2` присваивается еще какой-то магический смысл, и пошло-поехало. Класс невозможно тронуть, так как на побочные значения его членов завязана половина программы, которая благодаря политике нарушения инкапсуляции превратилась в запутанный клубок.

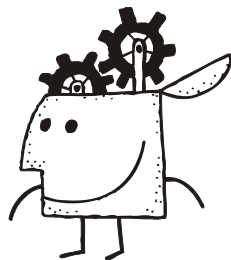
Хорошей практикой считается помещение переменной `count` в закрытой части класса и создание для него отдельных методов для установки значения (`setCount`) и извлечения (`getCount`).

Листинг 9.35. Модификация интерфейса класса `ArrayContract` (C++)

```
class ArrayContract
{
public:
    int getCount();
    void setCount(int);
    ...
private:
    int count;
    ...
}
```

Создание методов для членов классов позволяет не только предотвратить присвоение переменной корректного значения за счет проверки его в методе, но и вообще исключить операцию присвоения, как таковую. Более того, если в ходе создания Web-приложения возникает необходимость хранить в переменной `count` класса `ArrayContract` число договоров не только в текущей базе данных, но и соседнего сервера, исправить один метод `getCount()` будет гораздо проще, чем сотню обращений к переменной `count` по всему приложению.

Глава 10



Вместо заключения

Когда книга уже была практически готова к сдаче в издательство, мне пришлось консультировать трех человек, которые по своей специальности были как раз "IT-шниками". Все они были прекрасными работниками, существующая работа их устраивала, да и новую, в случае чего, они могли найти без особых проблем, но... У всех них были проблемы, по причине которых они практически не могли работать. Проблемы, к IT-области не имеющие никакого отношения. Работая с ними, я понял, что стоит написать еще одну главу, посвященную, скажем так, общефилософским вопросам. Жизненным вопросам. Потому что если вы, к примеру, не в ладах со временем и со здоровьем, если вы ощущаете себя неудачником, если вы негативно относитесь к окружающему миру и его населению, — то обо всем остальном (хорошем стиле программирования, взаимоотношению с заказчиками и т. д.) можно не беспокоиться. Потому что не понадобится. Сразу хочу сказать, что эта глава написана достаточно жестко. Это оттого, что в некоторых случаях добро должно быть с кулаками, и те вопросы, о которых идет речь в этой главе, как раз относятся к этой категории.

Управление временем

Начну с банальности: мы живем не вечно. И даже не очень много. А если сравнивать с геологическими масштабами, то откровенно мало. Рождаемся мы на свет голенькими, а уходим... Ну, в костюмчике там, в ботиночках, в рубашечке. И все. Больше ничего с собой не утащишь. То есть начали голыми, закончили в костюме. Небольшое положительное сальдо, как говорят экономисты. Можно сказать, не зря жизнь прожили. Правда, история человечества знает и таких, которые не только на пиджачок заработали. А еще и на память хорошую. Кто на сто лет, а кто и на столетия после себя. Не нужна вам память о себе? Деньги нужны? Тоже дело хорошее. Великое, я бы даже сказал. И внукам, и правнукам и пра-, пра-, праправнукам можно после себя много что оставить. Вон как Морган-основатель. И передают те-

перь из поколения в поколение Морганы-старшие Морганам-младшим свое дело. Или вон как Билл Гейтс. Который и на память, и потомкам на несколько поколений вперед... Правда, его очень многие ругают. И сильно. А знаете за что? Нет, не за Windows. И не за ошибки, там, всякие. А вот именно за это — за то, что он в конечном итоге и на память о себе, и потомкам много чего... А мы — только на пиджачок. И понимаем мы это в глубине души. И ругаемся от этого. Ах он, такой-сякой... Наше кровное украл... За наш счет, нашими муками. Только не он украл, а мы сами у себя крадем. Время крадем. Каждый день мы приходим на рынок и встаем за прилавков. И торгуем. Временем. Своим временем. Вернее, рынком и торгом это назвать сложно. Потому что на самом-то деле мы не торгуем, а отдаем. Просто так. Задаром.

— Эй, парень! Сколько тебе? Вечером пару часиков? Бери! За сколько? Ну, за пиво, за кружечку.

— Братан! День тебе? Забирай! За просто так. Хорошему человеку не жалко. Да и было б чего жалеть — всего-то день времени. Их, этих дней, у меня еще — о-го-го сколько!

— А для вас девушка мне и года не жалко!

И вот так каждый день. Парень вечерок прихватил, братан-друган два денька, девушка хорошенькая годик оттяпала. А у вас, возможно, для себя-то ничего уже и не осталось. Кроме долгов. Годик вы задолжали еще одной девушке, десять часиков — собаке соседей, которую будете выгуливать во время их отпуска, еще пару месяцев троюродному брату сестры жены, которому пообещали домик помочь построить... И это только оптовая распродажа. А сколько в розницу? Там в магазинчике дорогое полдня протолкались, мебель выбирали. Да не мы выбирали. А другие. А мы смотрели, как они выбирают. И мечтали. Что когда-то и мы вот так же... Такую же красивую и дорогую. Потом еще по улицам часик-другой послонялись, мечтая... О том, что когда-то будем по этим же улицам на роллс-ройсе. И непременно с личным шофером. Там пару часиков не решались в нужную дверь постучать, потому что боялись, что о нас подумают. Да, бросьте! Кто о нас что подумает? Кому интересно думать о тех, кто только на пиджачок да на ботиночки за всю жизнь...

А когда спохватываемся, когда припирает, когда врач говорит или сами догадываемся, что почти все наше время на исходе, оказывается, что не то, что человечеству про нас вспоминать, нам самим вспомнить особо нечего про свою жизнь. Кроме двух-трех крупных скандалов, пары-тройки значимых обид и поездки в Москву.

Примечание

Кстати, по одной из очень интересных теорий рак, к примеру, возникает из-за, так сказать, нарушений отношений со временем. Якобы наши клетки прекрасно

чувствуют, когда их хозяин не в ладах со временем и тоже начинают "чудить". А их нелады со временем приводят к тому, что они начинают перерождаться... Не знаю, как с научной точки зрения, а с жизненной примеров этой теории сколько угодно. И никаких особо заумных объяснений не надо для того, чтобы понять, что когда у человека временной клинч — ничего хорошего ждать не приходится. Потому что подобен такой человек двигателю, пошедшему вразнос. Шуму много, беспорядочные хаотические действия при практически нулевой выходной мощности. И конец близок. Так же и с человеком, который не в ладах со своим временем: вроде и бегал весь день и делал чего-то, а оказывается, ничего и не сделал практически. И также не стоит удивляться, если вдруг приключится какое-нибудь заболевание.

А ведь можно. Можно и по-другому. Эйнштейны и Менделеевы, Бетховены и Чайковские, Гейтсы и Форды и многие другие не из другой галактики на нашу грешную Землю прилетели. Так же, как и мы на белках замешаны. Только догадались в отличие от многих из нас до большего. Да нет, я не теорию относительности и Пятую Симфонию имею в виду. Я имею в виду, что догадались они до того, как свою жизнь так организовать, чтобы было в ней место теориям относительностей и Пятым Симфониям. То есть как со временем дружить догадались. А сколько не менее гениальных не написали, не спели, не открыли... А могли бы. Ума бы хватило. Времени не хватило. Потому что вместо того, чтобы открывать, петь и писать, они мечтали. Как спуют, напишут, откроют. А кто-то даже попытался сделать. Только в какой-то из вечеров не смог отказать другу детства Кольке, ныне дворовому опойке, выпить с ребятами стопочку за компанию. Да только назад уже не вернулся. Сначала компания не отпустила, потом сам не захотел, а потом уже и незачем было возвращаться. Два следующих раздела о них — о мечтателях и невозвращенцах.

Мечтать не вредно?

Мечтаете? Боевичок смотрите? Как там главный герой хуки налево и направо раздает. А в тех, кто от хука не хукнулся, он лезвием плюется. С пяти метров в левый глаз. И вот вы уже представляете, что главный герой — это вы. Что это не он, а вы хуки и апперкоты отвешиваете. А в следующем фильме, который через полчаса после того, в котором вы всех хукнули, ох как вы супостата пришпандорили с ноги. Так что у него нос в мозг провалился. Круто, ничего не скажешь. И вот, упиваясь собственной крутостью, вы легли спать-почивать, сил набираться. Чтоб на следующий день в то же самое время по тому же каналу в составе спецназа ГРУ уничтожить в Джелалабаде сводный полк наркоторговцев. Если в программе не ошиблись. Ну, а если ошиблись, значит, будете двумя часами позже по другому каналу помогать американскому спецназу проводить спецоперацию в джунглях Камбоджи. Вот только вечером, когда вы шли с работы, вас ждали не курьеры, вручающие пакет с новым спецзаданием, а наши русские шакалы. А вы

вместо того, чтобы отдать им имеющуюся при вас наличность и попросить прощения за то, что мало с собой взяли, вдруг вспомнили вчерашний фильм. И крикнули: "И-й-я-я!!!" Ну, как главный герой там орал, когда всех апперкотил. Шакалы изумились такому повороту дел и врезали вам так, что вы летели от улицы Пушкина до травматологического отделения вашей районной больницы. Потому что так всегда бывает. С мечтателями. Когда они с реальностью сталкиваются. И вот лежите вы в палате и слушаете музыку по плееру. Вместо того чтобы подумать, а с чего это вы здесь оказались. Надели наушники, а там такие песни поют... И вы представляете, что это не они, а вы поете. Ух-ты, какую ноту они взяли. Пардон, вы взяли. Аж рот в подражании открыли, и мышцы шеи напрягли...

А вы о чем мечтаете? О том, что когда-то бросите писать программы "на дядю" и откроете свое дело? А сами хоть одну свою программку помимо конторы протолкнуть пробовали? Пробовали? И что, получилось? Ничего не получилось? Потому что заказчик не понял, что ему предлагают по причине того, что в момент разговора с ним ваш голос дрожал на высокой ноте в диапазоне частот, близких к ультразвуковым? А ваш начальник, узнав об ваших поползновениях, бегал по кабинету аки взбесившийся енот и, в конце концов, сыграл с вами в известную игру с названием "накрылась премия в квартал"? При этом у вас из глаз брызгали слезы умиления — ведь мог и уволить. И теперь вы снова пишете программы "на дядю" и мечтаете по вечерам. Как вы приходите к заказчику и хорошо поставленным басом предлагаете ему свой продукт. И как там все восхищаются, заламывают руки в экстазе и тут же подписывают с вами договор на много тысяч американских купонов, в котором отдельным пунктом прописано, что все двери в этой фирме вы отныне можете открывать ногами. Ну, что ж. Мечтайте дальше. Вместо того чтобы поработать над собой вообще и над голосом в частности.

А вы мечтаете о том, как станете великим ученым? Как сделаете великое научное открытие, после осознания значимости которого вся научная общественность во главе с Президентом РАН к вашим ногам бухнется? И вот вам уже видится блестящая научная карьера, стада учеников вокруг, ловящих каждое ваше слово, не исключено, что и Нобелевская премия... Мечтайте. А если б не мечтали, а подумали бы трезво, то поняли бы, что сделать даже серьезное научное открытие по большому счету ничего не стоит. Точно вам говорю. Я делал. Надо лишь иметь мозги, желание работать и не бояться прослыть дураком. И можете считать, что открытие у вас в кармане. А вот "пробить" его, открытие это... Это очень непросто. Потому что академики у нас как паханы на зоне — каждый свой регион держит. И пускай какой-нибудь аспирантишка или доцентик новоиспеченный в подведомственном регионе попробует вдруг чего-нибудь открыть. Сделать-то вы открытие, можете, и сделали, и статью по этому поводу в научный журнал написали... А потом... Потом отдали ее на проверку какому-нибудь академику, который вас похвалил, по головке отеческой рукой потрепал. Только в статью, вами

написанную, включить "забыл". Бывает такое с академиками. И вот уже получается, что открытие сделали не вы, а академик при ближайшем сподвижничестве своего любимого ученика и еще пятерых наиболее преданных сотрудников института, академиком возглавляемого. Потому что, чтобы самому, в одиночку, пробить свое открытие, нужно быть как минимум доктором наук. Кандидатом — мало. Потому что это "на гражданке" звание кандидата наук слух обывателям ласкает. А в крупном университете, где их штук 850, знаете, кто такой кандидат наук? Это то, чем академики щи хлебают. Денежные. И этот пример с открытием еще не самый плохой. Тут хоть какие-то "высокие" переживания есть. А как правило, все происходит более банально. Без всяких открытий. А в виде каждодневной рутинной работы — постановки эксперимента по нанесению двести сорок девятой точки на калибровочную кривую.

Ну что, расстроил я вас? Получается, нельзя ничего сделать? Ни открытие, ни программу крутую? Можно. Все, что захотите. Можно и нападающих положить. Можно и открытие сделать. И программу крутую. Только делать надо, а не мечтать. И думать. Но не лежа на диване, а результативно думать. То есть — нетрадиционно. Потому что все традиционные мысли уже разобрали.

А то ведь как часто бывает? Жили-были муж с женой и был у них ребенок. Гордость семьи. В три года с табуретки читал стихи поэтов Серебряного века под слезы умиления гостей семьи. С соседскими детьми не дрался, штанов не рвал, короче мальчик-одуванчик. Пришла пора в школу идти. Только не понравилось ему в школе. Потому что там его били. А так как он штанов не рвал и с соседскими мальчишками не дрался, то сдачи дать не мог. "Ну, ничего, — думал он, — вот вырасту, стану сильным, всем им отомщу". Долгими вечерними часами сидел около речки, смотрел на воду и мечтал, как отомстит. А потом уже не мечтал, а просто ждал, когда школа закончится. И думал: "Вот поступлю в институт, — красота будет. Никаких обидчиков, учителей-придурков, не понимающих, что мне их биология нужна, как глухому магнитофон". Поступил. В первую сессию чуть не отчислили, несколько раз избили за то, что не дал списать... В общем, та же школа, только высшая. Ну, в смысле, что издевались там над ним на более высоком уровне. "Ну ладно, — мечтал юный студент, — пять лет не срок, зато хорошую профессию получу. Деньги буду лопатой грести, кандидатскую, глядишь, защищу, а может и докторскую". Выучился. Устроился на работу. Только и там все не слава богу: планы, склоки, начальники-полудурки... И понял он, что никакой кандидатской здесь никогда не напишет. А, идя вечером с работы, мечтал, как напишет книжку. Которая его прославит и от которой гонорары рекой литься будут. "Вот, — думал он, — напишу книжку, такая жизнь начнется! На работу треклятую ходить не надо, сиди себе и пиши. На встречи с читателями езди". Долго он так мечтал. И на работе мечтал, и дома. И когда начальник к себе "на ковер" вызывал и орал благим матом, он

мечтал об этой своей книжке, и когда жена пилила за то, что он даже мусор вынести не в состоянии. Он мечтал. Мечтал о своей книжке. Только не писал, а просто мечтал. Зато он уже видел, какой будет ее обложка. Он представлял, как принесет ее дуру-жене и как она бухнется перед ним на колени и будет просить прощения за то, что столько лет мешала его таланту. Он мечтал, как принесет ее полудурку-начальнику, положит на его дубовый стол и скажет... Ну, что-то скажет такое. Проникновенное и соответствующее моменту. Иногда он встречался с друзьями (когда жена отпускала), и он им рассказывал, что задумал написать книгу. И, пропустив стопочку, они вместе мечтали. Он мечтал о том, что, когда выйдет книга... Он мечтал. А всего одна клеточка в организме, не выдержав унижений, дуру-жену и полудурку-начальника, взяла и переродилась. В опухолевую клеточку. А она уже не мечтала. Она росла. Она свое дело знала. ...Когда он узнал, что у него рак, он снова мечтал. О том, как вылечится. И вот тогда... Эх, заживем! Он рассказывал врачам на обходах, как он будет жить и какую книгу напишет, когда вылечится. Врачи рассеянно слушали и вяло поддакивали. Потому что они такого за свою врачебную практику слышали очень много. Потому что в этих палатах каждый второй о чем-то мечтал до того, как сюда попал. И каждый первый мечтает о том, что он будет делать, когда отсюда выйдет. А он даже начал писать. В палате. Под стоны и хрипы умирающих больных, под запах разлагающихся частей тел. Он писал. И получалось красиво. Он уже не мечтал, а писал день и ночь напролет, даже когда уже сил не было, все равно писал. Потому что понял. Понял, как ему нужно было жить. Только не успел дописать. Потому что было уже поздно.

...Знаете, что такое цейтнот? Это когда кончилось время, отведенное на игру. И вроде, даже знаешь, как надо ходить, чтобы стопроцентно выиграть партию. Только уже поздно. Потому что флажок упал. Партия сыграна. В смысле — жизнь кончена.

И вот так с небольшими вариациями у очень многих. От мала до велика. Вы тоже мечтаете? Все еще мечтаете? Как сделаете, как напишете, как споете? И вроде бы знаете, что надо сесть и сделать, но все откладываете в долгую яму. Да, да в ту самую. Которая с червяками. Не надо так категорично? Всего лишь до завтра? Чтобы сегодня еще чуть-чуть спокойно помечтать, как завтра вы вдруг встанете в 5 утра (хотя до этого 10 лет вставали в 9), сделаете утреннюю пробежку (несмотря на то, что в последний раз бегали 15 лет назад, когда в школе учебную тревогу объявили), возьмете в руки гантели (если вспомните, как они выглядят) и... И значит вам конец. Потому что если отложили новую жизнь "до завтра", значит, отложили ее навсегда. Нет, конечно, не навсегда. Навсегда — это слишком долго. А всего лишь до той долгой ямы...

Но уж если вы сами так со своей жизнью поступаете, все откладывая ее и откладывая, то не обижайтесь на тех, кто вам досаждают. На начальника, вечно орущего, на вашу вторую половину, вечно всем недовольную, на со-

седей, которым все время надо чем-то помочь... А чего обижаться-то, если вам самим ваша жизнь не нужна? Вот нами и командуют все, кому не лень. Сначала родители, потом педагоги школьные и институтские, потом начальники... Потому и командуют, что видят: с вами можно делать все, что угодно без особого для себя вреда. Потому что когда вы хозяин своей жизни, никто вами командовать не будет. Потому что люди очень тонко чувствуют, кто слаб, а кто силен. Слабого — бьют, унижают, делают с ним, что хотят, а сильного — стороной обходят, а если и приближаются, то сугубо для того, чтобы к его мнению прислушаться.

О добре, которое нередко бывает с кулаками

Есть случаи, когда добро действительно должно быть с кулаками. И вопрос работы со временем — из их числа. Это как в медицине. При некоторых болезнях можно с пациентом посюсюкать, пожалеть его, пилюль сладеньких прописать и отправить лечиться недельки на три амбулаторно. А бывают ситуации, когда жизнь пациента висит на волоске. И надо действовать быстро, четко и без эмоций. Если надо, то жестко. Представьте себе, что на ваших глазах током ударило человека, и вот он лежит рядом с вами и не дышит. И пульса нет. А вы ему: "Миленький, ну очнись, пожалуйста. Тебя дома пирожок сладенький ждет". Бесполезно, не очнется. Не нужен ему уже ваш пирожок. Потому что в этом случае надо не уговоры вести и пирожки обещать, а максимально быстро делать сердечно-легочную реанимацию: искусственное дыхание и непрямой массаж сердца. И делая массаж сердца, можно даже сломать пару ребер — ничего страшного. Срастутся ребра. Лишь бы сердце удалось запустить, лишь бы жив остался. ...Теряющий время теряет жизнь. И значит, надо приступать к реанимационным мероприятиям. И если надо, то действовать жестко. По-другому, увы, часто не получается. В силу того, что люди нередко под видом заботы о других, заботятся только о себе, любимых. Любой психолог вам расскажет не один случай, когда родители под видом заботы о больных детях делали все для того, ...чтобы они не выздоровели. Сами, конечно, того не понимая. Простой и часто встречающийся пример. Мамаша приводит к психологу дочку, больную неврозом. Очень нередко в таких случаях психологи приходят к выводу, что лечить надо не дочку, а мамашу. Потому что детей до неврозов часто доводят сами родители (конечно, только из-за большой заботы о детях). И как только дочка начинает идти на поправку, мамаша с истериками и обвинениями в адрес врачей ("Вы мне совсем испортили дочь!") забирает дочь от врача. Парадокс, казалось бы? Да нет, закономерность. Причем часто встречающаяся. Два других случая, происходящих уже на моих глазах. Дочь-невротик, начитавшись умных психологических книжек, стала потихоньку понимать, что к чему, и самоизлечиваться от невроза. Чего не смогла перенести ее мамаша. Тоже невротик. И истерики закатывала, и в больницы "умирать" ложилась. Не сознательно, конечно. Она сама во все это верила, и

в больницы, и в свои умирания, и дочь, в конце концов, убедила. В том убедила, что книжки, которые она читает, — глупые, и что благодаря ей и ее книжкам мать чуть не дошла до инфаркта. Психика шестнадцатилетней девушки не выдержала такого испытания (психика взрослого человека такого не выдержит!), и она решилась на самоубийство. Съела упаковку таблеток. Ее откачали. Увы, ненадолго. Потому что потом на самоубийство решилась мать, написав в предсмертной записке примерно следующее "Я поняла, что неправильно тебя воспитывала и поэтому приняла это решение". Как говорится, без комментариев. Самоубийство матери было успешным, хотя, подозреваю, что она планировала его только инсценировать. Кстати, рассказывали, что на похоронах матери многие взрослые кретины (прошу прощения, но иного слова для них подобрать не могу) в открытую обвиняли дочь в смерти матери. Вторая попытка самоубийства ее дочери, которую она предприняла сразу после похорон, тоже, увы, была успешной. ...Почему так получилось? Потому что мать жила только для себя, искренне веря, что живет для дочери. А на деле, на подсознании заботой о больной дочери она оправдывала собственную неудавшуюся жизнь. А на деле она воровала жизнь своей дочери!

Еще пример. Тоже про попытку самоубийства. К счастью, неудавшуюся. Одного пятнадцатилетнего мальчика, который с криком "Не нужна мне ваша любовь, заберите ее себе" убежал из дома. От родителей. Которые его очень любили. Правда, любили. И, правда, что очень. Я нисколько не иронизирую. Вся беда в том, что **очень**. И по этой причине не давали ему ходить ни в секции рукопашного боя, ни в походы, никуда. А ему очень хотелось туда ходить. Они же его очень любили и боялись, что в секции он получит серьезную травму, что в походе его укусит какая-нибудь сумасшедшая и очень ядовитая кобра, специально в единственном числе заползшая в наши края, чтобы покуситься на жизнь их единственного сына. И так далее. А пацан просто хотел стать мужчиной. Только и всего. И никаких реальных предпосылок для серьезных опасений за его судьбу, при которых было бы оправданно столь серьезное вмешательство в его жизнь, не было. (Пример о том, когда такое вмешательство оправданно — чуть ниже.) Для ребенка жили эти родители? На словах и на сознательном уровне — конечно. Кто бы спорил. А на деле и на подсознании — для себя. Для своего спокойствия. А его интересы им были, как сейчас модно говорить, параллельны.

...Я это все рассказал для того, чтобы показать, что воровать время и жизнь могут даже очень близкие люди. Что уж говорить об иных... И что все это, нередко, делается с любовью... Что самое страшное. Под лозунгом "Я все сделаю для твоего счастья при условии, что ты останешься несчастным".

И, напоследок, в завершение этого раздела, небольшое авторское извинение. О том, что нередко примеры, приводимые в книге, с программированием не связаны. Потому что, к примеру, кровь, пардон, время, у всех вытекает одинаково. Что у программистов, что у экономистов, что у стале-

варов. И комплекс реанимационных мероприятий, проводимый в этих случаях, не зависит от того, кто вы по профессии. Поэтому если видите, что находитесь во временном цейтноте, если каждый день ничего не успеваете, если чувствуете, что все пролетает впустую и т. д., то имеете полное право действовать жестко. Иногда даже жестоко. Потому что выскочить из плена, не заколов парочку-другую часовых, как правило, не удастся. Потому что это ваша жизнь. Ваша, только ваша и ничья другая. И вы имеете полное право за нее бороться.

Скажи мне кто твой друг, и я скажу, кем ты станешь

Или, как метко сказал писатель А. Ильин, "с кем поведешься, о того и почищеешь". Помните, как в старинной поговорке говорится? "Как надену портупею, так тупею и тупею..." Это я о том, что бытие, как известно, определяет сознание. Что если родился шимпанзе вертлявым, человеком не станешь, что бы там Дарвин со последователи не говорили. Вон сколько уже ученые ждут, и все без толку. И мы ждем, как те ученые. Они свое ждут, а мы свое. Они ждут, пока шимпанзе очеловечится, а мы того, "что вдруг откуда ни возьмись" наш приятель (друг, брат, муж, сын, сосед, любовник — нужно подчеркнуть, недостающее вписать) Колька станет великим (бизнесменом, ученым, программистом, экономистом). Хотя все его приятели-кореша, с которыми он дружбу водит, "гимназий не кончали" и "Шиллера и Гете не читали". В общем, зря ждете. Не будет этого. А если вы действительно этого хотите, а не мечтаете, как персонажи предыдущего раздела, то брать надо вашего (друга, сына...) за шкуру и вытаскивать из этой компании. Как угодно. Здесь почти все средства хороши. Невзирая ни какие вопли типа "Что ж вы, падлы, меня перед друзьями унижаете?!" Как одни мои знакомые своего сына перед поступлением в институт. Было их четверо дружков-приятелей, друживших с детства, из которых трое, как бы так мягче сказать, были не "сильно интеллектуальными". Родители не были наивными людьми и между "дружбой до гроба" и будущим сына выбрали его будущее. Никакие методы, кроме силового, не помогли. Поэтому они по-смененно дежурили дома и караулили его, чтоб во двор не убежал, а к поступлению готовился. По вечерам, для профилактики, мамаша заливалась горячими слезами и играла с сыном в игру "Да за что ж нам с отцом такое наказание". Таская при этом сына за его роскошные кудри. Потом, успокоившись, они с отцом обсуждали план оперативных действий на следующий день. И так около года. "Купаться хочешь?" — "Пожалуйста!" Подгоняют машину к подъезду и везут на речку. Искупался — обратно в машину и домой. Конечно, было нелегко, даже очень трудно всем. И сыну. И родителям. И упреки взаимные были: "Как смеее меня свободы лишать?", "Вы ж меня перед пацанами унижаете"... Все было. Зато теперь своих родителей благода-

рит и в ножки им кланяется. Потому что из той компании один уже в гробу, а двое в тюрьме. И только наивный мечтатель может думать, что четвертого постигла бы иная судьба. И таких примеров с небольшими вариациями вы и сами наверняка много знаете. А если вы сами в таком положении, и взяты вас за волосы некому, то сядьте, подумайте и выберите. Что вам дороже. Эволюция или деградация. Да не мирового разума, а вашего собственного. И не надо успокаивать себя тем, что вы с 10 до 18 программируете там чего-то, а в своей любимой компании гуляете только по вечерам. Потому что прорвется где-нибудь ваша удаля молодецкая, обязательно прорвется. К примеру, на собеседовании, когда вы, получив задание, скажете все, что думаете о тех, кто его составлял, и обо всех их родственниках по женской линии. Пусть вполголоса. Главное, потом не удивляться, что получите отрицательный ответ, даже если все сделаете правильно. Или в споре с коллегой-программистом заедете ему правой в глаз. Потому что забудете, что в этой крупной IT-компании принята другая система аргументации, а не та, которой пользуются в вашем дворе. И не надо себе говорить, что "когда надо, вы соберетесь и будете вести себя прилично". Не бывает так. Чтоб вечером пить, гулять, отдыхать с честными пацанами, по какой-то там своей фене ботать, а утром чистым, выглаженным и приятно пахнущим предстать перед светлы очи начальства, получить задание на новую программу и весь день ее разрабатывать, изредка перебрасываясь с коллегами высокоумными замечаниями. Не бывает. Если только, конечно, вы какую-то разведшколу не окончили, в которой учат, как надо, чтоб вечером с корешами "за жизнь", утром с дипломатом "за политику", а к обеду инвалидом прикинуться и милостыню пособирать (для того чтобы было на что вечером с корешами посидеть). А нас, окончивших не разведшколу, а "среднюю №...", в этом случае ждет один и двух вариантов: либо вас с работы уволят, либо кореша во дворе на "правеж поставят за то, что бакланом стали". Потому что крайне сложно быть своим в столь разных социальных группах. Или, как более жестко сказал поэт И. Губерман, "Нельзя одной и той же попой сидеть во встречных поездах".

Еще пример на эту же тему. О том, как один очень одаренный юноша-музыкант победил на областном конкурсе по вокалу. Парень действительно был очень одаренный. Кроме того, что пел, он еще и блестяще играл на пяти, если мне не изменяет память, инструментах, с помощью которых сам же себе и аккомпанировал. Ему музыкальные педагоги прочили быструю и блестящую карьеру в музыке. Уже были даже подписаны какие-то контракты на его участие в достаточно денежных проектах, что его очень радовало, так как это позволяло быстро выбраться из той нищеты, в которой они жили долгие годы. (Его воспитывала одна мать, тратя при этом немалую часть своих доходов на его обучение, и, к его чести, он эти затраты оправдывал.) Также была договоренность о продолжении его учебы в московской консерватории, а в перспективе, вроде бы, даже и за границей. Но... Но не случи-

лось. Ни консерватории, ни "заграницы", ничего. Потому что в тот день, когда он эту премию получил, его старые дружки-приятели уговорили его это дело отметить. Ну, горды они очень за него были, так как в их дворе это был первый человек, который выбился выше вечерней школы. Он поначалу не соглашался, но уговорили. Друзья все-таки. Все проверенные, и их даже жалко немного — у него получилось выбиться, а у них нет. Да и гордость за него на их лицах прямо-таки светилась. Потом они пели и пили, пили и пели. Вернее он пел, и все вместе пили после каждой песни. А потом они убили человека. Были уже "в кондиции" и какой-то прохожий вроде бы что-то им сказал. Или им просто не понравилось, как он шел. И стали они его бить. И вроде не сильно, и вроде убивать не хотели, но — убили. Наш одаренный музыкант в прямом смысле в избиении не участвовал, и, говорят, даже пытался их остановить. Только в конце концов вышло иначе — на суде все говорили, что он один и участвовал, а семь человек никак не могли его остановить. Что мы там говорили? Друзья, гордость? Гордость и дружба это, конечно, хорошо, только когда впереди зона показалась, многие про гордость и дружбу забывают. А помнят лишь об инстинкте самосохранения. Потому что гордость и дружба это чудесно. На воле. Но зона впереди... Уж не знаю, так они, эти друзья, думали, или не так, но, в общем, оды о своей непричастности к этому мокрому делу они пели слаженнее любого профессионального хора. И, в общем, сел он. Один за всех и за то, чего не делал. В том числе и потому, что на суде молчал. От шока, вызванного и тем, что случилось, и поведением своих "друзей". Для которых он был "гордость". Вот, как говорится, и сказке конец. Потому что кончилась сказка и началась жесткая, как рашпиль, жизнь. Без лауреатских дипломов, московской консерватории, больших денег и прекрасного баритона, но с тюремной вонью, баландой, зековской робой и сиплым шепотом. А могла быть и сказка, если бы в тот вечер не пошел. Или если хотя бы ушел вовремя. Ну, не хочешь ты друзей обижать, не обижай. Выйди и выпей с ними три стопки, спой пару песен. И уйди. Потому что их жизнь с твоей уже давно не имела ничего общего. Потому что ничего страшного бы не было — даже они бы поняли и приняли. Но... Что было, то было.

Я, кстати, сам пытался учиться в вузе, одновременно подрабатывая на реч-флоте крановщиком плавкрана. По четвертый курс, пока работал, учился так себе, с троечки на четверочку. Бывало, и двоечки проскакивали. Причем по предмету, который достаточно хорошо знал еще в школе. Поэтому в итоге в бакалаврском дипломе имею четыре позорных тройки по основным предметам. А с четвертого курса по шестой (магистратура), как перестал работать, получал почти одни пятерки. И магистратуру закончил на одни пятерки. Кстати, одновременно с учебой и флотом я еще пытался заниматься научной деятельностью. Тоже ерунда получилась: одна статья и тезисы за два года. Так себе результатик. А вот потом, занявшись наукой целенаправ-

ленно, результаты действительно красивые получил. А заодно и все те премии, которые приведены в моем авторском профайле на обложке. Я ничего против речфлота не имею, наоборот, очень с большой теплотой вспоминаю то время и нисколько не жалею, что такое время у меня было. Просто на своем примере лишний раз объясняю, что некоторые вещи совмещать трудно, и пытаться, к примеру, освоить математический аппарат квантовой механики, работая крановщиком плавкрана, дело практически безнадежное.

Упражнение: куда уходит время?

И год как день, и день как жизнь.
Кругом страдания и невзгоды.
Мы ж экономим чьи то годы
За счет непрожитых своих...

Л. Филатов

Вы ничего не успеваете? Все время находитесь во временном цейтноте? Давайте выполним несложное упражнение. Возьмите листок и ручку и распишите по часам, что вы делали вчера. И сегодня. И так каждый день. Причем пишите все до мельчайших подробностей. Сколько минут у подъезда с соседями простояли, сколько минут автобус ждали и т. д. Интересные результаты получаются, когда такую статистику соберешь. По два часа в день с друзьями за пивком сидим и говорим, говорим... Ни о чем. По два часа в день в игрушки компьютерные режемся. Еще часик — на просмотр новостей, причем обязательно по всем каналам. Четыре дня в месяц на дни рождения улетело. И ведь не бог весть, какие друзья приглашали, просто развеяться захотелось. Да и по мелочи немало набегает. По 30 минут ждем нужного автобуса на остановке, в то время как до работы можно добраться пешком за 10 минут при сильном встречном ветре. Зато говорим, что у нас нет времени, чтобы гулять. Или вдруг выяснится, что мы с соседями у подъезда по 30 минут каждый день болтаем, сами того не замечая. Зато ребенок обижается, что сказку ему на ночь не прочитали — некогда было.

Когда я для себя составил такой временной график, то выяснил, что теряю около 5 часов времени каждый день. Просто ни на что. У вас не так? Все расписано по минуткам? Давайте посмотрим. Так, "просмотр кинофильма — два часа". Да нет, я совершенно не против просмотра кино, но ведь вы же его, оказывается, каждый день смотрите. Вы что — кинокритик? Нет, просто кино смотреть любите? Ну, тогда не нужно возмущаться, что вам времени не хватает, а надо просто честно себе признаться, что одна из целей вашей жизни — просмотр фильмов. Чужих. В смысле, снятых не вами. Потому что при таком подходе "свое кино" вы снять не успеете. А вы не успели на работе написать программу? Идем к вам на работу и смотрим, почему. Понятно. Я бы на вашем месте тоже не смог. Потому что сложно пи-

сать программу, общаясь по "аське" с парой десятков человек одновременно. Благодарите Бога хотя бы за то, что он сделал вас программистом, а не, к примеру, диспетчером аэропорта или водителем. Хотя не страшно, если ошибетесь или не сделаете чего-то. Хотя в тюрьму не посадят и живы останетесь.

Так что же получается? Все вообще распланировать? И стопочки лишней с приятелем не пропустить, и кружки пива с другом не выпить? Тут, как говорится, каждый выбирает по себе. Знаю немало людей, которые временем своим распоряжаются если не на 100%, то уж на 90% точно. И несколько не жалеют о том, что где-то с кем-то недоболтали или недопили. Потому что "взяли вершину". Постояли чуть-чуть, отдохнули, красотою невиданными повосхищались и дальше пошли. Другую вершину брать. Я сам, честно говоря, не могу похвастаться тем, что очень эффективно использую время. Такой вот плохо организованный уродился. И на гитаре побренчать мне хочется, и пивка с друзьями попить, и с соседями потрепаться... Но, выполнив это нехитрое упражнение, хотя бы научился "считать потери" и знаю, где и сколько теряю. А узнав, что в день около 5—6 часов уходит просто "в никуда", пришел в ужас и понял, почему я не успеваю всего того, что хотелось бы успеть. И сумел отвоевать у себя самого пару часов. Не могу сказать, что слишком много. А то время, что не удалось полностью отвоевать, попытался тратить с большей пользой, чем раньше. Хочется, к примеру, на гитарке побренчать? Бренчи. Но не на трех аккордах каждый день одно и то же, а разучивая каждый раз что-то новое. Чтобы не стыдно было на каком-нибудь банкете гитару в руки взять. (Очень, кстати, помогает завязыванию новых знакомств. Если хорошо играть, конечно.) А можно не только для банкета. Можно и более профессионально. Для дела. В прямом смысле слова. Спросите, где, к примеру, сотруднику IT-студии найти прямое применение своим музыкальным талантам? Да где угодно. Делаете вы, к примеру, мультимедиа-презентацию, а нужную музыкальную тему подобрать никак не можете, хотя и знаете, какой она должна быть. Взяли и наиграли сами. Вот тогда можно сказать, что каждодневное бренчанье пошло на пользу, и не считать это время потерянным. И с соседями, оказалось, можно трепаться не просто так, а на интересующие меня темы. Пивка попить вечером с друзьями? Никаких вопросов. Хотя каждый вечер. Если обсуждать попутно что-то полезное.

Упражнение на развитие временной интуиции

У многих беды со временем происходят из-за хронического неумения оценивать время. Проверить, как у вас обстоят дела в этом плане, очень просто. Для начала попробуйте поугадывать, за сколько времени вы сделаете то или иное дело. Причем дела выбирайте "разномасштабные". В смысле, разные по временной длительности. Начинайте с малого. Угадывайте для начала,

сколько времени вам понадобится для того, чтобы сделать небольшое по длительности дело: уборку квартиры, поход в магазин и т. д. Затем переходите к делам, длящимся несколько дольше, к примеру, день-два, допустим, написание небольшой программы. Ну и, наконец, попробуйте угадать, сколько времени вам понадобится на какой-нибудь "долгострой": написание большого проекта, серьезной статьи. Наиболее страшна ошибка в сторону преуменьшения планируемых временных затрат — это показатель нереальности вашего планирования. Вещь коварная, ни к чему хорошему еще никого не приводившая. Накапливаемые ошибки, нарастающие как снежный ком, в конце концов, могут и придавить...

Несколько советов по организации времени

□ Гибкое планирование.

Без планирования никуда. Но и крайне четкое планирование не менее опасно, чем когда вы вообще ничего не планируете. Две крайности, ни к чему хорошему не приводящие. Полностью "беспланный человек" подобен неуправляемой лодчонке, затерявшейся в морских волнах и плывущей туда, куда подует ветер. Антипод — человек, планирующий все и всегда до мелочей, до минуты — тоже малоприятная перспектива: любая случайность, вторгшаяся в его планы (а без случайностей не бывает), может выбить такого человека из колеи очень надолго. Истина, как ей и положено, где-то по середине.

□ Созидательное планирование.

Термин, только что мной придуманный. Речь вот о чем. Если в предыдущем абзаце мы говорили о планировании в стиле "Пойти туда-то, сделать то-то", то сейчас разговор о планировании в несколько ином смысле. Поясню на примере. Допустим, на четверг запланировал я встречу с кем-либо. Если просто запланировал, а не подготовился, то велика вероятность того, что встреча пройдет не так, как нужно, и это время будет потеряно, так как в лучшем случае придется встречаться по тому же вопросу еще раз. Если же к встрече подготовиться, пусть даже по пути на нее, результат будет намного продуктивнее. Опять же: вряд ли стоит продумывать все до мелочей — вполне достаточно определить главные направления, основные аспекты. И такое планирование тоже должно быть гибким. Точно так же с телефонными звонками. Не просто так снять трубку, а на несколько секунд остановиться и подумать, что же надо сказать. Чтобы потом еще три раза не перезванивать, вспоминая, что вот это забыл сказать, а еще и вот это... По себе знаю, даже банальный поход на рынок происходит результативнее (и быстрее!), когда перед этим на листочке четко себе выпишу, что надо купить.

❑ Разборка с текучкой.

Старайтесь не накапливать текучку: с текущими оперативными делами разбирайтесь быстро и жестко. Нет более удручающего зрелища, чем смотреть на свой стол, заваленный листочками с надписями "Сделать то-то", "Не забыть позвонить туда-то", "Отправить письмо тому-то". Для некоторых видов текучки можно выделить отдельное время: к примеру, когда у меня возрос объем личной бумажной корреспонденции, я через некоторое время постановил для себя, что хожу на почту отправлять письма два раза в месяц. А поначалу бегал чуть не каждый день, теряя на этом немало времени.

❑ "День святого никогда".

Устройте себе один день в неделю, когда ничего не делаете. Да, да — как во всем известной еврейской субботе. Для себя я этот день называю "днем святого никогда" (название одного из стихотворений Б. Брехта). И в этот день позволяю себе лениться сколько хочется, до изнеможения и делать всякую белиберду: детективы читать, в Интернете вволю ползать, фильм какой-нибудь посмотреть, наконец, просто поваляться на диване. А если что-то захочется поделать — запрещаю себе, просто не даю себе в этот день ничего делать. Ни в коем случае. Максимум — записать дельную мысль для книги или идею для бизнеса. Все остальное — побоку. Зато на следующий день это ничегонеделанье окупается сторицей.

❑ Градация дел.

Старайтесь сортировать свои дела и отделять их друг от друга. Срочные, важные, на обдумывание и т. д. Не копите "висяков" — если какое-то дело зависло, быстро разбирайтесь в причинах оно и принимайте решение: срочно исправить, отложить до какого-то момента или вообще отменить.

❑ Совмещение дел.

По возможности старайтесь совмещать дела. У нас, к примеру, текущие и каждодневные ответы на форуме часто идут и в книги.

Осторожно — хронофаги!

Хронофаги — это пожиратели времени. Вашего времени. Того самого времени, из которого складывается ваша жизнь. А значит — это пожиратели вашей жизни. И бороться с ними надо как можно беспощадней, так как вы боролись бы с тем, кто в темном переулке собрался покуситься на вашу жизнь. Чтобы вы лучше представляли, кто такие хронофаги, я процитирую блестящую статью о хронофагах Андре Моруа из его книги "Письма к незнакомке".

"Хронофаг — это чаще всего человек, у которого нет настоящего дела, и который, не зная, на что убить свое время, решает заполнить свой досуг, по-

жирая ваше. Наглость этой твари невероятна. Он пишет авторам, с которыми не знаком, требуя немедленного ответа. Он поведает вам о своей жизни и расспросит о вашей. Не надейтесь умаслить хронофага, бросив ему поглотить частичку вашего времени. Он ненасытен. Ваша терпимость побудит его являться снова и снова, писать вам и всячески надоедать.

— У меня много работы, — неуверенно скажете вы.

— В самом деле? — спросит хронофаг, — как интересно. И над чем же вы трудитесь?

— Над романом.

— Над романом? Но вся моя жизнь — роман!..

И вот его точно прищипорили. Полночь застанет вас на том же месте. Если он умудрится заманить вас к себе — вы погибли. Вы — кость, которую он затащил в свою конуру, и, уж будьте уверены, он обгложет вас дочиста. А если он натравит на вас своих приятелей, вас станет пожирать целая свора хронофагов. Эти твари группируются в сообщества и охотно делятся друг с другом добычей.

Отсюда мораль: держитесь с хронофагами твердо и беспощадно, истребляйте их. Мягкостью и шепетильностью ничего не добьетесь. Напротив, эти-то качества и создают микроклимат, в котором хронофаг благоденствует. Он необычайно живуч, и его нужно изничтожать. Мне претит насилие, но в данном случае оно необходимо".

Конец цитаты. Мне к этому прекрасному описанию хронофагов, сделанному А. Моруа, добавить почти нечего. Лишь два момента. Если вам не повезло настолько, что ваш руководитель — хронофаг, то дело совсем плохо. И мой вам совет, постарайтесь сделать так, чтобы он перестал быть вашим руководителем. Найдите другое место работы, в конце концов. Потому что ничего стоящего из этого сотрудничества все равно не выйдет. Ведь руководитель-хронофаг обладает еще одной паршивой особенностью: он, по понятным причинам, ничего не успевает. В том числе не успевает он выполнять и свои прямые руководящие обязанности. Значит, он будет их на кого-то перекладывать. И если вы относитесь к нему лояльно, то перекладывать он их будет на вас. Потому что перекладывают всегда на того, на кого проще переложить. А потом он вас же и будет ругать за то, что вы с этими обязанностями не справились. И второй момент. В *главе 4*, которая посвящена искусству ведения переговоров, мы нередко говорили о том, что клиента, грубо говоря, надо "ублажать". Из этого правила существует, пожалуй, единственное исключение: этого не надо делать, если ваш клиент — явный хронофаг. И если у вас нет достаточной психологической изворотливости, чтобы и дело сделать, и жертвой не пасть — лучше бегите от этого клиента. Хотя бы потому, что пока с ним возитесь, десяток других упустите.

О пользе разумного эгоизма

Для начала давайте скажем друг другу правду, от которой, как и от любой правды, сначала несколько не по себе, но зато потом легче всем. Правда заключается в следующем: все мы эгоисты. А альтруизма нет в принципе. Альтруизм придумали идеалисты, которые сами такие эгоисты, каких еще поискать. А все эгоисты делятся на умных, вернее разумных, и глупых.

...Имея счастливую возможность наблюдать за людьми в самых различных ситуациях, от банальной семейной Санта-Барбары до случаев экстремальных, пришел к выводу, что если кто-то говорит о каком-то альтруизме, он лжет. Причем лжет бессовестно. Правда лжет, как правило, в первую очередь самому себе. Понимаете ли, какая штука: неэгоистов не может быть по определению. По биологии. И не надо в слово "эгоист" вкладывать какой-то ругательный смысл. Эго в переводе означает всего лишь Я. И ничего более. А эгоист — это человек, думающий о себе. Так что давайте не будем себе врать и будем жить и работать только для себя. Потому что, когда люди живут для себя, то все только в выигрыше.

Ведь если я живу для себя, то я сделаю так, чтобы мои дети выросли умными и счастливыми, получили достойное образование и стали достойными людьми. Потому что мне, как человеку, живущему для себя, будет крайне неприятно, если мои дети вырастут придурками. А если я пойму, что в силу тех или иных причин не в состоянии обеспечить своим детям достойное будущее, то я лучше вообще не буду иметь детей. Потому что при таком раскладе ничего хорошего ни мне, ни им не светит. (Примеров сколько угодно на каждом шагу.)

Если я живу для себя, то я буду писать хорошие книги. Потому что, во-первых, мне, как и любому человеку, приятно, когда хвалят мои работы и неприятно, когда их ругают, а, во-вторых, хорошие книги приносят хорошие гонорары. Да, когда я пишу, я думаю только о себе, но при этом я обязан думать и о своих читателях, о тех проблемах, которые их волнуют в данный момент. В результате — все довольны.

Если я живу для себя, то я хорошо буду делать любое дело, которым занимаюсь. Или, если не могу что-то сделать хорошо, то лучше за это не возьмусь. Потому что мне первому будет неприятно, если я дело сделаю плохо. Я же его для себя делаю, а не для "дяди", денег или медалей. А деньги и медали — они тоже будут. Только сами по себе. Как второстепенное дополнение.

Если я живу для себя, то буду думать о своем здоровье. А значит, не доставлю своими болезнями хлопот своим близким.

Если я живу для себя, то буду заниматься благотворительностью. Не потому что бесконечно влюблен в тех, кому помогаю, а потому что знаю, что "де-

литься надо". И знаю, что те, кто "не делится", ничем хорошим не заканчивают. Что бы там по телевизору не показывали о красивой жизни хапнувших и не поделившихся. (По телевизору почти всегда самое интересное как раз и не показывают.)

И так далее. Вот это все и называется разумным эгоизмом. А если по-другому, если "для семьи и народа", то может так стать, что и семьи не будет, и народ презирать станет. Я не так давно консультировал одного парня, системного администратора по профессии. Очень, надо сказать, уважаемого и хорошо оплачиваемого системного администратора. А проблема у него была в том, что он не мог работать. Ну, не мог и все тут. За все свои большие деньги и за хорошее отношение коллектива — все равно не мог. Ночами не спал, потому что не мог спать, зная о том, что завтра — снова на работу. И на работе уже работал все хуже и хуже. ...Не буду дальше описывать все его перипетии, скажу лишь, в чем дело было. А дело было в том, что он кормил всю свою семью и всех родственников. Один. На себя ничего не тратя. Сюда тысячу, туда тысячу, сестре двоюродной безработной тысячу, которая не могла работать, так как у нее был невроз в связи с тем, что ее муж пил... Тысячу — на кодирование этого мужа, тысячу — на лечение невроза его жены, своей двоюродной сестры, пять тысяч — родителям, еще пять — не сумевшему устроиться в жизни брату... И так далее. А на себя уже ничего не оставалось. И, как бы он от этой мысли не убегал, оправдываясь своими высокими поступками, мысль эта его грызла. Ну, действительно ведь обидно: пашешь, пашешь, и все отдай. Зачем работать-то тогда? Вот эта мысль подсознательная и не давала ему работать. Когда мы до нее добрались, и все выяснили, дела пошли на лад. ...Я не говорю о том, что, к примеру, родителям помогать не надо. Наоборот. Я сам говорил выше, что "делиться надо". И не только с родителями. Но — именно делиться, а не отдавать все! Потому что в его случае на родителях вполне можно было бы остановиться — все остальные на свое пропитание вполне могли заработать и сами. Что, кстати, и случилось, когда он перестал их спонсировать. Попутно выяснилось, что муж двоюродной сестры и сам получал не самую плохую зарплату. Которую пропивал. (Действительно, зачем в семью деньги носить, если у жены есть родственник-спонсор.) Вот как бывает, когда живешь не для себя. Получается "ни себе, ни людям".

Если кого-то из читателей тезис "Живите для себя" возмущает (он поначалу возмущает очень многих, и меня тоже сначала возмущал), то позволю себе привести еще один пример, очень наглядно иллюстрирующий это правило. Пример достаточно кровавый в прямом смысле этого слова. Мне пришлось как-то оказывать первую медицинскую помощь парню с ранением бедренной артерии. Это одна из самых крупных артерий, пары минут кровотечения из которой хватит для того, чтобы человека не стало. Но если застать ситуацию вовремя, можно вполне помочь, что я и попытался сделать. Если бы не одно "но". Оказать ему помощь меня не пускала его, как позже выяс-

нилось, мать, которая легла на него всем своим телом, голосила во весь голос и на мои попытки оттащить ее отвечала, что, мол, уйдите все и не трогайте его, неужели вы не видите, как ему плохо. А ему действительно было уже не просто плохо, а очень плохо. Любила ли мать своего сына? Безусловно. Но, опять же, вся беда в том, что очень любила. И наверняка жила "только для него". На словах. А на деле из-за этой слепой любви в критической ситуации потеряла рассудок и чуть было своего же сына не угробила. Потому что, пока я с ней справлялся, уходили драгоценные секунды. А в таких ситуациях секунды промедления смерти подобны в самом прямом смысле этого слова. Этот пример я привел для того, что он очень хорошо иллюстрирует, что живем мы все на деле для себя. Чтобы не говорили на словах. И эта женщина тоже жила только для себя. Потому что если бы она действительно жила только для сына, она бы не мешала другим оказать ему помощь, а, живя для себя, и поняв, что лишается объекта своей любви и, поняв, что жизнь без него теряет для нее смысл, потеряла рассудок, что едва не привело к трагическому исходу.

Самое главное условие достижения успеха

Знаете какое? Наверняка знаете: для того, чтобы чего-то добиться, надо пахать. Все время. Если не пахать, то ничего не получится. Это уж точно. Потому что само по себе ничего не бывает. Вернее бывает. Само по себе всегда бывает только плохо. Даже если сначала все очень хорошо. Посмотрите на тех детей известных людей, которые палец о палец не ударили для того, чтобы стать достойными людьми. Потому что они считали, что у них уже все есть, что делать им ничего не надо, так как все и так будет хорошо: папа с мамой позаботились. Ну и кем они при таком подходе стали? Правильно. Тем сбродом, про который мы говорим, что вот этих лучше бы при рождении... Потому что мало того, что пьют, колются, так еще в этом состоянии по улицам ездят, людей сшибают... И сами живут очень недолго. Если, конечно, то, что они делают, можно назвать жизнью. Если бы вы знали, сколько детей великих и известных людей находятся на излечении в наркологических VIP-центрах... И сколько их оттуда ежемесячно выносят. Куда? На кладбище, конечно. Куда еще оттуда выносят?

И это еще хорошо, когда вы дитя известных людей. Хотя, как мы видим, даже они, имея прекрасный потенциал для развития себя, умудряются все про... Пропустить. И ничего путного не сделать. Что уж тогда говорить про нас, грешных, у которых нет мам и пап-министров или банкиров, дедушек-маршалов, бабушек — великих поэтесс... Но, тем не менее, все мы почему-то думаем, что станем известными и великими. Ну, если не великими, то богатыми точно. А между тем, пословица про то яблоко, которое от яблони

очень близко падает, — это очень верная пословица. Потому что научное обоснование имеет. Про генетику слышали? Которая говорит о том, что мы рождаемся не бог весть какими индивидуумами, а по наследству нам передается многое из того, что присуще нашим родителям. Цвет глаз, группа крови, предрасположенность к чему-либо, черты характера, — а значит, и модели поведения в тех или иных ситуациях, а, значит, — и возможности. В общем, "что посеешь, то и пожнешь", как в другой генетически верной пословице говорится. Теперь понаблюдаем, как рассуждают многие из нас, с генетикой не знакомые. Или знакомые, но считающие, что лично к ним вся эта генетика никакого отношения не имеет. Итак, мама-лифтер, папа-слесарь, а мы уверены, что станем великими актерами и актрисами, известными учеными и государственными деятелями. Что будем по заграницам разъезжать несмотря на то, что наши родители дальше своей деревни сроду не выезжали, что будем богатствами несметными владеть несмотря на то, что папа с мамой всю жизнь не могли концы с концами свести, что будем великими учеными, хотя родители даже института не закончили... Да, вот такие мы оптимисты. Уверены вот, и все тут. И ничего не предпринимаем, чтобы под эту уверенность хоть какой-то фундамент подвести. И обижаемся, когда нам говорят, что все эти наши мечты, скорее всего, просто мечты. При этом, совершенно справедливо ожидая, когда сажаем в саду свеклу, что вырастет свекла. А не виноградная лоза. Это мы знаем. И как на идиота посмотрим на того человека, который, глядя на капусту, с восторгом воскликнет: "Ой, у вас по осени какие сочные плоды манго на этой грядке вырастут!" Действительно, дурак он что-ли? Какие манго, когда капусту сажали... Не хотите про капусту? Пожалуйста, другой пример. Представьте себе, что у вашей дворняжки после, скажем так, любви с дворняжкой из соседнего двора родились вдруг неписанной красоты щенки чау-чау. Что говорите, не бывает такого? Правильно говорите — не бывает. Это мы все понимаем. Что от дворняжек рождаются только дворняжки. А не пекинесы. Не понимаем лишь, что с точки зрения генетики мы в этом смысле ничем от свеклы, капусты и дворняжек не отличаемся. И понятно, почему не понимаем. Потому что понять такое, значит, перестать жить радостно и беззаботно... Потому что все верим в свою исключительность, и признать обратное... Это все равно, что тут же голову на плаху положить.

Так что же выходит — ничего нельзя? Да нет, можно. И та же генетика говорит, что можно. Можно сделать так, чтобы на вишневом дереве вместе с вишнями абрикосы росли. Или просто одни абрикосы, без всяких там вишен. Только для этого надо к вишне абрикос привить. А если тут можно, то и нам можно. Стать министрами несмотря на то, что наши предки — дворники в десятом поколении. Только для этого тоже надо прививками заниматься. Не теми, которые от гриппа, а теми, которые от безнадёги: теми, при которых к нашим, унаследованным, чертам характера прививаются другие. Но для этого нужно костыли лечь, но попасть в социальный круг более

высокий, чем тот, который по наследству передали вам родители. Потому что генетика утверждает, что если к вишне привить вишню, винограда не получится. И если вы будете развиваться в том же обществе, что и ваши родители, то вы просто повторите их судьбу. И станете не тем, кем мечталось, а обыкновенным папой Ваней или мамой Верой. Ведь именно для того, чтобы попасть в иной социальный круг, потомственная деревенщина Ломоносов в Москву приплелся. И стал тем, кем стал — Ломоносовым. Именно поэтому многие наши родители, вслед за Ломоносовым, из деревень уезжали в города. И пытались поступить в институты или хотя бы техникумы. Чтобы хоть на ступень выше — так как нередко их родители, наши бабушки и дедушки, часто даже школу не заканчивали — война была в то время. Если же ваши родители по наследству передали вам очень выигрышный социальный круг, то ваша задача — его не потерять, как потеряли те, о которых мы говорили в начале этого раздела.

Вдох глубокий, руки шире!

Теперь о здоровье. Это тоже один из краеугольных камней успешного ведения дел, чем бы вы не занимались. Если вам доводилось испытывать серьезную боль, когда сознание сужается до микроскопических размеров, и когда в нем ни на что не остается места, кроме этой самой боли, — вы меня поймете. И вы в этом случае прекрасно знаете, что в таком состоянии нет места ни бизнесу, ни друзьям и подругам, никому и ничему... Только боль. Только бы вырваться, только бы выскочить. Где все эти проблемы, которые еще вчера трогали и занимали, где мысли о сорванном контракте? Боже, какая мелочь... И тут ты прекрасно осознаешь, что только здоровый физически человек может быть по-настоящему успешным. А те, кому, слава богу, не довелось ничего такого испытать, просто поверьте на слово, что это так. А если не хотите на слово, проведите простой эксперимент. Поднимите высоко руку и стукнетесь локтем об стол. Что, говорите, уже стучались? Не раз? Знаете, что больно? А сейчас вы не просто так стукнетесь, а со смыслом. Потому что после того, как ударитесь, и как вас накроет волна боли, попробуйте подумать о текущих делах. Не получилось? Поздравляю. Первый шаг на пути к успеху. Теперь вы точно знаете, что первично. Что не бывает в нездоровом теле здорового духа. И что больному телу не до высоких идей, не до бизнеса, не до программирования, не до... Не до чего. Больное и дряхлое тело подобно болоту — вязкому, склизкому и иногда смердящему. Делать ничего не хочется, хочется только валяться на диване, потягивать пивко и отупевшим взглядом буравить потолок... Любая здоровая идея увязнет и погибнет в таком теле-болоте, точно так же, как увязает и погибает в настоящих болотах то живое, что туда попадает.

Каким видом спорта заниматься? По большому счету, без разницы. Бег, плавание, туризм, рукопашный бой, лыжи, все это вместе взятое... Лишь бы не преферанс под водочку и не шахматы под сигареточку. И, наверное, не бодибилдинг. Потому что никакого толка от бугров мышц нет. Да и часть мозгов, говорят, со временем в мышцы перетекает.

Нет времени на спорт? Почитайте *раздел "Управление временем"* и найдите чуть-чуть. Найдется обязательно. Но это так — полусутка. А если серьезно, то возможно и времени нет именно потому, что спорт в долгий ящик задвинули. Как говорит М. С. Норбеков, "спорт — это машина времени: у того, кто занимается спортом, в сутках 240 часов, а у того, кто нет, — 24 минуты".

Но просто так сходить в спортзал, без соответствующего настроения, ради галочки, — это все равно, что при запоре зайти в туалет просто посидеть на унитазах, ничего путного не сделав. Толку никакого, а нервов можно много извести. Я немало лет был тесно связан с различными научно-исследовательскими институтами, и не уставал поражаться одной вещи — там все, ну или почти все, друг другу жалуются на свое все время ухудшающееся здоровье. Причем некоторые даже прикладывают силы к изменению ситуации — и массажик, там, и сауна. Но — нет здоровья и все тут. Потом понял, в чем дело. Не хотят выздоравливать. Вернее, очень хотят, из кожи вон лезут, только бы... Не выздороветь. Потому что либо неудачники, либо кандидаты в неудачники. Потому что как-то надо оправдываться за нищенскую зарплату, за скотское отношение, за не так прожитую жизнь... А чем оправдываться-то? А тем, что болен, поэтому и не смог. Не смог пойти поговорить с вышестоящим начальством, потому что после общения с ним язва обостряется. На полтора месяца. Не смог съездить в Москву, потому что уже десять лет не может ни на чем ездить без страха получить инфаркт в дороге вдали от скорой медицинской помощи. Не смог написать грант, потому что язва не дала, которая плевать хотела на высокое начальство и все равно сама по себе обострилась и не дала написать грант. Доходит до абсурда — люди начинают культивировать свои болезни: с удовольствием рассказывают друг другу о своих болезнях, чуть ли не гордятся тем, у кого круче заболевание. И, конечно, болеют. Потому что им не надо выздоравливать. Не для чего. Потому что когда-то они выбрали для себя путь этакого страдальца и неукоснительно ему следуют.

Примечание

Потому что если надо — то люди выздоравливают. Потому что когда очень надо — даже рак отступает. Даже в последней стадии. Не верите — почитайте журнал "Зож", там много таких примеров. Перескажу один из них. Женщина, болевшая раком в 4 стадии, признанная неоперабельной, сумела выздороветь. Сама. Потому что ей было очень надо: у нее была дочь-восьмиклассница, которую надо было поднимать на ноги и которая после ее смерти оставалась бы сиротой.

А те здоровые, которые попадают в эту клоаку немощных невротиков, тоже со временем заболевают. Даже если не хотят. Потому что — внушение. Потому что нельзя быть белой вороной, если все вокруг черные. Перекрасят на свой лад, обязательно перекрасят. Ах, ты здоровый? Тогда иди вон то сделай, а потом еще вот это... А мы почему не можем? Так мы ж больные, как тебе не стыдно!

Примечание

Конечно, не так прямо. Но в общем — именно так. Путем жалоб, стонов, охов и ахов, чтоб вы сами поняли, какой грех на душу возьмете, если будете просить вам помочь почти умирающего человека.

И вот вы уже, сначала подсознательно, тоже начинаете "заболевать". Как-то раз сказали "Ой, что-то у меня вот здесь болит". И — как разом все изменилось. Вас сразу перестали заставлять все делать, вернее даже запрещают: "Ты что, с ума сошел? В таком-то состоянии. У тебя же, судя по всему, гастрит (язвенная болезнь, остеохондроз и т. д.) обострился". Чайку предлагают, советы дают на перебой как излечиться... И постепенно вы "улетаете". Сначала начинаете незаметно для себя сгорбливаться при входе в родное заведение, потом не распрямляетесь и выходя из него... И тоже начинаете играть в болезнь. Поэтому: если видите, что в вашем коллективе имеет место быть "культ нездоровья", бегите оттуда. Остаться — себе дороже. А если нет возможности убежать, то хотя бы старайтесь проводить там как можно меньше времени. А также бегите подальше и побыстрее от тех "героев", которые изо дня в день, из года в год, "через не могу" прикладывают героические усилия, чтобы прийти на работу, а потом еще более героические, чтобы что-то на ней сделать. Вместо того, чтобы заняться своим здоровьем и вылечиться. А если не можете убежать, то хотя бы не слушайте тот бред, который они несут: "Вот я какой — у меня пятый день сердце болит, а я хожу на работу, а ты чуть-чуть загрипповал и на бюллетень". Затыкайте уши и не слушайте. Мне врачи из кардиоцентра рассказывали случай, как один высокопоставленный сердечник, несмотря на все протесты врачей, вместо того, чтобы лечиться, устраивал у себя в палате совещания. И разносы устраивал. И все время кому-то звонил. И почти перед самой операцией собрал у себя в палате форменное собрание. Со звонками по сотовому телефону, последними наставлениями и т. д. Наставления и впрямь оказались последними, потому что помер он, этот высокопоставленный сердечник. Особенно восхитительно, что друзья и близкие наперебой друг другу говорили: "Вот ведь какой герой — одной ногой стоя в могиле все равно продолжал работать". Герой... Вместо того, чтобы все силы на вытаскивание этой ноги бросить, он руками размахивал. И вполне закономерно потерял равновесие. Или, наоборот, обрел и теперь твердо стоит на обеих ногах. Только уже не на этом свете.

А вы грипп или ангину на ногах переносите, ходя на них, полусогнутых, на работу? Тоже зря. Да почему ж, спросите? Это ж грипп банальный, это ж не перед операцией... Так вот знайте, что этот перенесенный на ногах грипп может привести к различным страшным болезням сердца, к примеру, дилатационной кардиомиопатии. Это когда все камеры вашего сердца диффузно расширяются (во все стороны) и при этом ослабляется систолическая функция сердца. Попросту говоря, разрастается сердце как гриб трухлявый, а кровь качает все хуже и хуже... Кардинально спасти от этой болезни может только пересадка сердца. Которую производят далеко не всем и не часто, а в нашей стране в связи с известными гонениями на трансплантологов операции вообще сведены к минимуму. Не проводятся то есть. Понимаете? Гриппуете вы, и чтобы не подвести команду и закончить в срок проект, недельку переносили грипп на ногах. На работе. А потом через некоторое время чувствуете, что что-то не то: одышка, слабость, затем боли в сердце. Идете к врачу, а он вам и говорит, что жизни осталось 5—6 месяцев в лучшем случае. Недельку-то сэкономили, а жизнь потеряли.

Примечание

Я видел двадцатитрехлетнего человека, заболевшего в результате перенесенного на ногах гриппа этой болезнью. Честно скажу — просто страшно. Не то страшно, что видел, а то, как за какие-то пустяки можно расплатиться жизнью. И когда я, болея гриппом или ангиной, вдруг захочу "прогуляться" до работы на своих полусогнутых, обливаясь потом, я вспоминаю того человека. И желание совершить эту глупость быстро пропадает.

Поэтому бросьте все отговорки и займитесь собой. Своим телом. Своим здоровьем. И — в спортзал, на прогулку, в поход... Только одно маленькое замечание: по себе знаю, поначалу ох как не хочется. Никакого настроения нет. Но — без настроения нельзя. Еще раз повторяю: при занятиях спортом обязателен положительный настрой. А если его нет? Значит, надо стать немного актером и на первых порах создавать этот самый настрой искусственно. И до, и, обязательно (!) после занятий. Иначе ничего не получится: так многие бросали заниматься. И — гуляйте. На свежем воздухе. Такие прогулки воздадутся сторицей. Ведь очевидно, что когда через мозги протекает обогащенная кислородом кровь, мозги соображают лучше. А когда через них течет нечто типа канализации, обогащенной винцом, никотинчиком и окисью углерода, то и результат соответствующий: ни одна помойка еще ничего стоящего не произвела.

...После ряда случаев я понял, что случись, не дай бог, какая блокада, наше поколение ее не переживет. Хотя бы потому, что скончается от инсульта сразу после того, как узнает, что она началась. А случаи следующие. Первый про инсульт. Двадцатисемилетний (!) преуспевающий менеджер мгновенно умер от инсульта (!) после того, как ему сообщили по телефону, что договор,

подписания которого он ждал, не подписали. Телефонная трубка так и осталась у него в руке... А второй — про инфаркт. Моя знакомая, врач-реаниматолог кардиологической бригады "скорой помощи" рассказывала, что однажды приехала на вызов к двадцатичетырехлетнему (!) парню, у которого был самый натуральный инфаркт миокарда. И еще пара случаев из собственной практики оказания первой медицинской помощи. Один раз на моих глазах случился гипертонический криз у двадцатипятилетнего парня (до инсульта дело, к счастью, не дошло), а другой раз видел, как на улице гулявший с девушкой паренек скорчился от приступа стенокардии, как потом выяснилось. Как говорится, куда дальше. Случаи про инсульты и инфаркты у тех, кому около 30 или чуть за 30, рассказывать не буду, так как они, увы, стали почти обыденностью. При этом я здесь специально не рассказываю про аналогичные случаи с девушками, чтобы показать, что даже сильный пол стал, скажем так, не очень сильным. Вот так. Какие уж тут блокады... Какие подвиги... Если сейчас уже некоторым из тех, кому за 20, грозит только одна блокада — сердечная. В случае если лифт сломается и придется пешком на третий этаж лезть.

Поэтому в который раз повторяюсь, — занимайтесь спортом, ходите в походы, гуляйте наконец. Если хотите, чтобы ваша жизнь напоминала полнуюводную реку, а не заросшее тиной мелководье.

Немного о виктимологии

Виктимология — это наука о жертвах. То есть о людях, которым все время падает кирпич на голову, даже если они идут в чистом поле. Помните, у Чехова в пьесе "Вишневый сад" есть такой персонаж Епиходов, которого все звали "тридцать три несчастья"? Вот именно таких людей и почему они стали такими, и изучает виктимология. Тема эта очень обширная даже для книги, а не только для этого маленького раздела. По своей сути вся практическая психология ставит своей целью помочь человеку перестать быть жертвой. Здесь я лишь повторю мысль, многократно высказанную в этой главе. Работайте на успех, а не на неудачу! Помните, что вы — человек успеха, а не неудачник. А от последних бегите как можно дальше, какими бы прекрасными людьми они не были. Потому что помощь жертвам это удел высококвалифицированных психологов-профессионалов, а не наш с вами. А быть втянутым в орбиту жертвы очень просто. Поэтому, когда идете на встречу с заказчиком, создавайте в себе максимально позитивный настрой. Для этого, кстати, непосредственно перед встречей не мешает вспомнить что-нибудь приятное. Если же вы идете с ощущением собственного ничтожества перед вашими конкурентами, не удивляйтесь, что вам ничего не закажут. Помните, что "если назвался червяком, не обижайся на то, что тебя раздавят".

Примечание

Небольшой тест на виктимность. Возьмите нескольких людей и выкопайте яму. Неглубокую. А если хотите глубокую, только в этом случае жестко натяните сетку какую-нибудь, потому что некоторые потом в эту яму будут падать. Потом отведите испытуемых на некоторое расстояние от ямы (по прямой), завяжите им глаза и скажите, чтобы шли вперед. К яме. Суть в том, что нужно с завязанными глазами в эту яму не провалиться: остановиться сразу перед ней. А теперь смотрим, что получается. Первый пошел. Два шага сделал резво, а потом до самой ямы плелся, что твоя черепаха Тортилла. Но не упал. Подошел к самому краю, сделал шаг вперед и почувствовал, что дальше хода нет. Второй пошел. Бегом бежит! И — ооп-па! — касается одной ногой самого края ямы и ее перепрыгивает! Да... За этого можно не беспокоиться, у него в жизни получится все и даже больше. Третий — идет ровно, спокойно, но прямо у самого края встал, подумал и дальше не пошел. С этим тоже дальше все будет нормально. А вот четвертый... Четвертый шел бодро, уверенно, с улыбкой и упал в яму. Этого можно подозревать на виктимность. Второй раз, третий, четвертый, пятый — и все время падает. Все, почти точно — это виктим. Если виктим сто-процентный, так скажем, то можно наблюдать почти мистические вещи. Можно отвести его в сторону от ямы, и сказать, чтобы ничего не боялся и шел прямо. Потому что яма в стороне и ничего ему не грозит. Ба! А прямо-то ему не идет-ся, яма эта притягивает его как магнитом, и вот он заворачивает, заворачивает, и... И снова падает в яму. Хотя, казалось бы, шанс попасть в нее в данном случае ничтожный. И однако... В жизни этот человек будет делать примерно то же самое: из тысячи возможных решений он выберет самое неудачное. А на все удачные моменты жизни, которые ему подсказывает судьба, он будет плевать с высокой колокольни и попросту не замечать их. Дожидаясь того единственного момента, который самый плохой. Дожидаясь этой своей ямы... Но кроме четвертого есть еще и пятый. Мистика, но если отвести его вбок от ямы, то он все равно приходит к ней! Тест этот проводить вам совершенно необязательно, просто на его примере я попытался лишний раз объяснить, кто такие жертвы. Или виктимы, как мы условились их называть. Вот и в жизни такие люди делают то же самое.

А теперь несколько рекомендаций на предмет того, как не стать виктимом. Не стать жертвой, не стать неудачником.

Во-первых, обязательно верьте в себя, если хотите, чтобы что-то получилось. Вернее, не что-то, а именно то, что нужно. Вам нужно, а не кому-то другому. Вера в себя — штука очень важная. Если бы я, к примеру, не верил в себя, когда решил написать свою первую книгу ("Самоучитель РНР 5" в соавторстве с И. Симдяновым), то я никогда бы ее не написал. Потому что из тех, кому я сообщил о своем решении (опрометчиво, как я теперь понимаю), почти никто его не поддержал. Вернее, "не поддержал" — это мягко сказано. Я в то время был научным работником, и очень многие коллеги даже не пытались скрывать своего отрицательного отношения по поводу этой задумки. В лучшем случае мягко советовали, чтобы я все это бросил, потому что "не получится и только опозорюсь", в худшем — говорили, что я "высочка и вообще очень плохой и ненадежный человек" (какое все это

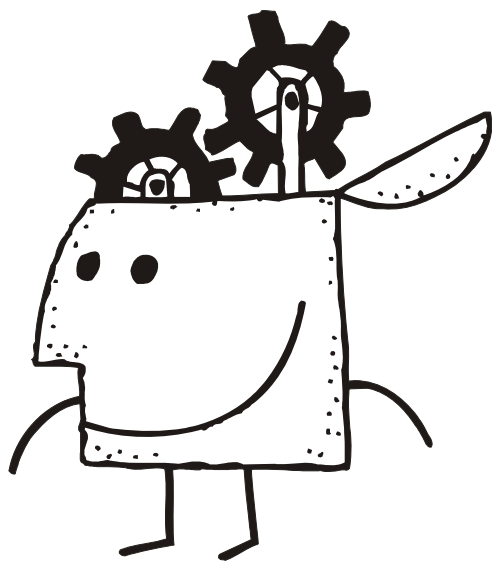
имело отношение к книге, я до сих пор слабо понимаю). Даже непосредственный мой научный руководитель, с которым вместе не одна научная статья была написана, и с которым были в неплохих отношениях, и тот "отлучил от дома" и обозвал выскочками. Он, в общем-то, хороший человек, но из песни слова не выкинешь — что было, то было. А я просто верил в то, что у нас это получится, и не верил всем остальным. Время показало, что я был прав. Поэтому, когда вы задумаете сделать что-то стоящее, не удивляйтесь тому, что вас, скажем так, "не поймут" и вместо ожидаемого лаврового венка повесят на шею какой-то обидный ярлык (чудак, выскочка и т. д.). Так бывает очень часто, и через это прошли очень многие. Потом будут и лавры, будут и все прочие атрибуты успеха, если только не перестанете верить себе и делать то, что решили. Помните, что "чудак и выскочка" от человека, добившегося успеха, отличается только тем, что первый успеха еще пока не добился.

Во-вторых, относитесь к жизни позитивно. Будьте оптимистом. Не идеалистом, а именно оптимистом. У Саши Черного есть стихи про оптимистов и пессимистов, в которых есть такие строки: "Как для коллеги-пессимиста земля — коллекция гробов". И если не хотите, чтобы ваша жизнь напоминала "коллекцию гробов", то нужно относиться к ней позитивно. В жизни есть и белое, и черное. Какой будет ваша жизнь, решать только вам — если будете на нее смотреть сквозь черную призму, то она и будет черной. Стопроцентно. "Как корабль назовешь, так он поплывет" — пословица, верная во всех отношениях. Давно известно, что плохое наиболее часто случается с теми, кто этого ждет, сознательно или подсознательно. Наверняка вы не раз слышали, как про какого-то человека говорят "словно заколдованный". Один "словно заколдованный" потому, что его бьют каждый раз, стоит ему только показаться на улицу, другого считает за честь любая встречная собака укусить, включая престарелых болонок-пацифисток, у третьего все время финансовые проблемы, потому что он умудряется быть должен даже тогда, когда в долг берут у него, и т. д. Я не знаю, почему так, но неприятности действительно имеют свойство притягиваться к тем, кто к ним предрасположен. И видя такие случаи, понимаешь, что ничего удивительного, в общем-то, и нет: посмотришь иной раз на человека, и видишь, что с ним просто не может не случиться какой-нибудь гадости. Она и случается. История знает мало примеров, когда озлобленные на жизнь люди добивались чего-то стоящего, а если и добивались, то удержать не могли и заканчивали очень плохо.

В-третьих, не надо бороться с препятствиями, надо их обходить. Этот принцип М. Литвак называет "принципом сперматозоида" (чуть позже мы поймем почему). И не стоит относиться к препятствиям как к вселенской трагедии — препятствия тоже нужны. И как на биологическом, так и на социальном уровне они выполняют одну и ту же функцию: указывают нам правильный путь. И если препятствий не будет вообще — тоже ничего хо-

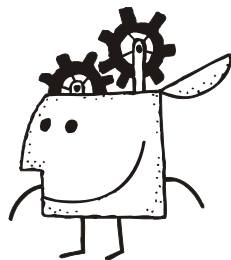
рошего. Вспомним, с чего начиналась наша жизнь. С того, что мы были сперматозоидами и стремились к яйцеклетке. А в трубах матки есть клетки реснитчатого эпителия, которые двигаются против движения сперматозоида. Мешая ему, с одной стороны, поскорее добраться до яйцеклетки, но, с другой стороны, указывая путь к ней. Наверное, тот сперматозоид, которым был каждый из нас, тоже мечтал, чтобы этих "клеток-препятствий" не было. Но они были, к счастью. К счастью, потому, что если в результате болезни у женщины эти клетки утрачивают способность к движению, то она становится бесплодной. То есть ни один сперматозоид до яйцеклетки не дойдет. А раз эти "клетки-препятствия" имелись, то у нас был шанс дойти до яйцеклетки и потом появиться на свет. Но шанс этот был выигрышным только в том случае, если мы не боролись с препятствиями, а обходили их (это по Литваку и называется принципом сперматозоида: не надо бороться с препятствиями, а надо их обходить). И в жизни также: если все разрешено (нет препятствий-ограничителей), то человек становится "бесплодным", в смысле, не дойдет до намеченной цели. Примеры можно приводить бесконечно. Начиная от банального дорожного бордюра и заканчивая, скажем, работой руководителя. Приведу пример с одним своим бывшим сотрудником. Очень талантливым. Которому я разрешил за его талант и хорошее ко мне отношение практически все: и работать по свободному графику, и спорить со мной без повода, и многое другое... Разрешил. И все испортил. Этому сотруднику, в первую очередь. Потому что, работая по свободному графику, он практически перестал работать вообще, занимаясь чем ни попадя, а споры потом переросли в откровенное хамство. То есть человек стал бесплодным.

В-четвертых, никогда не нойте. Даже если кажется, что есть повод. "Кажется" — потому, что при ближайшем рассмотрении оказывается, что повода-то и нет. А так, ерунда всякая. Но даже если есть, — все равно не нойте, не ругайтесь на жизнь, не проклинаяте ее. У Ю. Визбора есть хорошие стихи со словами "Кручина твоя — не причина, а только ступень для тебя" (кручина — это печаль). И это так. Если воспринимать "свои кручины" именно как ступень — все становится намного проще. А если с утра до вечера думать "Ну за что мне такое досталось", — есть очень большой шанс превратиться в 100%-ного виктима, которому потом достанется еще и все, что не досталось ранее.



ПРИЛОЖЕНИЯ

Приложение 1



Договорные отношения

В этом разделе на примере договора на создание сайта мы рассмотрим основные принципы заключения тех договоров, с которыми наиболее часто приходится работать сотрудникам ИТ-сферы. Также поговорим об основных принципах формирования технического задания. Комментарии по поводу некоторых тонкостей, содержащихся в договорах, будут даны в примечаниях. К сожалению, многие не уделяют должного внимания "бумажкам" и нередко относятся к составлению договоров достаточно невнимательно. Увы, жизнь показала, что так лучше не делать. За документами надо следить: не составленный вовремя документ или неверная фраза в нем могут лишить вас очень многого, если не всего. Примеров этому очень много: не один человек лишился своего бизнеса из-за неправильно составленного договора. Более того, некоторые люди и организации специально занимаются тем, что подлавливают "лохов от юриспруденции" и отсуживают у них немалые суммы, зачастую полностью разоряя. В ИТ-сфере я, правда, таких примеров еще не встречал, но, думаю, дело не за горами. Но примеров того, как из-за неаккуратно составленных бумажек закрывались ИТ-компании, видел немало. При этом, как вы понимаете, высокий уровень программистского мастерства сотрудников, хорошие человеческие качества руководителя и проч. никого не интересовали... Интересовала только неверно написанная фраза, под которой подписался руководитель. Поэтому, если есть возможность, консультируйтесь с юристами, желательно с родственниками или хорошими знакомыми, которые уберегут вас от многих подводных камней. Если такой возможности нет, значит, надо все изучать самому: иного выхода нет. Причем изучать так же, как вы изучаете программирование: основательно. Так изучать, чтобы ту часть юриспруденции, которая непосредственно вас касается, вы знали не хуже юристов.

Договор на создание сайта

г. Нижний Новгород

" _____ " 2006 г.

Примечание

Это — типовая "шапка" договора: сначала название договора, потом, строчкой ниже, — место и дата заключения. Не забывайте писать дату заключения договора: это очень частая оплошность, когда место под дату есть, а сама дата не вписана. А если нет даты, то многие пункты договора просто теряют свой смысл, поскольку сроки выполнения работ по договору отсчитываются с момента его заключения.

ООО "..." в лице директора Петрова П. П., действующего на основании Устава (именуемого в дальнейшем Исполнитель), с одной стороны, и ООО "..." (именуемого в дальнейшем Заказчик), в лице директора Иванова И. И., действующего на основании Устава, с другой стороны, заключили настоящий Договор о нижеследующем:

1. Предмет Договора

1.1. Заказчик поручает, а Исполнитель принимает на себя обязанности по разработке корпоративного сайта (в дальнейшем — Сайт) Заказчика.

Примечание

Конечно, это фраза стандартная: вы можете писать не "разработка сайта", а "разработка дизайна сайта", "разработка программных блоков сайта в составе (далее — перечисление блоков)" и т. д.

2. Обязательства сторон

2.1. Исполнитель обязуется оказывать услуги в соответствии с календарным планом выполняемых работ (Приложение № 1 к настоящему Договору), который является неотъемлемой частью настоящего Договора.

2.2. Исполнитель обязуется осуществлять гарантийную поддержку разработанного им сайта в течение __ месяцев после сдачи-приемки работ. Гарантийная поддержка включает в себя устранение неисправностей, ошибок и недочетов в разработанном сайте, не выявленных до сдачи-приемки работ. Гарантийная поддержка автоматически прекращается после несанкционированного вмешательства в коды сайта иных лиц, кроме представителей Исполнителя.

Примечание

Пункт о гарантийной поддержке очень скользкий. Будь моя воля, я бы его вообще не включал, но клиенты почти всегда смотрят на то, есть ли в договоре пункт о гарантийной поддержке. А не включал бы я его не потому, что не хочу гарантийную поддержку оказывать, а потому, что под этим каждый понимает что-то свое, и в случае возникновения споров разобраться кто прав, а кто нет, практически невозможно. Поэтому, если вы просто пишете, что "осуществляете гарантийную поддержку", то будьте готовы, в случае чего, к выяснениям того, что под этим понимает клиент. А он может понимать все что угодно: изменение строчки кода для него может тоже быть гарантийной поддержкой, а то, что вам из-за этой строчки придется весь код перелопатить, его не волнует. Поэтому обязательно пишите фразу "Гарантийная поддержка включает в себя устранение неисправностей, ошибок и недочетов в разработанном сайте, не выявленных до сдачи-приемки работ". Этой фразой вы конкретизируете то, что понимается под гарантийной поддержкой. А следующей фразой вы говорите о том, что проблемы чужой некомпетентности — это не ваши проблемы.

2.3. Исполнитель не несет ответственности за работоспособность сервера, на котором будет располагаться Сайт Заказчика.

Примечание

Обязательная фраза, которая избавит вас от многих проблем. Для большей части клиентов понятия "сервер" не существует, и если у них не "показывается сайт в Интернете", то виноват тот, кто этот сайт делал. Увы, точка зрения очень распространенная.

2.4. Заказчик обязуется оплатить оказанные ему услуги в сроки и в порядке, которые указаны в п. 4 настоящего Договора.

2.5. Заказчик обязуется предоставлять материалы, необходимые для создания Представительства в течение 2 (двух) рабочих дней с момента их запроса Исполнителем. В случае задержки Заказчиком предоставления вышеуказанных материалов срок действия Договора автоматически продлевается на количество рабочих дней, равное времени задержки.

Примечание

Эту фразу тоже советую писать. Дело в том, что клиентам "как всегда" надо все быстро и сразу, и нередко они сами пишут очень малые сроки исполнения договора, и сами же в эти сроки не укладываются, когда от клиента требуется большой объем информации по наполнению сайта. Правда, для того, чтобы срок договора юридически грамотно пролонгировать, вы должны запросить информацию у клиента официально, направив официальный запрос. И после этого, если ответа не последовало, нужно направить официальное уведомление о пролонгации договора, в котором можно написать примерно следующее: "В связи с тем, что запрошенная нами __ числа информация к нам не поступи-

ла, согласно пункту 2.5 Договора №__ на создание сайта от ____ 2006 г. срок действия договора пролонгируется на количество рабочих дней, равное времени задержки".

2.6. Заказчик вправе ознакомиться у Исполнителя с ходом работ на любом этапе выполнения работ, не вмешиваясь в его деятельность.

2.7. Исполнитель обязуется не предоставлять третьим лицам материалы, составляющие предмет коммерческой тайны Заказчика.

2.8. Заказчик обязуется не предоставлять третьим лицам материалы, составляющие предмет коммерческой тайны Исполнителя.

3. Порядок выполнения и сдачи-приемки работ

3.1. Исполнитель начинает выполнение работ по настоящему Договору с момента подписания Договора.

3.2. Заказчик и Исполнитель, по взаимному соглашению, могут изменять очередность выполнения этапов работ, перечисленных в календарном плане работ (Приложение № __ к настоящему договору).

3.3. По завершении работ по настоящему Договору Исполнитель предоставляет Заказчику результаты выполненных работ и акт сдачи-приемки, который Заказчик утверждает в течение двух дней с момента его получения или дает мотивированный отказ от приемки работ.

Примечание

Пишите количество дней, в течение которого заказчик должен утвердить акт приемки-сдачи работ, иначе он может утверждать его очень долго.

3.4. В случае мотивированного отказа Заказчика от приемки работ сторонами составляется двусторонний акт с указанием необходимых доработок и сроков их выполнения.

4. Стоимость и порядок оплаты работ

4.1. Стоимость работ, поручаемых Исполнителю по настоящему Договору, определяется в соответствии с установленными Исполнителем тарифами, и календарным планом работ, подписанных Исполнителем и Заказчиком, и прилагаемыми к настоящему Договору.

4.2. Заказчик в течение __ рабочих дней с момента подписания акта сдачи-приемки этапа работ перечисляет на расчетный счет Исполнителя стоимость выполнения этапа работ в размере 100 (сто) процентов от стоимости этапа работ.

Примечание

Количество дней, в течение которых вы ждете на вашем счете сумму работ по договору, тоже не забывайте писать, иначе, как и в случае с актом приемки-сдачи работ, это может быть достаточно долго.

Примечание

Фраза "стоимость выполнения этапа работ в размере 100 (сто) процентов от стоимости этапа работ", конечно, условна и может меняться в зависимости от конкретной ситуации. Вы можете работать с предоплатой, тогда в этом пункте указываете размер предоплаты, можете брать оплату не поэтапно, а сразу за весь договор и т. д.

5. Ответственность сторон

5.1. За неисполнение или ненадлежащее исполнение своих обязательств по настоящему Договору стороны несут ответственность в соответствии с действующим законодательством Российской Федерации.

5.2. При просрочке Исполнителем срока выполнения работ по настоящему Договору Исполнитель выплачивает Заказчику неустойку в размере 0,1% стоимости оплаченных и невыполненных работ за каждый день просрочки. Выплата неустойки не освобождает Исполнителя от выполнения работ по настоящему Договору.

Примечание

Вы, в принципе, можете поставить любой процент неустойки — это дело ваше и вашего клиента. Обычно процент неустойки рассчитывается как 1% от **ставки рефинансирования ЦБ РФ**. То есть, к примеру, если сейчас ставка рефинансирования составляет 18%, то, значит, размер неустойки будет равен 0,18% от суммы договора.

5.3. При просрочке Заказчиком оплаты работ в соответствии с п. 4.2 настоящего Договора Заказчик выплачивает Исполнителю неустойку в размере 0,1% от невыплаченной в срок суммы за каждый день просрочки. Выплата неустойки не освобождает Заказчика от обязанностей оплаты стоимости выполненных работ.

5.4. Исполнитель не несет юридической, материальной или иной ответственности за содержание, качество и соответствие действующему законодательству информации, размещенной Заказчиком на Сайте.

Примечание

Эту фразу тоже советую писать: никогда точно не известно, какую информацию будет размещать клиент на своем сайте (хотя бы потому, что иногда приходит-

ся разрабатывать только "движок" и ничего при этом не знать о планируемом информационном наполнении). А отвечать за то, что вами не делалось, вряд ли стоит.

5.5. Исполнитель не несет ответственности по претензиям Заказчика к качеству соединения с сетью Интернет, связанным с качеством функционирования сетей интернет-провайдеров, с функционированием оборудования и программного обеспечения Заказчика и другими обстоятельствами, находящимися вне компетенции Исполнителя.

Примечание

Тоже нужная фраза. Очень нередко, когда по каким-то причинам не работает сайт, во всем начинают винить разработчика. Дело в том, многие руководители не очень хорошо знакомы "с компьютерами", и если не работает сайт, то, значит, виноват тот, кто его делал: то, что нет Интернета или "валяется хостинг" — это их не волнует. В чем нет ничего удивительного — они не обязаны это знать. Поэтому, если вы при создании, скажем, сайта общаетесь не с профессионалами от IT, а с "гражданскими лицами", советую вам перед началом работы провести небольшой ликбез: объяснить, что за Интернет ответственен интернет-провайдер, за физическое размещение сайта в сети — хостинг-провайдер, что доменные имена покупаются не у вас, а у компаний-регистраторов и т. д.

5.6. Исполнитель не несет материальной или иной ответственности за возможные материальные или иные потери, которые может понести Заказчик, связанные с тем, что по тем или иным причинам Сайт не функционировал или функционировал неправильно.

Примечание

Этот пункт старайтесь писать везде, чтобы вас не оштрафовали на крупную сумму за какую-нибудь упущенную выгоду, которую понес Заказчик в силу того, что с сайтом/ПО что-то произошло. Не все клиенты хотят подписывать договор с таким пунктом, особенно в том случае, если вы разрабатываете какое-либо ПО, связанное с проведением финансовых операций. Но даже если вы обязаны нести ответственность за неработоспособное ПО, я вам настоятельно советую, чтобы не разориться в одночасье, сделать две вещи. Во-первых, четко прописать разграничение ответственности: что, к примеру, за код вы отвечаете, а за сбои в сети Интернет — нет. Также советую написать, что вы не отвечаете за ошибки в ПО, возникшие вследствие его неправильной эксплуатации (пускай потом доказывают в суде, что ни разу не эксплуатировали его неправильно). Во-вторых, нужно написать фразу "В случае наложения на Исполнителя штрафных санкций общая сумма всех штрафов по Договору не может превышать стоимости услуг, оказанных в рамках настоящего Договора" (этой фразой вы, по крайней мере, обезопасите себя от того, чтобы не потратить на штрафы все деньги, которые у вас имеются).

5.7. Споры, не урегулированные между сторонами по взаимному согласию, подлежат разрешению в судебном порядке.

6. Дополнительные условия

6.1. Авторские права на разработанный Сайт являются объектом смежных прав Исполнителя и Заказчика (при условии полной оплаты выполненной работы) на данный Сайт либо его элементы при тиражировании. Распространение или размещение Сайта или его элементов на других сайтах допускаются с обязательной ссылкой на субъектов смежных прав.

Примечание

Об авторских правах см. Приложение 2.

6.2. Исполнитель не несет ответственности за задержку предоставления результатов работ, связанную с возможными сбоями соединения с сетью Интернет и качеством функционирования сетей интернет-провайдеров.

7. Сроки действия Договора

7.1. Договор вступает в силу с момента его подписания Сторонами и действует до "___" _____ 2006 г., а в части финансовых обязательств — до полного их исполнения.

Примечание

Советую вам писать фразу "а в части финансовых обязательств — до полного их исполнения" — знаю несколько случаев (правда, не из мира программирования), когда после окончания договора заказчик отказался платить деньги за услугу на том основании, что договор закончился.

7.2. Срок выполнения работ по Договору: в течение ___ календарных дней с момента подписания Договора. Исполнитель имеет право выполнить работы досрочно.

8. Порядок расторжения Договора

8.1. Досрочное расторжение Договора может иметь место по соглашению сторон либо по основаниям, предусмотренным действующим на территории Российской Федерации гражданским законодательством.

8.2. При нарушении одной из Сторон условий настоящего Договора другая Сторона имеет право на одностороннее расторжение Договора, предварительно известив об этом другую Сторону не менее чем за пять рабочих дней до расторжения Договора.

8.3. Сторона, решившая расторгнуть Договор, уведомляет другую Сторону по факсу, указанному в п. 11.

8.4. В случае установления нецелесообразности или невозможности дальнейшего проведения работ или установления неизбежности получения отрицательного результата заинтересованная Сторона вносит предложение о досрочном расторжении настоящего Договора, которое должно быть рассмотрено в течение 2 (двух) рабочих дней.

9. Действие непреодолимой силы

Примечание

Раздел о форсмажорных обстоятельствах обязателен для любого грамотно составленного договора. Здесь он приведен в стандартном варианте. Если подходить к этому вопросу более основательно, то можно этот пункт расширить, добавив более реальные форсмажорные обстоятельства. К примеру, у вас на два дня в офисе отключили свет, и вы не смогли вовремя сдать работы. Это в принципе тоже форсмажорное обстоятельство. Естественно, оправдываясь за срыв работ таким образом, вы должны будете иметь на руках справки о том, что это действительно имело место быть.

9.1. Ни одна из Сторон не несет ответственности перед другой Стороной за задержку или невыполнение обязательств по настоящему Договору, обусловленных обстоятельствами, возникшими помимо воли и желания Сторон и которые нельзя предвидеть или избежать, включая объявленную или фактическую войну, гражданские волнения, эпидемии, блокаду, эмбарго, землетрясения, наводнения, пожары и другие стихийные бедствия.

9.2. Сторона, которая не исполняет своего обязательства, должна дать извещение другой Стороне о препятствии и его влиянии на исполнение обязательств по Договору.

10. Заключительные положения

10.1. Любые изменения и дополнения к настоящему Договору действительны лишь при условии, что они совершены в письменной форме и подписаны уполномоченными на то представителями Сторон. Приложения к настоящему Договору составляют его неотъемлемую часть.

10.2. Настоящий Договор составлен в двух экземплярах на русском языке. Оба экземпляра идентичны и имеют одинаковую силу. У каждой из Сторон находится один экземпляр настоящего Договора.

11. Реквизиты сторон

От Заказчика	От Исполнителя
/Реквизиты/ Генеральный директор _____/	/Реквизиты/ Генеральный директор _____/

Примечание

Основной текст договора закончен. Далее следуют приложения к договору. Обычно это календарный план работ по договору (Приложение 1) и техническое задание (Приложение 2).

Приложение № 1

к договору №____ от " __ " _____ 2006 г.

Календарный план работ по Договору

Примечание

Календарный план работ обычно оформляется в виде таблицы, приведенной ниже.

№ этапа работ	Вид работ	Срок исполнения с момента заключения договора, календарные дни	Стоимость, руб., в т. ч. НДС 18%
1.	Создание дизайна сайта	7	6000
2.	Создание программного блока "Каталог продукции"	12	10 000
Итого:			16 000

От Заказчика	От Исполнителя
/Реквизиты/ Генеральный директор _____/	/Реквизиты/ Генеральный директор _____/

Примечание

В конце каждого приложения обязательно идут реквизиты сторон.

Техническое задание

На техническом задании остановимся подробнее. Потому что это вещь крайне серьезная. Хотя бы потому, что ...не всегда надо техническое задание делать. Вернее, к оформлению технического задания нужно подходить очень ответственно не только с технической стороны, но и с психологической. Проблема в том, что когда вы начинаете разрабатывать сайт, вы не всегда знаете, чем все это закончится. И клиент не знает, особенно если он первый раз заказывает сайт и совершенно не представляет себе, как будет протекать работа. Здесь каждый раз нужно смотреть по ситуации.

Вопросы и ответы

В этом разделе приведены наиболее частые вопросы, задаваемые нам посетителями сайта softtime.ru по теме заключения договоров, и наши ответы на них.

Примечание

Авторские названия ников сохранены. Вопросы участников форума выделены курсивом. Там, где автор ответа не приводится, значит, отвечаю я, М. Кузнецов. В других случаях пишется автор ответа (И. Симдянов или С. Голышев).

Вопрос:

Какое должно быть ТЗ? С чем надо идти к заказчику? О чем не надо спрашивать и что надо спросить обязательно?

Ответ:

ТЗ в первую очередь должно быть удобным. И для Вас и для Заказчика. Для Вас — в том, что Вы должны быть гарантированы от дополнительной разработки ресурсоемких вещей, не оговоренных при заключении договора. Но если в ТЗ прописать все и вся, то это будет очень неудобно для Заказчика. Он Вас попросит шрифт там где-нибудь поменять или сделать некоторые изменения в дизайне, а Вы ему: "Нет, дорогой. В ТЗ написано, что шрифт должен быть такой-то и такой-то. И все тут". Такого, конечно, следует избегать. Сложность составления ТЗ в том, что зачастую Заказчик не знает, что ему нужно до тех пор, пока не увидит какие-то результаты работы и не поработает вместе с Исполнителем над созданием сайта. И тут ничего не поделаешь — объективная реальность. Только тогда он уже "в теме" и готов к

составлению ТЗ. И вот выяснить, что именно хочет Заказчик до того, как он стал "в теме", и составляет основную трудность на переговорах. Кроме того, не со всяким Заказчиком можно сделать ТЗ. Иногда приходится работать просто "с колес", полагаясь сугубо на собственный опыт и интуицию. Вот, к примеру, звонят нам с завода им. Свердлова и говорят: "Ребята, к нам тут высокая правительственная делегация приезжает. Срочно нужно сделать видеоролик. Минут на 8—10". "А когда приезжает?" — спрашиваем. "Через четыре дня". Разговор происходит в четверг, а делегация приезжает в понедельник. А видеоролик сделать с нуля на оборонную тематику для оборонного завода — это та еще песня. Хотя бы с многочисленными согласованиями, когда он уже готов. Да и чисто технически много чего: провести фото- и видеосъемку (с влезанием на 40-метровые ректификационные колонны), написать текст, утвердить его, записать дикторский голос (то еще удовольствие, даже если звукозаписывающая аппаратура нормальная), подобрать музыкальную тематику, которая будет идти фоном, свести воедино весь звук, сделать видео, и свести затем все это воедино. И все это за четыре дня, из которых два выходных. А 10 минут "полезного времени" — это немало. ...Какое уж тут ТЗ. Еще пример. Монтируем супер-пупер мини-АТС на 640 номеров на строящейся супер-пупер турбазе. Составляют люди проект, в том числе на предмет того, как линии будут в здании проходить. Здесь по стене, там по потолку и т. д. :) Составили, приезжают на эту турбазу. И понимают, что чего-то они в этом мире не понимают. А чего именно — пока еще не понимают. Смотрят в проект и видят, что не так: стены нет! Которая была еще неделю назад. По которой планировали провода тянуть. А за то время, пока проект составляли, была произведена перепланировка. Вот и составляй тут ТЗ...

Двойственно я отношусь к ТЗ. Грамотное ТЗ можно составить только в том случае, если от Заказчика в его составлении принимает участие специалист, понимающий, о чем идет речь. А это бывает не часто. Когда бывает — хорошо. А в некоторых случаях от этих ТЗ никакого проку нет. Только потеря времени и нервов. И тут всегда надо смотреть, что легче — составить ТЗ или его не составлять. Если это серьезный проект, этак на полгода-год с серьезной ответственностью и серьезными деньгами, то ТЗ составлять надо. Но уж тогда надо составлять его серьезно. И вместе с Заказчиком — объясняя ему что, как и почему. И на составление такого ТЗ может затратиться четверть всего времени, что ушло на проект. Вот тогда это будет реальное ТЗ, по которому можно реально работать. А если речь идет о каком-то сайте на \$300... Да зачем оно мне, это ТЗ нужно? Я его просоставляю дольше. Проще обойтись обыкновенными техническими требованиями, когда просто в плане работ в договоре перечисляем, что нужно сделать... В общем, по ситуации всегда надо смотреть.

>С чем надо идти к заказчику? О чем не надо спрашивать и что надо спросить обязательно?

Здесь от Заказчика много зависит — от того, что он из себя представляет. Точно могу сказать, что если вы едете на переговоры к какому-нибудь главе администрации, — с ним о ТЗ говорить ни в коем случае не надо. В лучшем случае — с теми техническими спецами, которым он Вас передоверит. А вот если он сам берет "дело на контроль" и говорит, что все, что касается сайта, обсуждать только лично с ним (бывает с главами такое), то о ТЗ вообще ни с кем говорить не надо. Он скажет: "Мужики, ну вот сделайте мне так, чтобы душа сначала свернулась, а потом развернулась! Ну, чтоб классно было". А Вы ответите: "Сделаем, т-щ Глава!". Вот и все Ваше ТЗ :) Я крамолу, может быть, скажу, но если Вы ему как человек понравитесь, то и то, что Вы сделали, ему понравится. Вот и все. А вот если сайт Вам заказал человек, который сам неплохо программирует на PHP или еще на чем-нибудь, то с ним ТЗ вполне можно посоставлять и технические вопросы самые разные загодя обсудить. Но если Заказчик технически не грамотен... Понимаете, какая штука. Вот приходит ко мне, скажем, водопроводчик, которого я вызвал, чтобы он мне трубы сменил. А я в сантехнике разбираюсь как попугай в металлургии. И если меня этот сантехник попросит ему какой-то проект предоставить, я его просто выгоню. И найду тех, кто мне и сделает хорошо, и дураком меня выставлять не будет, заставляя отвечать на вопросы, в которых я нисколько не смыслю. Вот и с ТЗ надо так же... Чтоб хуже не вышло.

Ответ (И. Симдянов):

В больших проектах сначала составляют требования, потом проектную документацию. По мере продвижения проекта меняются требования заказчика — считается удачей, если требования не изменяются более чем на 25%, так как изменения в середине и конце проекта часто требуют очень серьезных изменений в архитектуре и очень дороги. Кроме того, такие изменения практически невозможно осуществлять без ущерба качеству кода. По оценкам исследователей, от 50 до 80% всех усилий и затрат в больших проектах уходит на исправление требований, вносимых после начала кодирования.

Примечание

Последнее предложение в ответе И. Симдянова — очень важный довод в пользу того, что составлять ТЗ нужно. Но... Составлять его надо, как уже говорилось выше, только с компетентным в данных вопросах человеком, — иначе от этого ТЗ просто не будет никакой пользы.

Ответ (С. Голышев):

ТЗ — это что-то вроде анкеты, которую должен заполнить Заказчик перед оформлением заказа. При составлении ТЗ надо исходить из того, что Заказчик не разбирается в сайтах вообще. А такое случается сплошь и рядом — и это нормально. ТЗ затрагивает очень много вопросов, и чтобы заполнить

такое ТЗ Заказчик должен потратить свое время. Что не совсем правильно. Заказчик тратит свои деньги, а время должен тратить Исполнитель. И на большинство вопросов должен ответить не Заказчик, а разработчик, предварительно выяснив это у Заказчика.

Вопрос:

Договора и т. п. необходимо оформлять при создании сайтов на заказ?

Ответ:

Лучше оформлять. Хотя, частники, как правило, этого не делают. Но если вы представляете студию, а вместе с вами работают еще хотя бы два человека, то имеет смысл писать договор. Потому что риск, что заказчик не выплатит деньги, есть всегда. Правда, должен сказать, что сейчас серьезные организации сами будут за то, чтобы составить договор, и вряд ли пойдут на бездоговорные отношения. Если это, конечно, не скрипт за \$50, который им может быть проще оплатить наличными.

Но договор, во-первых, не панацея от невыплат. А во-вторых, договора надо составлять юридически грамотно. Иначе их лучше вообще не составлять — хуже будет. Потому что, прописав в договоре "не ту фразу", вы рискуете потерять больше, чем если вам просто не заплатят за скрипт. Таких случаев много. Причем некоторые фирмы даже специализируются на подобных "подставах", суть которых в том, что в фирме есть очень грамотные юристы, которые вам такого в договоре напишут и так вас убедят, что это единственно верная форма данного договора, что вы им только благодарны будете. А потом — арбитраж. И обдирают незадачливого договорщика как липку.

По поводу невыплат. Реальная ситуация такова, что сколько бы договоров не писалось, все друг другу выплаты задерживают. Редко, когда вовремя все получишь. И ничего страшного тут нет. Работа есть работа. Другое дело, опять же, что если договор написан просто для проформы, и вы сами не знаете, что в нем написано, — лучше его не писать, а работать на "честном слове". Надежней выйдет. Потому что, к примеру, по очень многим договорам на создание сайтов, которые я видел в Интернете, я как заказчик могу со спокойной совестью ничего исполнителю не выплачивать. По многим причинам. Суть очень многих нехороших вещей, происходящих с "начинающими юристами", в том, что берутся вполне добротные договора, но не подходящие к их ситуации. То есть, к примеру, есть у нас на сайте какой-то договор. Нормальный вполне договор, который я много раз заключал. Но. Заключал его я в определенной ситуации. И для той ситуации он нормален, и никто со мной ничего не сделает по этому договору. А другому и в другой ситуации он уже может не подходить. И часто, чтобы адаптировать договор, в нем всего пару фраз надо изменить, и все будет ОК, но надо всегда обязательно смотреть по ситуации. Один из простых примеров. Заключили вы, допустим, в сентябре договор, в котором оплата производится по-

этапно или после завершения всех работ. Главное — не 100% предоплаты. Прикинули, что дел там при самом плохом раскладе на два месяца, т. е. к ноябрю закончите. И с чистой совестью поставили срок окончания договора 31 октября. Или даже — 31 декабря. Как обычно для "круглости даты" пишут. Так вот, если я вам не заплачу деньги до 31 декабря, я, в принципе, могу вам их вообще больше не платить. Потому что у нас с вами срок договора истек. Может быть, вы подадите в арбитраж. Там — 50 на 50. Знаю и выигранные дела в такой ситуации, и проигранные. Но если речь идет о сайте, то, вы, скорее всего, никаких арбитражей учинять не будете. Потому что посчитаете и придете к выводу, что больше потеряете, чем денег вернете. И таких примеров много.

В общем, к договорам подходите серьезно. Одна неверно оформленная бумажка многим предпринимателям стоила их бизнеса.

Вопрос:

Как называется договор, когда меня принимает на работу заказчик и оплачивает заказ по этому договору?

Ответ:

Это смотря как он вас принимает на работу... Скорее всего, вы имеете в виду трудовой договор на срочную работу. А может он вас по совместительству принимает, мы же не знаем... Как правило, если идет работа с исполнителем, который является физическим лицом, то оформляется либо договор с ним как с физическим лицом (обычный договор на создание сайта или программного обеспечения, только не с юридическим, а с физическим лицом), либо оформляется трудовой договор (наиболее частый вариант), либо вообще ничего не оформляется — такое тоже нередко бывает.

Вопрос (коммерческое предложение):

Здравствуйте. Мне хотелось бы узнать:

- 1) что представляет собой коммерческое предложение по созданию сайта?*
- 2) как правильно его составить (общие советы)?*
- 3) где посмотреть юридически правильно составленный образец?*

Ответ:

>1) что представляет собой коммерческое предложение по созданию сайта?

Обычное коммерческое предложение. Начинающееся словами "Предлагаем Вам"...

>2) как правильно его составить (общие советы)?

В левом верхнем углу — реквизиты компании, в правом — кому (к примеру: "Директору ООО Иванову И. И."). Затем слова "коммерческое предложе-

ние". Затем 1 фраза "Предлагаем Вам" (пишете, что предлагаете). Затем, если одной фразы недостаточно, пишете, что включает в себя комплекс услуг. Затем — стоимость, затем — сроки реализации (в вашем случае сроки изготовления сайта). В конце — дата, подпись. И печать, разумеется.

>3) где посмотреть юридически правильно составленный образец?

А что значит юридически правильно составленный? Коммерческое предложение — это ведь на самом деле ничего особенного. Их по несколько штук на неделе пишут. Главное в коммерческом предложении — суть: что предлагаете и по какой цене.

Пример обычного коммерческого предложения на создание сайта привожу ниже.

/реквизиты вашей организации/	Директору ООО "Эльбрус" Иванову И. И.
-------------------------------	--

Коммерческое предложение

Предлагаем Вам услуги по созданию сайта Вашей компании. Комплекс услуг включает в себя:

- ☐ разработка дизайна;
- ☐ блок "Новости" с системой администрирования;
- ☐ форма отправки сообщений с сайта;
- ☐ программный блок "Тестирование/Анкетирование/Опросы";
- ☐ блок "Вакансии" с системой администрирования;
- ☐ каталог продукции с системой администрирования;
- ☐ система публикации прайс-листов;
- ☐ система полнотекстового поиска по сайту.

Общая стоимость — 20 000 руб.

Верстка всех информационных страниц и обработка графических элементов (фотографии, схемы, карта проезда) входит в вышеназванную стоимость сайта.

Примерный срок исполнения — 1 месяц.

Руководитель IT-студии SoftTime

(здесь подпись)

/Кузнецов М. В./

Вопрос:

Если мы будем заключать соглашения или контракты на создание сайтов, должны ли мы платить налог? И вообще, можно подробнее о налогах в сети? А вот как стать юр. лицом? Что для этого надо? И чего это будет стоить?

Ответ:

>Есть вопрос. Вот если мы будем заключать соглашения или контракты на создание сайтов, должны ли мы платить налог?

Если у вас нет юридического лица, то о контрактах в нормальном понимании этого слова речь вряд ли может идти. Ваш способ проводки денег — это либо простая "наличка", либо трудовые соглашения. Ну, а тут уже Ваше дело — декларировать этот доход или нет. Только не декларируйте, если получаете деньги простым налом — в этом случае это дело уже не только Ваше, но и той фирмы, от которой Вы эти деньги получили. Думаю, они в таком случае будут не очень довольны :)

>И вообще можно подробнее о налогах в сети?

Совершенно не важно, где вы работаете: в сети или нет — налоги для всех одинаковы. Рано или поздно любая успешная команда выходит из тени и как-то легализуется. Если у Вас пойдут дела, Вы неизбежно придете к легализации. В том числе потому, что "серьезные конторы" с частниками работают редко. Хотя бы из-за того, что частник не несет юридической ответственности по своим обязательствам (особенно, когда "наличкой" расплачиваются), и никаких гарантий от него никто не ждет. И в этом случае, когда станете, к примеру, юридическим лицом, вот тут-то и ощутите на себе в полную силу всю прелесть нашего налогового законодательства:

- 18% НДС (если у вас нет фирмы, которая работает по "упрощенке" с такими же фирмами или ИП (индивидуальные предприниматели));
- 13% — подоходный налог на заработную плату;
- 36,8% — начисление на оплату труда;
- 1,2% — социальный налог.

Про налог на прибыль и некоторые другие налоги я пока не говорю, потому что прибыль на начальном этапе при таких налогах извлекается редко: дай бог начинающей студии "по нулям все свести", т. е. зарплату выплатить, налоги уплатить, всякие расходные нужды удовлетворить (бензин, канцелярщина и т. д.), и чтобы на амортизационные расходы что-то осталось.

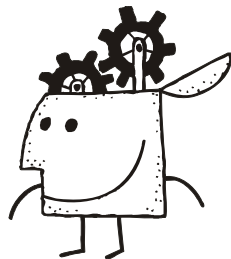
>А вот как стать юр. лицом? Что для этого надо? И чего это будет стоить?

Для этого надо в первую очередь выбрать форму собственности: ООО, ЗАО, какая-то из форм некоммерческой организации и т. д. Форм собственности немало, и тут разговор надо вести более предметно. Решитесь — проконсультируем более подробно.

Но, прежде всего, нужно "подбить бабки" и решить, потянете ли Вы юридическое лицо. Зарегистрироваться-то не очень сложно — сейчас много фирм и частных юристов, которым только деньги плати, они и регистрируют, и уставные документы составят, и ликвидируют потом :) Вопрос не в регистрации, а в том, что будет после. А после — надо жить. И на первых порах, скорее всего, — выживать. В этом ничего страшного нет, через это прошли почти все, но представлять себе это надо четко. Поначалу на вас свалится немало проблем, о существовании которых Вы ранее могли и не подозревать. В общем, здесь все, как в воспитании детей: маленькие дети — маленькие проблемы, большие дети — большие проблемы :)

Понимаете, ведь прежде чем решить становиться/не становиться, надо четко представлять себе цель, для чего все это делается. Все всегда с цели начинается, и образование юридического лица — не исключение. Если человеку нужно юридическое лицо для того, чтобы, грубо говоря, пыль в глаза пустить, — то лучше этого не делать. Такая фирма будет не жить, а кое-как существовать, никого не радуя. Под пусканием пыли в глаза имею в виду следующее. Многие организуют юридические лица сугубо из тщеславных побуждений — для того, чтобы стать в них директорами :) Только сегодня поутру имел крутой и неприятный разговор с одним "директором", который мне пять раз за минуту сказал "Я же ведь — директор!" :) Многие организуют юридические лица для того, чтобы продемонстрировать миру, что у них нынче дела идут лучше, чем раньше — мол, вчера мы были просто командой, а сегодня — уже целая организация. И так далее. Многие — из-за заблуждения, считая, что юридическое лицо избавит их от тех или иных проблем. Да ни от чего не избавит, а вот много новых — создаст. Юридическое лицо надо делать только тогда, когда в нем назрела реальная, а не мифическая необходимость. Вот если Вы говорите, что Вам нужно юридическое лицо, потому что с частниками вообще перестали работать — тогда да. Я первый скажу, что Вы правы. Или для того, чтобы участвовать в каком-то крупном тендере, в котором Вы чувствуете силы участвовать. Или, на худой конец, потому что Ваша команда сказала: "Слушай, хватит жить как попало, хотим фирму, хотим в ней работать, в офисе сидеть, а не по шалашам, хотим социальный пакет, и вроде потянуть ведь все это можем — денег немало заработали и отложили, на первое время хватит". Тогда, скорее всего, тоже — да. А во всех остальных случаях организовывать юр. лицо вряд ли стоит.

Приложение 2



Авторское право

В этом приложении мы рассмотрим основные положения закона об авторском праве и смежных правах. Если не все положения этого закона, то, по крайней мере, его основные понятия желательно знать всем, кто пишет программы, делает сайты, разрабатывает архитектуру приложений. Потому что с точки зрения авторского права — все это произведения, а значит, на них распространяется закон об авторском праве и смежных правах. Наша практика показывает, что с вопросами авторского права по тем или иным причинам в разное время сталкиваются практически все разработчики: у кого-то украли скрипт с сайта и выложили под своим именем, кто-то, наоборот, сам "позаимствовал" что-то с чужого сайта, считая это общедоступной информацией, а на самом деле у нее была группа авторов... В конце этого приложения приведены наиболее типичные вопросы по авторскому праву, которые задают нам посетители нашего форума и читатели наших книг.

По поводу закона об авторском праве и смежных правах хотелось бы отметить две, казалось бы, взаимоисключающие вещи. Во-первых, закон несложный и вполне понятный даже для не юристов. Во-вторых, почему-то именно юристы в нем разбираются, скажем так, не очень хорошо. Да простят они меня за такие нападки, но найти юриста, хорошо знающего закон об авторском праве, дело очень непростое. Знаю по собственному опыту и не только по собственному. Поэтому, если уж помощи ждать неоткуда, значит, надо изучать авторское право самим, и если придется, то и выступать в суде. В чем тоже нет ничего страшного: мне приходилось выступать в суде в качестве эксперта по авторскому праву, и достаточно успешно. Хотя моя специальность далека от юриспруденции. Но, опять же, повторюсь: страшного и исключительного в этом нет ничего. Более того, это, наверное, скоро станет общепринятой нормой, когда человек должен знать те законы, с которыми ему приходится работать, не только не хуже юристов, но и лучше их. По крайней мере, это избавит вас от очень многих проблем, в чем на "своей шкуре" убедились многие успешные бизнесмены, аудиторы и т. д. Не

секрет, что бухгалтера и аудиторы знают налоговый кодекс намного лучше многих "шатных" юристов, бизнесмены прекрасно знают все законы, связанные с предпринимательской деятельностью, связисты — доки в законе о связи, и юристов к этому закону вообще никогда не привлекают, и т. д. И точно так же разработчикам ПО надо разбираться в законе об авторском праве.

Поскольку закон действительно несложный, мы не будем говорить вначале об общих положениях этого закона, как это иногда принято на университетских лекциях, а сразу начнем работать с его текстом. В примечаниях — наши комментарии к этому закону.

Закон

"Об авторском праве и смежных правах"

(с изменениями от 19 июля 1995 г., 20 июля 2004 г.)

Изменения, предусмотренные Федеральным законом от 20 июля 2004 г. № 72-ФЗ, вступили в силу с 28 июля 2004 года (дня официального опубликования Федерального закона в "Российской газете"), кроме изменений к статьям 16, 37, 38 и 39, относящихся к введению нового "права на доведение до всеобщего сведения", которые вступают в силу с 1 сентября 2006 года.

Содержание закона

Раздел I. ОБЩИЕ ПОЛОЖЕНИЯ

Статья 1. Предмет регулирования

Статья 2. Законодательство Российской Федерации об авторском праве и смежных правах

Статья 3. Международные договоры

Статья 4. Основные понятия

Раздел II. АВТОРСКОЕ ПРАВО

Статья 5. Сфера действия авторского права

Статья 6. Объект авторского права. Общие положения

Статья 7. Произведения, являющиеся объектами авторского права

Статья 8. Произведения, не являющиеся объектами авторского права

Статья 9. Возникновение авторского права. Презумпция авторства

Статья 10. Соавторство

Статья 11. Авторское право составителей сборников и других составных произведений

Статья 12. Авторское право переводчиков и авторов других производных произведений

Статья 13. Авторское право на аудиовизуальные произведения

Статья 14. Авторское право на служебные произведения

Статья 15. Личные неимущественные права

Статья 16. Имущественные права

Статья 17. Право доступа к произведениям изобразительного искусства. Право следования

Статья 18. Воспроизведение произведения в личных целях без согласия автора и без выплаты авторского вознаграждения

Статья 19. Использование произведения без согласия автора и без выплаты авторского вознаграждения

Статья 20. Использование произведений путем репродуцирования

Статья 21. Свободное использование произведений, постоянно расположенных в местах, открытых для свободного посещения

Статья 22. Свободное публичное исполнение

Статья 23. Свободное воспроизведение для судебных целей

Статья 24. Свободная запись краткосрочного пользования, производимая организациями эфирного вещания

Статья 25. Свободное воспроизведение программ для ЭВМ и баз данных. Декомпилирование программ для ЭВМ

Статья 26. Воспроизведение произведения в личных целях без согласия автора с выплатой авторского вознаграждения

Статья 27. Срок действия авторского права

Статья 28. Общественное достояние

Статья 29. Переход авторского права по наследству

Статья 30. Передача имущественных прав. Авторский договор

Статья 31. Условия авторского договора

Статья 32. Форма авторского договора

Статья 33. Авторский договор заказа

Статья 34. Ответственность по авторскому договору

Раздел III. СМЕЖНЫЕ ПРАВА

Статья 35. Сфера действия смежных прав

Статья 36. Субъекты смежных прав

Статья 37. Права исполнителя

Статья 38. Права производителя фонограммы

Статья 39. Использование фонограммы, опубликованной в коммерческих целях, без согласия производителя фонограммы и исполнителя

Статья 40. Права организаций эфирного вещания

Статья 41. Права организации кабельного вещания

Статья 42. Ограничения прав исполнителя, производителя фонограммы, организации эфирного или кабельного вещания

Статья 43. Срок действия смежных прав

Раздел IV. КОЛЛЕКТИВНОЕ УПРАВЛЕНИЕ ИМУЩЕСТВЕННЫМИ ПРАВАМИ

Статья 44. Цели коллективного управления имущественными правами

Статья 45. Организации, управляющие имущественными правами на коллективной основе

Статья 46. Функции организаций, управляющих имущественными правами на коллективной основе

Статья 47. Обязанности организаций, управляющих имущественными правами на коллективной основе

Раздел V. ЗАЩИТА АВТОРСКИХ И СМЕЖНЫХ ПРАВ

Статья 48. Нарушение авторских и смежных прав. Контрафактные экземпляры произведения и фонограммы

Статья 48.1. Технические средства защиты авторского права и смежных прав

Статья 48.2. Информация об авторском праве и о смежных правах

Статья 49. Гражданско-правовые способы защиты авторского права и смежных прав

Статья 49.1. Конфискация контрафактных экземпляров произведений или фонограмм

Статья 50. Способы обеспечения иска по делам о нарушении авторских и смежных прав

Тексты статей

Раздел I. ОБЩИЕ ПОЛОЖЕНИЯ

Статья 1. Предмет регулирования

Настоящий Закон регулирует отношения, возникающие в связи с созданием и использованием произведений науки, литературы и искусства (авторское право), фонограмм, исполнений, постановок, передач организаций эфирного или кабельного вещания (смежные права).

Примечание

Как видите, все произведения охраняются двумя видами прав: авторскими правами и смежными. Различие между ними в том, что они охраняют разные категории произведений.

Примечание

Сам процесс создания произведений закон не регулирует — в сфере действия закона только конечные результаты этого процесса, т. е. уже созданные произведения.

Статья 2. Законодательство Российской Федерации об авторском праве и смежных правах

Законодательство Российской Федерации об авторском праве и смежных правах основывается на Конституции Российской Федерации и состоит из Гражданского кодекса Российской Федерации, настоящего Закона, Закона Российской Федерации от 23 сентября 1992 года № 3523-1 "О правовой охране программ для электронных вычислительных машин и баз данных", федеральных законов.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

Статья 3. Международные договоры

Если международным договором, в котором участвует Российская Федерация, установлены иные правила, чем те, которые содержатся в настоящем Законе, то применяются правила международного договора.

Примечание

Эта статья важна для тех, кто много работает с западными партнерами, потому что, по сути, говорит о приоритете международных договоров над национальным законодательством. Статья основывается на п. 4 ст. 15 Конституции РФ, которая, в частности, говорит о следующем: "Если международным договором Российской Федерации установлены иные правила, чем предусмотренные за-

коном, то применяются правила международного договора". Россией подписаны следующие международные договоры в области авторского права: Всемирная конвенция об авторском праве (в новой редакции 1995 г.); Бернская конвенция об охране литературных и художественных произведений (с 1995 г.); ряд двусторонних соглашений с Арменией, Австрией, Болгарией, Венгрией, Китаем, Кубой, Польшей, Чехией, Словакией, Швецией и др.; конвенция об охране интересов производителей фонограмм от незаконного воспроизводства их фонограмм (с 1995 г.), Соглашение о сотрудничестве в области охраны авторского права и смежных прав (с 1993 г.). Поэтому, если вы работаете с иностранными партнерами, изучайте не только наш закон, но и приведенные международные соглашения, потому что работать вам, возможно, придется по ним.

Статья 4. Основные понятия

Для целей настоящего Закона указанные ниже термины имеют следующее значение:

- автор — физическое лицо, творческим трудом которого создано произведение;
- аудиовизуальное произведение — произведение, состоящее из зафиксированной серии связанных между собой кадров (с сопровождением или без сопровождения их звуком), предназначенное для зрительного и слухового (в случае сопровождения звуком) восприятия с помощью соответствующих технических устройств; аудиовизуальные произведения включают кинематографические произведения и все произведения, выраженные средствами, аналогичными кинематографическим (теле- и видеофильмы, диафильмы и слайдфильмы и тому подобные произведения), независимо от способа их первоначальной или последующей фиксации;
- база данных — объективная форма представления и организации совокупности данных (статей, расчетов и т. д.), систематизированных таким образом, чтобы эти данные могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ);

Примечание

Нередкий вопрос: "какой должна быть база данных, чтобы она охранялась законом об авторском праве?". Так вот, согласно этому определению, никакой специальной организации данных в базе для охраны ее законом об авторском праве не требуется. Закон также ничего не говорит и о количестве данных в базе данных.

Примечание

Важно то, что данное определение говорит о том, что база данных только тогда база данных, когда данные обрабатываются с помощью ЭВМ. То есть список

телефонов, записанных на листочке, базой данных не является, а тот же список, занесенный в компьютер, — уже база данных.

- воспроизведение произведения — изготовление одного или более экземпляров произведения или его части в любой материальной форме, в том числе в форме звуко- и видеозаписи, изготовление в трех измерениях одного или более экземпляров двухмерного произведения и в двух измерениях — одного или более экземпляров трехмерного произведения; запись произведения в память ЭВМ также является воспроизведением;

Примечание

Здесь очень важный момент состоит в том, что "запись произведения в память ЭВМ также является воспроизведением". При этом "запись в память" не нужно трактовать слишком узко: запись на жесткий диск, на CD, загрузка файлов на сервер, размещение файлов в Интернете — это все — воспроизведение. То есть скачали вы, к примеру, фонограмму с сайта, — так вот, вы ее воспроизвели не в тот момент, когда стали слушать, а тогда, когда закачали ее себе на жесткий диск.

Примечание

Воспроизведение — это не творческий, а механический процесс, и он не влечет за собой появление нового объекта авторского права. Но воспроизведение нужно отличать от **творческого повтора**. Творческий повтор — это уже не воспроизведение, и при творческом повторе появляется новый объект авторского права. К примеру, копирование от руки картины — это творческий повтор, и на такую копию авторское право распространяется.

- воспроизведение фонограммы — изготовление одного или более экземпляров фонограммы или ее части на любом материальном носителе;
- запись — фиксация звуков и (или) изображений с помощью технических средств в какой-либо материальной форме, позволяющей осуществлять их неоднократное восприятие, воспроизведение или сообщение;
- изготовитель аудиовизуального произведения — физическое или юридическое лицо, взявшее на себя инициативу и ответственность за изготовление такого произведения; при отсутствии доказательств иного изготовителем аудиовизуального произведения признается физическое или юридическое лицо, имя или наименование которого обозначено на этом произведении обычным образом;
- изготовитель фонограммы — физическое или юридическое лицо, взявшее на себя инициативу и ответственность за первую звуковую запись исполнения или иных звуков; при отсутствии доказательств иного изготовителем фонограммы признается физическое или юридическое лицо, имя или наименование которого обозначено на этой фонограмме и (или) на содержащем ее футляре обычным образом;

- исполнение — представление произведений, фонограмм, исполнений, постановок посредством игры, декламации, пения, танца в живом исполнении или с помощью технических средств (телерадиовещания, кабельного телевидения и иных технических средств); показ кадров аудиовизуального произведения в их последовательности (с сопровождением или без сопровождения звуком);
- исполнитель — актер, певец, музыкант, танцор или иное лицо, которое играет роль, читает, декламирует, поет, играет на музыкальном инструменте или иным образом исполняет произведения литературы или искусства (в том числе эстрадный, цирковой или кукольный номер), а также режиссер-постановщик спектакля и дирижер;
- обнародование произведения — осуществленное с согласия автора действие, которое впервые делает произведение доступным для всеобщего сведения путем его опубликования, публичного показа, публичного исполнения, передачи в эфир или иным способом;
- опубликование (выпуск в свет) — выпуск в обращение экземпляров произведения, фонограммы с согласия автора произведения, производителя фонограммы в количестве, достаточном для удовлетворения разумных потребностей публики исходя из характера произведения, фонограммы;
- передача в эфир — сообщение произведений, фонограмм, исполнений, постановок, передач организаций эфирного или кабельного вещания для всеобщего сведения (включая показ или исполнение) посредством их передачи по радио или телевидению (за исключением кабельного телевидения). При передаче произведений, фонограмм, исполнений, постановок, передач организаций эфирного или кабельного вещания в эфир через спутник под передачей в эфир понимается прием сигналов с наземной станции на спутник и передача сигналов со спутника, посредством которых произведения, фонограммы, исполнения, постановки, передачи организаций эфирного или кабельного вещания могут быть доведены до всеобщего сведения независимо от фактического приема их публикой;
- передача организации эфирного или кабельного вещания — передача, созданная самой организацией эфирного или кабельного вещания, а также по ее заказу за счет ее средств другой организацией;
- показ произведения — демонстрация оригинала или экземпляра произведения непосредственно или на экране с помощью пленки, диапозитива, телевизионного кадра или иных технических средств, а также демонстрация отдельных кадров аудиовизуального произведения без соблюдения их последовательности;
- последующая передача в эфир — последующая передача в эфир ранее переданных в эфир произведений, фонограмм, исполнений, постановок, передач организаций эфирного или кабельного вещания;

- ❑ программа для ЭВМ — объективная форма представления совокупности данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определенного результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ, и порождаемые ею аудиовизуальные отображения;

Примечание

С программами для ЭВМ, как всегда, все очень непросто. Возможно, вы будете удивлены, но п. 1 ст. 7 Закона относит программы для ЭВМ к литературным произведениям (см. далее указанную статью). А из этого, в том числе, следует, что в области международного права на программы распространяется Бернская конвенция об охране литературных и художественных произведений, участницей которой является и Россия. Поэтому то, что многие программисты, работающие "на Запад", не изучают Бернскую конвенцию, поскольку она "для литераторов", — это неправильно.

- ❑ произведение декоративно-прикладного искусства — двухмерное или трехмерное произведение искусства, перенесенное на предметы практического пользования, включая произведение художественного промысла или произведение, изготовляемое промышленным способом;
- ❑ публичный показ, публичное исполнение или сообщение для всеобщего сведения — любые показ, исполнение или сообщение произведений, фонограмм, исполнений, постановок, передач организаций эфирного или кабельного вещания непосредственно либо с помощью технических средств в месте, открытом для свободного посещения, или в месте, где присутствует значительное число лиц, не принадлежащих к обычному кругу семьи, независимо от того, воспринимаются ли произведения, фонограммы, исполнения, постановки, передачи организаций эфирного или кабельного вещания в месте их сообщения или в другом месте одновременно с сообщением произведений, фонограмм, исполнений, постановок, передач организаций эфирного или кабельного вещания;
- ❑ режиссер — постановщик спектакля — лицо, осуществившее постановку театрального, циркового, кукольного, эстрадного или иного спектакля (представления);
- ❑ репродуцирование (репрографическое воспроизведение) — факсимильное воспроизведение в любых размере и форме одного или более экземпляров оригиналов или копий письменных и других графических произведений путем фотокопирования или с помощью других технических средств, иных, чем издание; репрографическое воспроизведение не включает в себя хранение или воспроизведение указанных копий в электронной (включая цифровую), оптической или иной машиночитаемой форме;

- ❑ сдавать в прокат (внаем) — предоставлять экземпляры произведения или фонограммы во временное пользование в целях извлечения прямой или косвенной коммерческой выгоды;
- ❑ сообщать — показывать, исполнять, передавать в эфир или совершать иное действие (за исключением распространения экземпляров произведения или фонограммы), посредством которого произведения, фонограммы, исполнения, постановки, передачи организаций эфирного или кабельного вещания становятся доступными для слухового и (или) зрительного восприятия, независимо от их фактического восприятия публикой;
- ❑ сообщать для всеобщего сведения по кабелю — сообщать произведения, фонограммы, исполнения, постановки, передачи организаций эфирного или кабельного вещания для всеобщего сведения посредством кабеля, провода, оптического волокна или с помощью аналогичных средств;
- ❑ фонограмма — любая исключительно звуковая запись исполнений или иных звуков;
- ❑ экземпляр произведения — копия произведения, изготовленная в любой материальной форме;
- ❑ экземпляр фонограммы — копия фонограммы на любом материальном носителе, изготовленная непосредственно или косвенно с фонограммы и включающая все звуки или часть звуков, зафиксированных в этой фонограмме.

Раздел II. АВТОРСКОЕ ПРАВО

Статья 5. Сфера действия авторского права

1. Авторское право:

- 1) распространяется на произведения, обнародованные на территории Российской Федерации или необнародованные, но находящиеся в какой-либо объективной форме на территории Российской Федерации, и признается за авторами (их правопреемниками) независимо от их гражданства;
- 2) распространяется на произведения, обнародованные за пределами территории Российской Федерации или необнародованные, но находящиеся в какой-либо объективной форме за пределами территории Российской Федерации, и признается за авторами, являющимися гражданами Российской Федерации (их правопреемниками);
- 3) распространяется на произведения, обнародованные за пределами территории Российской Федерации или необнародованные, но находящиеся в какой-либо объективной форме за пределами территории Российской Федерации

Федерации, и признается на территории Российской Федерации за авторами (их правопреемниками) — гражданами других государств в соответствии с международными договорами Российской Федерации.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

2. Произведение также считается впервые опубликованным в Российской Федерации, если в течение 30 дней после даты первого опубликования за пределами Российской Федерации оно было опубликовано на территории Российской Федерации.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

3. При предоставлении на территории Российской Федерации охраны произведению в соответствии с международными договорами Российской Федерации автор произведения определяется по закону государства, на территории которого имел место юридический факт, послуживший основанием для обладания авторским правом.
4. Предоставление на территории Российской Федерации охраны произведению в соответствии с международными договорами Российской Федерации осуществляется в отношении произведений, не перешедших в общественное достояние в стране происхождения произведения вследствие истечения установленного в такой стране срока действия авторского права и не перешедших в общественное достояние в Российской Федерации вследствие истечения предусмотренного настоящим Законом срока действия авторского права.

При предоставлении охраны произведению в соответствии с международными договорами Российской Федерации срок действия авторского права на территории Российской Федерации не может превышать срок действия авторского права, установленный в стране происхождения произведения.

(Федеральный закон от 20 июля 2004 г. № 72-ФЗ)

Примечание

Произведение получает охрану законом, если оно отвечает хотя бы одному критерию, содержащемуся в данной статье. Это критерии гражданства и места первого опубликования.

Примечание

С критерием гражданства, вроде бы, все понятно, а вот с критерием места первого опубликования существует ряд тонкостей. Во-первых, если произведение было впервые обнародовано в стране, с которой Россия не имеет международных договоров по авторским правам, то это произведение не охраняется законом об авторском праве. Во-вторых, также следует считать неохраняемыми произведения, впервые обнародованные, скажем, в космосе, открытом море,

Антарктиде. Но это — неохраняемость по критерию места первого опубликования, а ведь есть еще и критерий гражданства: а закон говорит, что авторское право вступает в действие, если удовлетворяется хотя бы один из двух критериев. И если даже, к примеру, российский гражданин создаст произведение на территории страны, с которой у России нет международных договоров (неохраняемость произведения по критерию места первого опубликования), то это произведение тем не менее будет охраняться, так как автор произведения является российским гражданином (соблюдается критерий гражданства).

Статья 6. Объект авторского права.

Общие положения

1. Авторское право распространяется на произведения науки, литературы и искусства, являющиеся результатом творческой деятельности, независимо от назначения и достоинства произведения, а также от способа его выражения.
2. Авторское право распространяется как на обнародованные произведения, так и на необнародованные произведения, существующие в какой-либо объективной форме:
 - письменной (рукопись, машинопись, нотная запись и т. д.);
 - устной (публичное произнесение, публичное исполнение и т. д.);
 - звуко- или видеозаписи (механической, магнитной, цифровой, оптической и т. д.);
 - изображения (рисунок, эскиз, картина, план, чертеж, кино-, теле-, видео- или фотокадр и т. д.);
 - объемно-пространственной (скульптура, модель, макет, сооружение и т. д.);
 - в других формах.
3. Часть произведения (включая его название), которая удовлетворяет требованиям пункта 1 настоящей статьи и может использоваться самостоятельно, является объектом авторского права.
4. Авторское право не распространяется на идеи, методы, процессы, системы, способы, концепции, принципы, открытия, факты.
5. Авторское право на произведение не связано с правом собственности на материальный объект, в котором произведение выражено.

Передача права собственности на материальный объект или права владения материальным объектом сама по себе не влечет передачи каких-либо авторских прав на произведение, выраженное в этом объекте, за исключением случаев, предусмотренных ст. 17 настоящего Закона.

Примечание

Заметим такой интересный момент: согласно п. 1, 2 ст. 6 "произведение — это результат творческой деятельности автора". А творческая деятельность — это деятельность человеческого мозга, который не создает предметы материального мира. Однако охрана произведения наступает только с момента, когда результат творческой деятельности представлен в какой-либо материальной форме. При этом сама "материальная форма" не охраняется авторским правом — она охраняется правом собственности.

О субъективной и объективной новизне

Различают объективную и субъективную новизну творческого результата. **Субъективная новизна** — это неизвестность полученного результата для самого создателя. А **объективная новизна** — это неизвестность полученного результата не только для первооткрывателя этого результата, но и для остальных лиц. Так вот, авторским правом охраняются только творческие результаты, обладающие объективной новизной.

О параллельно полученных результатах

Интересный момент состоит в том, что авторское право не охраняет творческие результаты, которые созданы параллельно лицами, работающими независимо друг от друга. Дело в том, что если бы такие результаты охранялись авторским правом, то пришлось бы либо признавать соавторами всех лиц, получивших данный результат, либо признавать автором только то лицо, которое первым получило данный результат (создавать правила приоритета). Но ни первого, ни второго в авторском праве нет! Еще раз повторим: в законе нет правила приоритета для лиц, получивших результат параллельно. Поэтому когда, скажем, ученые говорят фразу типа "Мы обогнали научную группу из Н-ска на месяц в получении этого результата и приоритет в получении авторских прав на этот результат за нами" — они ошибаются. Потому что закон в данном случае не будет охранять результат, а только форму его изложения, скажем, научную статью, которая уже будет считаться самостоятельным произведением. Поэтому вопрос с авторскими правами на результаты, полученные параллельно, — сложный вопрос, который регулируется уже не только авторским правом, а, к примеру, и патентным правом и другими законами РФ. Вывод: авторское право охраняет не все объективно новые творческие результаты, а только те из них, которые не повторяются при параллельном творчестве, т. е. являются уникальными. Неповторяющиеся творческие результаты называют еще **оригинальными**. Кстати, в законах об авторском праве многих других стран об этом написано прямо: авторским правом охраняются только **оригинальные творческие результаты**. К примеру, в г. Москве было заседание суда по поводу действия авторского права на словосочетание "петербургские тайны", при этом имелось в виду название известного телевизионного сериала. Однако было установлено, что авторы сериала не имеют права претендовать на новое авторское право на это словосочетание, так как век назад примерно в одно и то же время в Санкт-Петербурге вышли в свет литературные произведения с такими же названиями, но принадлежащие разным авторам. Поэтому, даже если авторы сериала вложили новый смысл в эти слова, авторское право на них им принадлежать не может.

Об оригинальных частях произведения

В п. 3 данной статьи говорится об очень важном принципе: авторским правом охраняется не только произведение в целом, но и любая его оригинальная часть. К примеру, строфа стихотворения охраняется авторским правом так же, как и само стихотворение. При этом некоторые части произведения оригинальными не являются и их использование третьими лицами авторских прав не нарушает. Однако неоригинальные части могут охраняться, к примеру, как товарные знаки, или другими законодательствами. Например, если название вашей фирмы очернили в рекламном ролике, вы не можете подать в суд на очернителя за нарушение авторского права, так как названия фирм редко признаются оригинальными. Но в этом случае вас охраняет закон о недобросовестной конкуренции.

Примечание

Важно то, что п. 5. ст. 6 устанавливается, что авторское право на произведение и право собственности на материальный объект, с помощью которого это произведение выражено, — это совершенно разные, существующие независимо друг от друга вещи.

Статья 7. Произведения, являющиеся объектами авторского права

1. Объектами авторского права являются:

- литературные произведения (включая программы для ЭВМ);

Примечание

Важный момент — программы относятся к категории литературных произведений. Поэтому далее в законе все, что относится к "литературному произведению", — относится и к программам, даже если специально этого не указано. К сожалению, этот пункт не знают и некоторые юристы. Один юрист, к примеру, в споре со мной утверждал, что один из пунктов авторского права не относится к программным продуктам, потому что в законе специально об этом не сказано.

Примечание

К литературным произведениям также относятся, к примеру, химические формулы, нотные знаки, изображенные на бумаге, и т. д.

- драматические и музыкально-драматические произведения, сценарные произведения;
- хореографические произведения и пантомимы;
- музыкальные произведения с текстом или без текста;
- аудиовизуальные произведения (кино-, теле- и видеофильмы, слайд-фильмы, диафильмы и другие кино- и телепроизведения);

- произведения живописи, скульптуры, графики, дизайна, графические рассказы, комиксы и другие произведения изобразительного искусства;
 - произведения декоративно-прикладного и сценографического искусства;
 - произведения архитектуры, градостроительства и садово-паркового искусства;
 - фотографические произведения и произведения, полученные способами, аналогичными фотографии;
 - географические, геологические и другие карты, планы, эскизы и пластические произведения, относящиеся к географии, топографии и к другим наукам;
 - другие произведения.
2. Охрана программ для ЭВМ распространяется на все виды программ для ЭВМ (в том числе на операционные системы), которые могут быть выражены на любом языке и в любой форме, включая исходный текст и объектный код.

Примечание

Перевод программы с одного языка на другой не является переводом в смысле авторского права.

3. К объектам авторского права также относятся:
- производные произведения (переводы, обработки, аннотации, рефераты, резюме, обзоры, инсценировки, аранжировки и другие переработки произведений науки, литературы и искусства);
 - сборники (энциклопедии, антологии, базы данных) и другие составные произведения, представляющие собой по подбору или расположению материалов результат творческого труда;
 - производные произведения и составные произведения охраняются авторским правом независимо от того, являются ли объектами авторского права произведения, на которых они основаны или которые они включают.

О сборниках

Технический труд по составлению сборника не приводит к появлению нового объекта авторского права. К примеру, телефонная книга с номерами жителей города не будет охраняться авторским правом, но может охраняться как база данных, если занесена в компьютер.

Статья 8. Произведения, не являющиеся объектами авторского права

Не являются объектами авторского права:

- ☐ официальные документы (законы, судебные решения, иные тексты законодательного, административного и судебного характера), а также их официальные переводы;
- ☐ государственные символы и знаки (флаги, гербы, ордена, денежные знаки и иные государственные символы и знаки);

Примечание

К примеру, согласно этому пункту, С. Михалков как автор слов к гимну России не является обладателем каких-либо авторских прав на этот гимн.

- ☐ произведения народного творчества;
- ☐ сообщения о событиях и фактах, имеющие информационный характер.

Статья 9. Возникновение авторского права. Презумпция авторства

1. Авторское право на произведение науки, литературы и искусства возникает в силу факта его создания. Для возникновения и осуществления авторского права не требуется регистрации произведения, иного специального оформления произведения или соблюдения каких-либо формальностей.

Обладатель исключительных авторских прав для оповещения о своих правах вправе использовать знак охраны авторского права, который помещается на каждом экземпляре произведения и состоит из трех элементов:

- латинской буквы "С" в окружности: "©";
- имени (наименования) обладателя исключительных авторских прав;
- года первого опубликования произведения.

Примечание

К примеру: © Кузнецов М. В., Симдянов И. В., 1994.

2. При отсутствии доказательств иного автором произведения считается лицо, указанное в качестве автора на оригинале или экземпляре произведения.
3. При опубликовании произведения анонимно или под псевдонимом (за исключением случая, когда псевдоним автора не оставляет сомнения в его личности) издатель, имя или наименование которого обозначено на произведении, при отсутствии доказательств иного считается представителем автора в соответствии с настоящим Законом и в этом качестве имеет право защищать права автора и обеспечивать их осуществление.

Это положение действует до тех пор, пока автор такого произведения не раскроет свою личность и не заявит о своем авторстве.

Примечание

Основной принцип всей системы авторского права — то, что авторское право возникает с момента создания произведения, а не с момента его обнародования. Но вместе с тем, как мы уже говорили, так как авторское право не знает понятия приоритета, точное время создания произведения не является серьезным правовым моментом — главное сам факт появления произведения. Приоритет — это область патентного права. Однако мы все-таки советуем созданные произведения обнародовать. Потому что авторство иногда приходится доказывать. И если вы намерены заранее обеспечить доказательства своего авторства на то или иное произведение, то можете обратиться, к примеру, к нотариусу, который зарегистрирует ваше произведение. Другой вариант — обращение в Российское Авторское Общество (РАО), в которое вы можете обратиться с просьбой о регистрации вашего произведения как объекта интеллектуальной собственности. Регистрация — платная, при этом размер оплаты зависит от того, как высоко вы оцените созданное вами произведение.

Статья 10. Соавторство

1. Авторское право на произведение, созданное совместным творческим трудом двух или более лиц (соавторство), принадлежит соавторам совместно независимо от того, образует ли такое произведение одно неразрывное целое или состоит из частей, каждая из которых имеет самостоятельное значение.

Часть произведения признается имеющей самостоятельное значение, если она может быть использована независимо от других частей этого произведения.

Каждый из соавторов вправе использовать созданную им часть произведения, имеющую самостоятельное значение, по своему усмотрению, если иное не предусмотрено соглашением между ними.

2. Право на использование произведения в целом принадлежит соавторам совместно.

Взаимоотношения соавторов могут определяться соглашением между ними.

Если произведение соавторов образует одно неразрывное целое, то ни один из соавторов не вправе без достаточных к тому оснований запретить использование произведения.

О соавторстве

Соавторство устанавливается по полученному результату, а не по процессу работы: т. е. если результат труда какого-то лица не нашел отражения в конечном произведении, то такое лицо соавтором не является. Это следует из уже

обсуждаемого нами ранее положения о том, что авторское право не распространяется на сам процесс создания творческого произведения, а только на конечный результат этого процесса. Интересен тот факт, что лицо, принимавшее участие в генерации идей, которые потом нашли отражение в произведении, соавтором не становится в силу п. 4. ст. 6.

Соглашение о соавторстве

Соавторами признаются те лица, которые создали произведение "совместным трудом". При этом под совместным трудом понимается соглашение между соавторами о создании соавторского произведения. Если такого соглашения нет, то ни о каком соавторстве нет речи.

О делимом и неделимом соавторстве

Соавторство бывает делимым и неделимым. Делимое соавторство, это когда, к примеру, авторы слов и музыки к песне заключили между собой договор о соавторстве, согласно которому авторство слов принадлежит одному, а авторство музыки другому соавтору. В этом случае они могут по своему усмотрению распоряжаться "своими" частями авторского права (поэт — словами, композитор — музыкой), не ставя в известность другого соавтора. Но в результате их совместного авторства рождается новый объект авторского права — песня. Неделимое соавторство основывается на том, что авторы, по соглашению или без оного, создают неделимое произведение. К примеру, если относительно той же песни будет известно, что авторы песни Иванов и Петров, но между ними нет договора о соавторстве, в котором они разделяют права на части произведения (к примеру, на слова и музыку), или есть договор, в котором они договорились, что произведение неделимое, то в этом случае произведение принадлежит в равной степени обоим соавторам. И ни один из них не имеет права использовать произведение либо его часть без согласия другого соавтора.

Статья 11. Авторское право составителей сборников и других составных произведений

1. Автору сборника и других составных произведений (составителю) принадлежит авторское право на осуществленные им подбор или расположение материалов, представляющие результат творческого труда (составительство).

Составитель пользуется авторским правом при условии соблюдения им прав авторов каждого из произведений, включенных в составное произведение.

Авторы произведений, включенных в составное произведение, вправе использовать свои произведения независимо от составного произведения, если иное не предусмотрено авторским договором.

Авторское право составителя не препятствует другим лицам осуществлять самостоятельный подбор или расположение тех же материалов для создания своих составных произведений.

2. Издателю энциклопедий, энциклопедических словарей, периодических и продолжающихся сборников научных трудов, газет, журналов и других периодических изданий принадлежат исключительные права на использование таких изданий. Издатель вправе при любом использовании таких изданий указывать свое наименование либо требовать такого указания.

Авторы произведений, включенных в такие издания, сохраняют исключительные права на использование своих произведений независимо от издания в целом.

Статья 12. Авторское право переводчиков и авторов других производных произведений

1. Переводчикам и авторам других производных произведений принадлежит авторское право на осуществленные ими перевод, переработку, аранжировку или другую переработку.

Переводчик и автор другого производного произведения пользуется авторским правом на созданное им произведение при условии соблюдения им прав автора произведения, подвергшегося переводу, переработке, аранжировке или другой переработке.

2. Авторское право переводчиков и авторов других производных произведений не препятствует иным лицам осуществлять свои переводы и переработки тех же произведений.

Статья 13. Авторское право на аудиовизуальные произведения

1. Авторами аудиовизуального произведения являются:

- режиссер-постановщик;
- автор сценария (сценарист);
- автор музыкального произведения (с текстом или без текста), специально созданного для этого аудиовизуального произведения (композитор).

2. Заключение договора на создание аудиовизуального произведения влечет за собой передачу авторами этого произведения изготовителю аудиовизуального произведения исключительных прав на воспроизведение, распространение, публичное исполнение, сообщение по кабелю для всеобщего сведения, передачу в эфир или любое другое публичное сообщение аудиовизуального произведения, а также на субтитрование и дублирование текста аудиовизуального произведения, если иное не предусмотрено в договоре. Указанные права действуют в течение срока действия авторского права на аудиовизуальное произведение.

Изготовитель аудиовизуального произведения вправе при любом использовании этого произведения указывать свое имя или наименование либо требовать такого указания.

3. При публичном исполнении аудиовизуального произведения автор музыкального произведения (с текстом или без текста) сохраняет право на вознаграждение за публичное исполнение его музыкального произведения.
4. Авторы произведений, вошедших составной частью в аудиовизуальное произведение, как существовавших ранее (автор романа, положенного в основу сценария, и другие), так и созданных в процессе работы над ним (оператор-постановщик, художник-постановщик и другие), пользуются авторским правом каждый на свое произведение.

Статья 14. Авторское право на служебные произведения

1. Авторское право на произведение, созданное в порядке выполнения служебных обязанностей или служебного задания работодателя (служебное произведение), принадлежит автору служебного произведения.
2. Исключительные права на использование служебного произведения принадлежат лицу, с которым автор состоит в трудовых отношениях (работодателю), если в договоре между ним и автором не предусмотрено иное.

Размер авторского вознаграждения за каждый вид использования служебного произведения и порядок его выплаты устанавливаются договором между автором и работодателем.

3. Работодатель вправе при любом использовании служебного произведения указывать свое наименование либо требовать такого указания.
4. На создание в порядке выполнения служебных обязанностей или служебного задания работодателя энциклопедий, энциклопедических словарей, периодических и продолжающихся сборников научных трудов, газет, журналов и других периодических изданий (п. 2 ст. 11 настоящего Закона) положения настоящей статьи не распространяются.

Примечание

Эта статья очень важная и в ее рамках происходит большое количество споров по вопросам авторского права. Каждый из нас создает большое число служебных произведений в рамках своих служебных обязанностей. Так вот: авторские права на служебное произведение всегда принадлежат автору служебного произведения. А исключительные права на использование принадлежат работодателю. То есть вы, как работник, не можете использовать служебное произведение в коммерческих целях. И части этого произведения в том числе. К примеру, создали вы сайт, что и входило в ваши служебные обязанности, и за что вам выплачивалась зарплата — ни дизайн, ни один из программных блоков вы в коммерческих целях использовать не можете: на все это у вас есть только те права, которые дает вам авторское право (см. ниже).

Примечание

Служебные обязанности могут возникать только при наличии трудового договора. Согласно ст. 15, 56 Трудового кодекса РФ, трудовой договор — это соглашение между работодателем и работником, в котором работодатель обязуется предоставить работнику работу и выплачивать зарплату, а работник обязуется лично выполнять определенную этим договором трудовую функцию. То есть если, к примеру, есть трудовая книжка, социальный пакет — то это явные признаки трудовых отношений.

Примечание

Еще раз отметим, что при заключении трудового договора работодателю переходят все исключительные авторские права на служебное произведение. То есть у автора служебного произведения остаются только личные неимущественные права: право авторства, право на имя, право на обнародование и право на защиту репутации автора (п. 1 ст. 15).

Статья 15. Личные неимущественные права

1. Автору в отношении его произведения принадлежат следующие личные неимущественные права:
 - право признаваться автором произведения (право авторства);
 - право использовать или разрешать использовать произведение под подлинным именем автора, псевдонимом либо без обозначения имени, т. е. анонимно (право на имя);
 - право обнародовать или разрешать обнародовать произведение в любой форме (право на обнародование), включая право на отзыв;

Примечание

Под правом на отзыв понимается отказ автора от ранее принятого им решения на обнародование произведения.

- право на защиту произведения, включая его название, от всякого искажения или иного посягательства, способного нанести ущерб чести и достоинству автора (право на защиту репутации автора).
2. Автор имеет право отказаться от ранее принятого решения об обнародовании произведения (право на отзыв) при условии возмещения пользователю причиненных таким решением убытков, включая упущенную выгоду. Если произведение уже было обнародовано, автор обязан публично оповестить о его отзыве. При этом он вправе изъять за свой счет из обращения ранее изготовленные экземпляры произведения. При создании служебных произведений положения настоящего пункта не применяются.

Примечание

Последняя фраза этого пункта говорит о том, что автор не вправе использовать право на отзыв, если произведение является служебным. Однако автор может не передавать свое произведение работодателю, возместив при этом стоимость затраченных на создание произведения средств. Поэтому эта норма начинает действовать лишь тогда, когда автор уже передал служебное произведение работодателю.

3. Личные неимущественные права принадлежат автору независимо от его имущественных прав и сохраняются за ним в случае уступки исключительных прав на использование произведения.

Статья 16. Имущественные права

1. Автору в отношении его произведения принадлежат исключительные права на использование произведения в любой форме и любым способом.
2. Исключительные права автора на использование произведения означают право осуществлять или разрешать следующие действия:
 - воспроизводить произведение (право на воспроизведение);
 - распространять экземпляры произведения любым способом: продавать, сдавать в прокат и т. д. (право на распространение);
 - импортировать экземпляры произведения в целях распространения, включая экземпляры, изготовленные с разрешения обладателя исключительных авторских прав (право на импорт);
 - публично показывать произведение (право на публичный показ);
 - публично исполнять произведение (право на публичное исполнение);
 - сообщать произведение (включая показ, исполнение или передачу в эфир) для всеобщего сведения путем передачи в эфир и (или) последующей передачи в эфир (право на передачу в эфир);
 - сообщать произведение (включая показ, исполнение или передачу в эфир) для всеобщего сведения по кабелю, проводам или с помощью иных аналогичных средств (право на сообщение для всеобщего сведения по кабелю);
 - переводить произведение (право на перевод);
 - перерабатывать, аранжировать или другим образом перерабатывать произведение (право на переработку);
 - сообщать произведение таким образом, при котором любое лицо может иметь доступ к нему в интерактивном режиме из любого места и в любое время по своему выбору (право на доведение до всеобщего сведения).

(Федеральный закон от 20 июля 2004 г. № 72-ФЗ)

2.1. Исключительные права автора на использование дизайнерского, архитектурного, градостроительного и садово-паркового проекта включают также практическую реализацию таких проектов. Автор принятого архитектурного проекта вправе требовать от заказчика предоставления права на участие в реализации своего проекта при разработке документации для строительства и при строительстве здания или сооружения, если иное не предусмотрено в договоре.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

3. Если экземпляры правомерно опубликованного произведения введены в гражданский оборот посредством их продажи, то допускается их дальнейшее распространение без согласия автора и без выплаты авторского вознаграждения.

Право на распространение экземпляров произведения путем сдачи их в прокат принадлежит автору независимо от права собственности на эти экземпляры.

4. Размер и порядок исчисления авторского вознаграждения за каждый вид использования произведения устанавливаются в авторском договоре, а также в договорах, заключаемых организациями, управляющими имущественными правами авторов на коллективной основе, с пользователями.
5. Ограничения указанных в п. 2 настоящей статьи прав авторов устанавливаются ст. 17—26 настоящего Закона при условии, что такое использование не наносит неоправданного ущерба нормальному использованию произведения и не ущемляет необоснованным образом законные интересы автора.

Статья 17. Право доступа к произведениям изобразительного искусства.

Право следования

1. Автор произведения изобразительного искусства вправе требовать от собственника произведения предоставления возможности осуществления права на воспроизведение своего произведения (право доступа). При этом от собственника произведения нельзя требовать доставки произведения автору.
2. Переход права собственности на произведение изобразительного искусства (возмездно или безвозмездно) от автора к другому лицу означает первую продажу этого произведения.

В каждом случае публичной перепродажи произведения изобразительного искусства (через аукцион, галерею изобразительного искусства, художественный салон, магазин и т. д.) по цене, превышающей предыдущую не менее чем на 20 процентов, автор имеет право на получение от продавца вознаграждения в размере 5 процентов от перепродажной цены

(право следования). Указанное право является неотчуждаемым и переходит только к наследникам автора по закону на срок действия авторского права.

Статья 18. Воспроизведение произведения в личных целях без согласия автора и без выплаты авторского вознаграждения

1. Допускается без согласия автора и без выплаты авторского вознаграждения воспроизведение правомерно обнародованного произведения исключительно в личных целях, за исключением случаев, предусмотренных ст. 26 настоящего Закона.
2. Положение п. 1 настоящей статьи не применяется в отношении:
 - воспроизведения произведений архитектуры в форме зданий и аналогичных сооружений;
 - воспроизведения баз данных или существенных частей из них;
 - воспроизведения программ для ЭВМ, за исключением случаев, предусмотренных ст. 25 настоящего Закона;
 - репродуцирования книг (полностью) и нотных текстов.

Примечание

Как видно, п. 2. ст. 18 запрещает воспроизведение программ. То есть, попросту говоря, вы не можете без согласия автора даже скачать программу из Интернета, независимо от того, будете вы ею пользоваться потом или нет (согласно п. 7 ст. 4). Скачивать программу, пользоваться программой вы можете только в том случае, если на то есть разрешение автора, которое, к примеру, может быть написано в лицензионном соглашении, распространяемом с программой, или просто на сайте автора. Если такого разрешения нет, то, строго говоря, вы не имеете права скачивать программу, даже если нет никаких явных запретов на скачивание. И уж, тем более, не можете передвигать программу третьим лицам.

Статья 19. Использование произведения без согласия автора и без выплаты авторского вознаграждения

1. Допускается без согласия автора и без выплаты авторского вознаграждения, но с обязательным указанием имени автора, произведение которого используется, и источника заимствования:

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

1) цитирование в оригинале и в переводе в научных, исследовательских, полемических, критических и информационных целях из правомерно обнародованных произведений в объеме, оправданном целью цитирования, включая воспроизведение отрывков из газетных и журнальных статей в форме обзоров печати;

О цитировании

Здесь важно то, что цитирование допускается только с определенной целью. При этом объем цитирования определяется его целью: т. е. если, к примеру, вы пишете критическую статью, в которой критикуете один абзац из произведения кого-то автора, то цитировать вы можете только тот абзац, который критикуете, а не половину произведения и даже не два абзаца.

- 2) использование правомерно обнародованных произведений и отрывков из них в качестве иллюстраций в изданиях, в радио- и телепередачах, звуко- и видеозаписях учебного характера в объеме, оправданном поставленной целью;
 - 3) воспроизведение в газетах, передача в эфир или сообщение по кабелю для всеобщего сведения правомерно опубликованных в газетах или журналах статей по текущим экономическим, политическим, социальным и религиозным вопросам или переданных в эфир произведений такого же характера в случаях, когда такие воспроизведение, передача в эфир или сообщение по кабелю не были специально запрещены автором;
 - 4) воспроизведение в газетах, передача в эфир или сообщение по кабелю для всеобщего сведения публично произнесенных политических речей, обращений, докладов и других аналогичных произведений в объеме, оправданном информационной целью. При этом за автором сохраняется право на опубликование таких произведений в сборниках;
 - 5) воспроизведение или сообщение для всеобщего сведения в обзорах текущих событий средствами фотографии, путем передачи в эфир или сообщения для всеобщего сведения по кабелю произведений, которые становятся увиденными или услышанными в ходе таких событий, в объеме, оправданном информационной целью. При этом за автором сохраняется право на опубликование таких произведений в сборниках;
 - 6) воспроизведение правомерно обнародованных произведений без извлечения прибыли рельефно-точечным шрифтом или другими специальными способами для слепых, кроме произведений, специально созданных для таких способов воспроизведения.
2. Допускается без согласия автора и без выплаты авторского вознаграждения предоставление во временное безвозмездное пользование библиотеками экземпляров произведений, введенных в гражданский оборот законным путем. При этом экземпляры произведений, выраженных в цифровой форме, в том числе экземпляры произведений, предоставляемых в порядке взаимного использования библиотечных ресурсов, могут предоставляться во временное безвозмездное пользование только в помещениях библиотек при условии исключения возможности создать копии этих произведений в цифровой форме.

(Федеральный закон от 20 июля 2004 г. № 72-ФЗ)

Примечание

Под автором в данной статье понимается владелец исключительных авторских прав.

Статья 20. Использование произведений путем репродуцирования

Допускается без согласия автора и без выплаты авторского вознаграждения, но с обязательным указанием имени автора, произведение которого используется, и источника заимствования репродуцирование в единичном экземпляре без извлечения прибыли:

- правомерно опубликованного произведения библиотеками и архивами для восстановления, замены утраченных или испорченных экземпляров, предоставления экземпляров произведения другим библиотекам, утратившим по каким-либо причинам произведения из своих фондов;
- отдельных статей и малообъемных произведений, правомерно опубликованных в сборниках, газетах и других периодических изданиях, коротких отрывков из правомерно опубликованных письменных произведений (с иллюстрациями или без иллюстраций) библиотеками и архивами по запросам физических лиц в учебных и исследовательских целях;
- отдельных статей и малообъемных произведений, правомерно опубликованных в сборниках, газетах и других периодических изданиях, коротких отрывков из правомерно опубликованных письменных произведений (с иллюстрациями или без иллюстраций) образовательными учреждениями для аудиторных занятий.

Статья 21. Свободное использование произведений, постоянно расположенных в местах, открытых для свободного посещения

Допускается без согласия автора и без выплаты авторского вознаграждения воспроизведение, передача в эфир или сообщение для всеобщего сведения по кабелю произведений архитектуры, фотографии, изобразительного искусства, которые постоянно расположены в месте, открытом для свободного посещения, за исключением случаев, когда изображение произведения является основным объектом таких воспроизведения, передачи в эфир или сообщения для всеобщего сведения по кабелю или когда изображение произведения используется для коммерческих целей.

Статья 22. Свободное публичное исполнение

Допускается без согласия автора и без выплаты авторского вознаграждения публичное исполнение музыкальных произведений во время официальных и религиозных церемоний, а также похорон в объеме, оправданном характером таких церемоний.

Статья 23. Свободное воспроизведение для судебных целей

Допускается без согласия автора и без выплаты авторского вознаграждения воспроизведение произведений для судебного производства в объеме, оправданном этой целью.

Статья 24. Свободная запись краткосрочного пользования, производимая организациями эфирного вещания

Организация эфирного вещания может без согласия автора и без выплаты дополнительного вознаграждения делать запись краткосрочного пользования того произведения, в отношении которого эта организация получила право на передачу в эфир, при условии, что такая запись производится организацией эфирного вещания с помощью ее собственного оборудования и для ее собственных передач. При этом организация обязана уничтожить такую запись в течение шести месяцев после ее изготовления, если более продолжительный срок не был согласован с автором записанного произведения. Такая запись может быть сохранена без согласия автора произведения в официальных архивах, если запись носит исключительно документальный характер.

Статья 25. Свободное воспроизведение программ для ЭВМ и баз данных. Декомпилирование программ для ЭВМ**Примечание**

На эту статью программистам следует обратить внимание, так как она говорит непосредственно о программах.

1. Лицо, правомерно владеющее экземпляром программы для ЭВМ или базы данных, вправе без получения разрешения автора или иного обладателя исключительных прав на использование произведения и без выплаты дополнительного вознаграждения:
 - 1) внести в программу для ЭВМ или базу данных изменения, осуществляемые исключительно в целях ее функционирования на технических средствах пользователя, осуществлять любые действия, связанные с функционированием программы для ЭВМ или базы данных в соответствии с ее назначением, в том числе запись и хранение в памяти ЭВМ (одной ЭВМ или одного пользователя сети), а также исправление явных ошибок, если иное не предусмотрено договором с автором;
 - 2) изготовить копию программы для ЭВМ или базы данных при условии, что эта копия предназначена только для архивных целей и для замены правомерно приобретенного экземпляра в случаях, когда оригинал программы для ЭВМ или базы данных утерян, уничтожен или стал непригоден для использования. При этом копия программы для ЭВМ или базы

данных не может быть использована для иных целей, чем указано в подпункте 1 настоящего пункта, и должна быть уничтожена в случае, если владение экземпляром этой программы для ЭВМ или базы данных перестает быть правомерным.

2. Лицо, правомерно владеющее экземпляром программы для ЭВМ, вправе без согласия автора или иного обладателя исключительных прав и без выплаты дополнительного вознаграждения воспроизвести и преобразовать объектный код в исходный текст (декомпилировать программу для ЭВМ) или поручить иным лицам осуществить эти действия, если они необходимы для достижения способности к взаимодействию независимо разработанной этим лицом программы для ЭВМ с другими программами, которые могут взаимодействовать с декомпилируемой программой, при соблюдении следующих условий:

- 1) информация, необходимая для достижения способности к взаимодействию, ранее не была доступна этому лицу из других источников;

- 2) указанные действия осуществляются в отношении только тех частей декомпилируемой программы для ЭВМ, которые необходимы для достижения способности к взаимодействию;

- 3) информация, полученная в результате декомпилирования, может использоваться лишь для достижения способности к взаимодействию независимо разработанной программы для ЭВМ с другими программами, не может передаваться иным лицам, за исключением случаев, если это необходимо для достижения способности к взаимодействию независимо разработанной программы для ЭВМ с другими программами, а также не может использоваться для разработки программы для ЭВМ, по своему виду существенно схожей с декомпилируемой программой для ЭВМ, или для осуществления любого другого действия, нарушающего авторское право.

3. Применение положений настоящей статьи не должно наносить неоправданного ущерба нормальному использованию программы для ЭВМ или базы данных и не должно ущемлять необоснованным образом законные интересы автора или иного обладателя исключительных прав на программу для ЭВМ или базу данных.

Статья 26. Воспроизведение произведения в личных целях без согласия автора с выплатой авторского вознаграждения

1. В изъятие из положений ст. 37 и 38 настоящего Закона допускается без согласия автора произведения, исполнителя и производителя фонограммы, но с выплатой им вознаграждения воспроизведение аудиовизуального произведения или звукозаписи произведения исключительно в личных целях.

2. Вознаграждение за воспроизведение, указанное в п. 1 настоящей статьи, выплачивается изготовителями или импортерами оборудования (аудио- и видеомagnetофоны, иное оборудование) и материальных носителей (звуко- и (или) видеопленки и кассеты, лазерные диски, компакт-диски, иные материальные носители), используемых для такого воспроизведения.

Сбор и распределение этого вознаграждения осуществляются одной из организаций, управляющих имущественными правами авторов, производителей фонограмм и исполнителей на коллективной основе, в соответствии с соглашением между этими организациями (ст. 44 настоящего Закона). Если этим соглашением не предусмотрено иное, указанное вознаграждение распределяется в следующей пропорции: сорок процентов — авторам, тридцать процентов — исполнителям, тридцать процентов — производителям фонограмм.

Размер вознаграждения и условия его выплаты определяются соглашением между указанными изготовителями и импортерами, с одной стороны, и организациями, управляющими имущественными правами авторов, производителей фонограмм и исполнителей на коллективной основе, с другой стороны, а в случае, если стороны не достигнут такого соглашения, — специально уполномоченным органом Российской Федерации.

3. Вознаграждение не выплачивается применительно к оборудованию и материальным носителям, указанным в абзаце первом п. 2 настоящей статьи, которые являются предметом экспорта, а также к профессиональному оборудованию, не предназначенному для использования в домашних условиях.

Статья 27. Срок действия авторского права

1. Авторское право действует в течение всей жизни автора и 70 лет после его смерти, кроме случаев, предусмотренных настоящей статьей.

Право авторства, право на имя и право на защиту репутации автора охраняются бессрочно.

2. Автор вправе в том же порядке, в каком назначается исполнитель завещания, указать лицо, на которое он возлагает охрану права авторства, права на имя и права на защиту своей репутации после своей смерти. Это лицо осуществляет свои полномочия пожизненно.

При отсутствии таких указаний охрана права авторства, права на имя и права на защиту репутации автора после его смерти осуществляется его наследниками или специально уполномоченным органом Российской Федерации, который осуществляет такую охрану, если наследников нет или их авторское право прекратилось.

3. Авторское право на произведение, обнародованное анонимно или под псевдонимом, действует в течение 70 лет после даты его правомерного обнародования.

Если в течение указанного срока автор произведения, выпущенного анонимно или под псевдонимом, раскроет свою личность или его личность не будет далее оставлять сомнений, то применяется положение абзаца первого п. 1 настоящей статьи.

4. Авторское право на произведение, созданное в соавторстве, действует в течение всей жизни и 70 лет после смерти последнего автора, пережившего других соавторов.
5. Авторское право на произведение, впервые выпущенное в свет после смерти автора, действует в течение 70 лет после его выпуска.

В случае, если автор был репрессирован и реабилитирован посмертно, то срок охраны прав, предусмотренный настоящей статьей, начинается с 1 января года, следующего за годом реабилитации.

В случае, если автор работал во время Великой Отечественной войны или участвовал в ней, то срок охраны авторских прав, предусмотренный настоящей статьей, увеличивается на 4 года.

6. Исчисление сроков, предусмотренных настоящей статьей, начинается с 1 января года, следующего за годом, в котором имел место юридический факт, являющийся основанием для начала течения срока.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

Статья 28. Общественное достояние

1. Истечение срока действия авторского права на произведения означает их переход в общественное достояние.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ — исключен второй абзац данного пункта)

2. Произведения, перешедшие в общественное достояние, могут свободно использоваться любым лицом без выплаты авторского вознаграждения. При этом должно соблюдаться право авторства, право на имя и право на защиту репутации автора (ст. 15 настоящего Закона).
3. Правительством Российской Федерации могут устанавливаться случаи выплаты специальных отчислений за использование на территории Российской Федерации произведений, перешедших в общественное достояние. Такие отчисления выплачиваются в профессиональные фонды авторов, а также организациям, управляющим имущественными правами авторов на коллективной основе, и не могут превышать одного процента от прибыли, полученной за использование таких произведений.

Статья 29. Переход авторского права по наследству

Авторское право переходит по наследству.

Не переходит по наследству право авторства, право на имя и право на защиту репутации автора произведения. Наследники автора вправе осуществлять защиту указанных прав. Эти правомочия наследников сроком не ограничиваются.

При отсутствии наследников автора защиту указанных прав осуществляет специально уполномоченный орган Российской Федерации.

Статья 30. Передача имущественных прав. Авторский договор

1. Имущественные права, указанные в ст. 16 настоящего Закона, могут передаваться только по авторскому договору, за исключением случаев, предусмотренных ст. 18—26 настоящего Закона.

Передача имущественных прав может осуществляться на основе авторского договора о передаче исключительных прав или на основе авторского договора о передаче неисключительных прав.

2. Авторский договор о передаче исключительных прав разрешает использование произведения определенным способом и в установленных договором пределах только лицу, которому эти права передаются, и дает такому лицу право запрещать подобное использование произведения другим лицам.

Право запрещать использование произведения другим лицам может осуществляться автором произведения, если лицо, которому переданы исключительные права, не осуществляет защиту этого права.

3. Авторский договор о передаче неисключительных прав разрешает пользователю использование произведения наравне с обладателем исключительных прав, передавшим такие права, и (или) другим лицам, получившим разрешение на использование этого произведения таким же способом.
4. Права, передаваемые по авторскому договору, считаются неисключительными, если в договоре прямо не предусмотрено иное.

Примечание

В ст. 30 прямо указано, что содержание любого авторского договора — это передача имущественных прав, и ничего более. Причем передать вы можете как исключительные, так и неисключительные авторские права.

Примечание

Согласно п. 2 настоящей статьи, лицо, юридическое или физическое, которому передаются исключительные авторские права, не вправе передать их кому-

либо еще. То есть если, как мы уже говорили, права на служебное произведение, скажем, сайт, принадлежат работодателю, и он может делать с сайтом что хочет, кроме одного: он не может дать такое же право полноправного распоряжения сайтом другим лицам. Однако этим лицам он может передать неисключительные авторские права.

Статья 31. Условия авторского договора

1. Авторский договор должен предусматривать: способы использования произведения (конкретные права, передаваемые по данному договору); срок и территорию, на которые передается право; размер вознаграждения и (или) порядок определения размера вознаграждения за каждый способ использования произведения, порядок и сроки его выплаты, а также другие условия, которые стороны сочтут существенными для данного договора.

При отсутствии в авторском договоре условия о сроке, на который передается право, договор может быть расторгнут автором по истечении пяти лет с даты его заключения, если пользователь будет письменно уведомлен об этом за шесть месяцев до расторжения договора.

При отсутствии в авторском договоре условия о территории, на которую передается право, действие передаваемого по договору права ограничивается территорией Российской Федерации.

2. Все права на использование произведения, прямо не переданные по авторскому договору, считаются не переданными.

Предметом авторского договора не могут быть права на использование произведения, неизвестные на момент заключения договора.

3. Вознаграждение определяется в авторском договоре в виде процента от дохода за соответствующий способ использования произведения или, если это невозможно осуществить в связи с характером произведения или особенностями его использования, в виде зафиксированной в договоре суммы либо иным образом.

Минимальные ставки авторского вознаграждения устанавливаются Советом Министров — Правительством Российской Федерации. Минимальные размеры авторского вознаграждения индексируются одновременно с индексацией минимальных размеров заработной платы.

Если в авторском договоре об издании или ином воспроизведении произведения вознаграждение определяется в виде фиксированной суммы, то в договоре должен быть установлен максимальный тираж произведения.

4. Права, переданные по авторскому договору, могут передаваться полностью или частично другим лицам лишь в случае, если это прямо предусмотрено договором.

5. Предметом авторского договора не могут быть права на использование произведений, которые автор может создать в будущем.
6. Условие авторского договора, ограничивающее автора в создании в будущем произведений на данную тему или в данной области, является недействительным.
7. Условия авторского договора, противоречащие положениям настоящего Закона, являются недействительными.

Статья 32. Форма авторского договора

1. Авторский договор должен быть заключен в письменной форме. Авторский договор об использовании произведения в периодической печати может быть заключен в устной форме.
2. При продаже экземпляров программ для ЭВМ и баз данных и предоставлении массовым пользователям доступа к ним допускается применение особого порядка заключения договоров, установленного Законом Российской Федерации "О правовой охране программ для электронных вычислительных машин и баз данных".

Примечание

Это достаточно важный момент, заключающийся в том, что по отношению к программам (и только к ним!) на основании п. 3 ст. 14 Закона РФ "О правовой охране программ для электронно-вычислительных машин и баз данных" от 23.09.1992 допускается применение особого порядка заключения договоров. Суть этого "особого порядка" в том, что, получая экземпляр материального носителя программы или базы данных, покупатель считается подписавшим договор, суть которого изложена на этом экземпляре. То есть специальных "бумажных" договоров в этом случае не требуется.

Статья 33. Авторский договор заказа

1. По авторскому договору заказа автор обязуется создать произведение в соответствии с условиями договора и передать его заказчику.
2. Заказчик обязан в счет обусловленного договором вознаграждения выплатить автору аванс. Размер, порядок и сроки выплаты аванса устанавливаются в договоре по соглашению сторон.

Статья 34. Ответственность по авторскому договору

1. Сторона, не исполнившая или ненадлежащим образом исполнившая обязательства по авторскому договору, обязана возместить убытки, причиненные другой стороне, включая упущенную выгоду.

2. Если автор не представил заказное произведение в соответствии с условиями договора заказа, он обязан возместить реальный ущерб, причиненный заказчику.

Раздел III. СМЕЖНЫЕ ПРАВА

Примечание

Закон говорит о том, что существует четыре категории смежных прав: права исполнителей на их исполнения и постановки (исполнительские права); права производителей фонограмм на их фонограммы (фонограммные права), права организаций эфирного вещания на их передачи в эфир, права организаций кабельного вещания на их передачи для всеобщего сведения. Статья 36 говорит о том, что существуют три субъекта смежных прав: исполнители, производители фонограмм, организации эфирного и кабельного вещания. При этом исполнителями могут быть только физические лица, а производителями фонограмм и организаторами вещания как физические, так и юридические лица. Поскольку разработчики программных продуктов редко сталкиваются со смежными правами (исключение — создатели различных презентаций, видеороликов и т. д.), этот раздел Закона мы оставляем без дальнейших комментариев.

Статья 35. Сфера действия смежных прав

1. Права исполнителя признаются за ним в соответствии с настоящим Законом в случаях, если:
 - 1) исполнитель является гражданином Российской Федерации;
 - 2) исполнение, постановка впервые имели место на территории Российской Федерации;
 - 3) исполнение, постановка записаны на фонограмму, охраняемую в соответствии с положениями п. 2 настоящей статьи;
 - 4) исполнение, постановка, не записанные на фонограмму, включены в передачу в эфир или по кабелю, охраняемую в соответствии с положением п. 3 настоящей статьи.
2. Права производителя фонограммы признаются за ним в соответствии с настоящим Законом в случаях, если:
 - 1) производитель фонограммы является гражданином Российской Федерации или юридическим лицом, имеющим официальное местонахождение на территории Российской Федерации;
 - 2) фонограмма впервые опубликована на территории Российской Федерации.
3. Права организации эфирного или кабельного вещания признаются за ней в соответствии с настоящим Законом в случае, если организация

имеет официальное местонахождение на территории Российской Федерации и осуществляет передачи с помощью передатчиков, расположенных на территории Российской Федерации.

4. Предоставление на территории Российской Федерации охраны в соответствии с настоящим Законом объектам смежных прав иностранных физических и юридических лиц на основании международных договоров Российской Федерации осуществляется в отношении соответствующих исполнения, фонограммы, передачи в эфир, передачи по кабелю, не перешедших в общественное достояние в стране их происхождения вследствие истечения установленного в такой стране срока действия смежных прав и не перешедших в общественное достояние в Российской Федерации вследствие истечения предусмотренного настоящим Законом срока действия смежных прав.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

Статья 36. Субъекты смежных прав

1. Субъектами смежных прав являются исполнители, производители фонограмм, организации эфирного или кабельного вещания.
2. Производитель фонограммы, организация эфирного или кабельного вещания осуществляют свои права, указанные в настоящем разделе, в пределах прав, полученных по договору с исполнителем и автором записанного на фонограмме или передаваемого в эфир или по кабелю произведения.

Разрешение на использование постановки, полученное от режиссера-постановщика спектакля, не отменяет необходимости получения разрешения у других исполнителей, участвующих в постановке, а также у автора исполняемого произведения.

3. Исполнитель осуществляет указанные в настоящем разделе права при условии соблюдения прав автора исполняемого произведения.
4. Для возникновения и осуществления смежных прав не требуется соблюдения каких-либо формальностей. Производитель фонограммы и исполнитель для оповещения о своих правах вправе использовать знак охраны смежных прав, который помещается на каждом экземпляре фонограммы и (или) на каждом содержащем ее футляре и состоит из трех элементов:

- латинской буквы "P" в окружности: ©;
- имени (наименования) обладателя исключительных смежных прав;
- года первого опубликования фонограммы.

Статья 37. Права исполнителя

1. Кроме предусмотренных настоящим Законом случаев, исполнителю в отношении его исполнения или постановки принадлежат следующие исключительные права:
 - право на имя;
 - право на защиту исполнения или постановки от всякого искажения или иного посягательства, способного нанести ущерб чести и достоинству исполнителя;
 - право на использование исполнения или постановки в любой форме, включая право на получение вознаграждения за каждый вид использования исполнения или постановки.
2. Исключительное право на использование исполнения или постановки означает право осуществлять или разрешать осуществлять следующие действия:
 - 1) передавать в эфир или сообщать для всеобщего сведения по кабелю исполнение или постановку, если используемые для такой передачи исполнение или постановка не были ранее переданы в эфир или не осуществляются с использованием записи;
 - 2) записывать ранее не записанные исполнение или постановку;
 - 3) воспроизводить запись исполнения или постановки;
 - 4) передавать в эфир или по кабелю запись исполнения или постановки, если первоначально эта запись была произведена не для коммерческих целей;
 - 5) сдавать в прокат опубликованную в коммерческих целях фонограмму, на которой записаны исполнение или постановка с участием исполнителя. Это право при заключении договора на запись исполнения или постановки на фонограмму переходит к производителю фонограммы; при этом исполнитель сохраняет право на вознаграждение за сдачу в прокат экземпляров такой фонограммы (ст. 39 настоящего Закона);
 - 6) сообщать запись исполнения или постановки для всеобщего сведения таким образом, при котором любое лицо может иметь доступ к ней в интерактивном режиме из любого места и в любое время по своему выбору (право на доведение до всеобщего сведения).

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)
3. Исключительное право исполнителя, предусмотренное подп. 3 п. 2 настоящей статьи, не распространяется на случаи, когда:
 - 1) первоначальная запись исполнения или постановки была произведена с согласия исполнителя;

- 2) воспроизведение исполнения или постановки осуществляется в тех же целях, для которых было получено согласие исполнителя при записи исполнения или постановки;
- 3) воспроизведение исполнения или постановки осуществляется в тех же целях, для которых была произведена запись в соответствии с положениями ст. 42 настоящего Закона.
4. Разрешения, указанные в п. 2 настоящей статьи, выдаются исполнителем, а при исполнении коллективом исполнителей — руководителем такого коллектива посредством заключения письменного договора с пользователем.
5. Разрешения, указанные в подп. 1, 2 и 3 п. 2 настоящей статьи, на последующие передачи исполнения или постановки, осуществление записи для передачи и воспроизведения такой записи организациями эфирного или кабельного вещания не требуются, если они прямо предусмотрены договором исполнителя с организацией эфирного или кабельного вещания. Размер вознаграждения исполнителю за такое использование также устанавливается в этом договоре.
6. Заключение договора между исполнителем и изготовителем аудиовизуального произведения на создание аудиовизуального произведения влечет за собой предоставление исполнителем прав, указанных в подп. 1, 2, 3 и 4 п. 2 настоящей статьи.
Предоставление исполнителем таких прав ограничивается использованием аудиовизуального произведения и, если в договоре не установлено иное, не включает прав на отдельное использование звука или изображения, зафиксированных в аудиовизуальном произведении.
7. Исключительные права исполнителя, предусмотренные п. 2 настоящей статьи, могут передаваться по договору другим лицам.

Статья 38. Права производителя фонограммы

1. Кроме предусмотренных настоящим Законом случаев, производителю фонограммы в отношении его фонограммы принадлежат исключительные права на использование фонограммы в любой форме, включая право на получение вознаграждения за каждый вид использования фонограммы.
2. Исключительное право на использование фонограммы означает право осуществлять или разрешать осуществлять следующие действия:
 - 1) воспроизводить фонограмму;
 - 2) переделывать или любым иным способом перерабатывать фонограмму;
 - 3) распространять экземпляры фонограммы, т. е. есть продавать, сдавать их в прокат и т. д.;

4) импортировать экземпляры фонограммы в целях распространения, включая экземпляры, изготовленные с разрешения производителя этой фонограммы;

5) сообщать фонограмму для всеобщего сведения таким образом, при котором любое лицо может иметь доступ к ней в интерактивном режиме из любого места и в любое время по своему выбору (право на доведение до всеобщего сведения).

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

3. Если экземпляры правомерно опубликованной фонограммы введены в гражданский оборот посредством их продажи, то допускается их дальнейшее распространение без согласия производителя фонограммы и без выплаты вознаграждения.

Право на распространение экземпляров фонограммы путем сдачи их в прокат принадлежит производителю фонограммы независимо от права собственности на эти экземпляры.

4. Исключительные права производителя фонограммы, предусмотренные п. 2 настоящей статьи, могут передаваться по договору другим лицам.

**Статья 39. Использование фонограммы,
опубликованной в коммерческих целях,
без согласия производителя фонограммы и исполнителя**

1. В изъятие из положений ст. 37 и 38 настоящего Закона допускается без согласия производителя фонограммы, опубликованной в коммерческих целях, и исполнителя, исполнение которого записано на такой фонограмме, но с выплатой вознаграждения:

- 1) публичное исполнение фонограммы;
- 2) передача фонограммы в эфир;
- 3) сообщение фонограммы для всеобщего сведения по кабелю.

1.1. Положения п. 1 настоящей статьи не распространяются на доведение до всеобщего сведения фонограммы.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

2. Сбор, распределение и выплата вознаграждения, предусмотренного п. 1 настоящей статьи, осуществляются одной из организаций, управляющих правами производителей фонограмм и исполнителей на коллективной основе (ст. 44 настоящего Закона), в соответствии с соглашением между этими организациями. Если этим соглашением не предусмотрено иное, указанное вознаграждение распределяется между производителем фонограммы и исполнителем поровну.

3. Размер вознаграждения и условия его выплаты определяются соглашением между пользователем фонограммы или объединениями (ассоциациями) таких пользователей, с одной стороны, и организациями, управляющими правами производителей фонограмм и исполнителей, с другой стороны, а в случае, если стороны не достигнут такого соглашения, — специально уполномоченным органом Российской Федерации.
Размер вознаграждения устанавливается за каждый вид использования фонограммы.
4. Пользователи фонограмм должны предоставлять организации, указанной в п. 2 настоящей статьи, программы, содержащие точные сведения о количестве использований фонограммы, а также иные сведения и документы, необходимые для сбора и распределения вознаграждения.

Статья 40. Права организаций эфирного вещания

1. Кроме предусмотренных настоящим Законом случаев, организации эфирного вещания в отношении ее передачи принадлежат исключительные права использовать передачу в любой форме и давать разрешение на использование передачи, включая право на получение вознаграждения за предоставление такого разрешения.
2. Исключительное право давать разрешение на использование передачи означает право организации эфирного вещания разрешать осуществление следующих действий:
 - 1) одновременно передавать в эфир ее передачу другой организации эфирного вещания;
 - 2) сообщать передачу для всеобщего сведения по кабелю;
 - 3) записывать передачу;
 - 4) воспроизводить запись передачи;
 - 5) сообщать передачу для всеобщего сведения в местах с платным входом.
3. Исключительное право организации эфирного вещания, предусмотренное подп. 4 п. 2 настоящей статьи, не распространяется на случаи, когда:
 - запись передачи была произведена с согласия организации эфирного вещания;
 - воспроизведение передачи осуществляется в тех же целях, в которых была произведена ее запись в соответствии с положениями ст. 42 настоящего Закона.

Статья 41. Права организации кабельного вещания

1. Кроме предусмотренных настоящим Законом случаев, организации кабельного вещания в отношении ее передачи принадлежат исключительные права использовать передачу в любой форме и давать разрешение на использование передачи, включая право на получение вознаграждения за предоставление такого разрешения.
2. Исключительное право давать разрешение на использование передачи означает право организации кабельного вещания разрешать осуществление следующих действий:
 - 1) одновременно сообщать для всеобщего сведения по кабелю ее передачу другой организации кабельного вещания;
 - 2) передавать передачу в эфир;
 - 3) записывать передачу;
 - 4) воспроизводить запись передачи;
 - 5) сообщать передачу для всеобщего сведения в местах с платным входом.
3. Исключительное право организации кабельного вещания, предусмотренное подп. 4 п. 2 настоящей статьи, не распространяется на случай, когда:
 - запись передачи была произведена с согласия организации кабельного вещания;
 - воспроизведение передачи осуществляется в тех же целях, в которых была произведена ее запись в соответствии с положениями ст. 42 настоящего Закона.

Статья 42. Ограничения прав исполнителя, производителя фонограммы, организации эфирного или кабельного вещания

1. В изъятие из положений ст. 37—41 настоящего Закона допускаются без согласия исполнителя, производителя фонограммы, организации эфирного или кабельного вещания и без выплаты вознаграждения использование исполнения, постановки, передачи в эфир, передачи по кабелю и их записей, а также воспроизведение фонограмм:
 - 1) для включения в обзор о текущих событиях небольших отрывков из исполнения, постановки, фонограммы, передачи в эфир или по кабелю;
 - 2) исключительно в целях обучения или научного исследования;
 - 3) для цитирования в форме небольших отрывков из исполнения, постановки, фонограммы, передачи в эфир или по кабелю при условии, что такое цитирование осуществляется в информационных целях. При этом любое использование организацией эфирного или кабельного вещания

экземпляров фонограммы, опубликованной в коммерческих целях, для передачи в эфир или по кабелю возможно только при соблюдении положений ст. 39 настоящего Закона;

4) в иных случаях, которые установлены положениями раздела II настоящего Закона в отношении ограничения имущественных прав автора произведений литературы, науки и искусства.

2. В изъятие из положений ст. 37—41 настоящего Закона допускаются без согласия исполнителя, производителя фонограммы, организации эфирного или кабельного вещания использование передачи в эфир или по кабелю и ее записи, а также воспроизведение фонограммы в личных целях. Воспроизведение фонограммы допускается при условии выплаты вознаграждения в соответствии со ст. 26 настоящего Закона.
3. Не применяются положения ст. 37, 38, 40 и 41 настоящего Закона в отношении получения разрешения исполнителя, производителя фонограммы и организации эфирного вещания на осуществление записей краткосрочного пользования исполнения, постановки или передачи, на воспроизведение таких записей и на воспроизведение фонограммы, опубликованной в коммерческих целях, если запись краткосрочного пользования или воспроизведение осуществляются организацией эфирного вещания с помощью ее собственного оборудования и для ее собственной передачи при условии:
 - 1) предварительного получения организацией эфирного вещания разрешения на передачу в эфир самой постановки, исполнения или передачи, в отношении которых в соответствии с положениями настоящего пункта осуществляется запись краткосрочного пользования или воспроизведение такой записи;
 - 2) ее уничтожения в пределах срока, который устанавливается в отношении записи краткосрочного пользования произведений литературы, науки и искусства, производимой организацией эфирного вещания в соответствии с положениями ст. 24 настоящего Закона, за исключением единственного экземпляра, который может быть сохранен в официальных архивах на основании его исключительно документального характера.
4. Ограничения, предусмотренные настоящей статьей, применяются без ущерба нормальному использованию фонограммы, исполнения, постановки, передачи в эфир или по кабелю и их записей, а также включенных в них произведений литературы, науки и искусства и без ущемления законных интересов исполнителя, производителя фонограммы, организации эфирного или кабельного вещания и авторов указанных произведений.

Статья 43. Срок действия смежных прав

1. Права, предусмотренные настоящим разделом в отношении исполнителя, действуют в течение 50 лет после первого исполнения или постановки.

Права исполнителя на имя и на защиту исполнения или постановки от всякого искажения или иного посяательства, установленные ст. 37 настоящего Закона, охраняются бессрочно.

2. Права, предусмотренные настоящим разделом в отношении производителя фонограммы, действуют в течение 50 лет после первого опубликования фонограммы либо в течение 50 лет после ее первой записи, если фонограмма не была опубликована в течение этого срока.
3. Права, предусмотренные настоящим разделом в отношении организации эфирного вещания, действуют в течение 50 лет после осуществления такой организацией первой передачи в эфир.
4. Права, предусмотренные настоящим разделом в отношении организации кабельного вещания, действуют в течение 50 лет после осуществления такой организацией первой передачи по кабелю.
5. Исчисление сроков, предусмотренных п. 1, 2, 3 и 4 настоящей статьи, начинается с 1 января года, следующего за годом, в котором имел место юридический факт, являющийся основанием для начала течения срока.

6. В случае, если исполнитель был репрессирован и реабилитирован по-смертно, то срок охраны прав, предусмотренный настоящей статьей, начинает действовать с 1 января года, следующего за годом реабилитации.

В случае, если исполнитель работал во время Великой Отечественной войны или участвовал в ней, то срок охраны прав, предусмотренный настоящей статьей, увеличивается на 4 года.

7. К наследникам (в отношении юридических лиц — к правопреемникам) исполнителя, производителя фонограммы, организации эфирного или кабельного вещания переходит право разрешать использование исполнения, постановки, фонограммы, передачи в эфир или по кабелю и на получение вознаграждения в пределах оставшейся части сроков, указанных в п. 1, 2, 3 и 4 настоящей статьи.

Раздел IV. КОЛЛЕКТИВНОЕ УПРАВЛЕНИЕ ИМУЩЕСТВЕННЫМИ ПРАВАМИ

Статья 44. Цели коллективного управления имущественными правами

1. В целях обеспечения имущественных прав авторов, исполнителей, производителей фонограмм и иных обладателей авторских и смежных прав в случаях, когда их практическое осуществление в индивидуальном поряд-

ке затруднительно (публичное исполнение, в том числе на радио и телевидении, воспроизведение произведения путем механической, магнитной и иной записи, репродуцирование и другие случаи), могут создаваться организации, управляющие имущественными правами указанных лиц на коллективной основе.

2. Такие организации создаются непосредственно обладателями авторских и смежных прав и действуют в пределах полученных от них полномочий на основе устава, утверждаемого в порядке, установленном законодательством.
3. Допускается создание либо отдельных организаций по различным правам и различным категориям обладателей прав, либо организаций, управляющих разными правами в интересах разных категорий обладателей прав, либо одной организации, одновременно управляющей авторскими и смежными правами.

Статья 45. Организации, управляющие имущественными правами на коллективной основе

1. В соответствии с настоящим Законом организация, управляющая имущественными правами на коллективной основе, не вправе заниматься коммерческой деятельностью.

По отношению к деятельности такой организации не применяются ограничения, предусмотренные антимонопольным законодательством.

2. Полномочия на коллективное управление имущественными правами передаются непосредственно обладателями авторских и смежных прав добровольно на основе письменных договоров, а также по соответствующим договорам с иностранными организациями, управляющими аналогичными правами. Такие договоры не являются авторскими, и на них не распространяются положения ст. 30—34 настоящего Закона.

Любой автор, его наследник или иной обладатель авторских и смежных прав, охраняемых в соответствии с разделом III настоящего Закона, вправе передать по договору осуществление своих имущественных прав такой организации, а организация обязана принять на себя осуществление этих прав на коллективной основе, если управление такой категорией прав относится к уставной деятельности этой организации.

Указанные организации не вправе осуществлять использование произведений и объектов смежных прав, полученных для управления на коллективной основе.

3. На основе полномочий, полученных в соответствии с п. 2 настоящей статьи, организация, управляющая имущественными правами на коллективной основе, предоставляет лицензии пользователям на соответствующие способы использования произведений и объектов смежных прав. Условия

таких лицензий должны быть одинаковыми для всех пользователей одной категории. Указанные организации не вправе отказать в выдаче лицензии пользователю без достаточных на то оснований.

Такие лицензии разрешают использование предусмотренными в них способами всех произведений и объектов смежных прав и предоставляются от имени всех обладателей авторских и смежных прав, включая и тех, которые не передали организации полномочий в соответствии с п. 2 настоящей статьи.

Все возможные имущественные претензии обладателей авторских и смежных прав к пользователям, связанные с использованием их произведений и объектов смежных прав по таким лицензиям, должны быть урегулированы организацией, предоставляющей такие лицензии.

4. Организация, управляющая имущественными правами на коллективной основе, вправе сохранять невостребованное вознаграждение, включая его в распределяемые суммы или обращая на другие цели в интересах представляемых ею обладателей авторских и смежных прав по истечении трех лет с даты его поступления на счет организации.

Статья 46. Функции организаций, управляющих имущественными правами на коллективной основе

Организация, управляющая имущественными правами на коллективной основе, должна выполнять от имени представляемых ею обладателей авторских и смежных прав и на основе полученных от них полномочий следующие функции:

- 1) согласовывать с пользователями размер вознаграждения и другие условия, на которых выдаются лицензии;
- 2) предоставлять лицензии пользователям на использование прав, управлением которых занимается такая организация;
- 3) согласовывать с пользователями размер вознаграждения в тех случаях, когда эта организация занимается сбором такого вознаграждения без выдачи лицензии (п. 2 ст. 26, п. 2 и 3 ст. 39 настоящего Закона);
- 4) собирать предусмотренное лицензиями вознаграждение и (или) вознаграждение, предусмотренное п. 3 настоящей статьи;
- 5) распределять и выплачивать собранное в соответствии с п. 4 настоящей статьи вознаграждение представляемым ею обладателям авторских и смежных прав;
- 6) совершать любые юридические действия, необходимые для защиты прав, управлением которых занимается такая организация;
- 7) осуществлять иную деятельность в соответствии с полномочиями, полученными от обладателей авторских и смежных прав.

Статья 47. Обязанности организаций, управляющих имущественными правами на коллективной основе

1. Деятельность организации, управляющей имущественными правами на коллективной основе, осуществляется в интересах обладателей авторских и смежных прав, представляемых такой организацией. В этих целях организация должна выполнять следующие обязанности:
 - 1) одновременно с выплатой вознаграждения представлять обладателям авторских и смежных прав отчеты, содержащие сведения об использовании их прав;
 - 2) использовать собранное в соответствии с положениями п. 4 ст. 46 настоящего Закона вознаграждение исключительно для распределения и выплаты обладателям авторских и смежных прав. При этом организация вправе вычитать из собранного вознаграждения суммы на покрытие своих фактических расходов по сбору, распределению и выплате такого вознаграждения, а также суммы, которые направляются в специальные фонды, создаваемые этой организацией с согласия и в интересах представляемых ею обладателей авторских и смежных прав;
 - 3) распределять и регулярно выплачивать собранные суммы вознаграждения, за вычетом сумм, указанных в подп. 2 настоящего пункта, пропорционально фактическому использованию произведений и объектов смежных прав.
2. Обладатели авторских и смежных прав, не предоставившие полномочий организации в отношении сбора вознаграждения, предусмотренного в п. 4 ст. 46 настоящего Закона, вправе потребовать от организации выплатить причитающееся им вознаграждение в соответствии с произведенным распределением, а также исключить свои произведения и объекты смежных прав из лицензий, предоставляемых этой организацией пользователям.

Раздел V. ЗАЩИТА АВТОРСКИХ И СМЕЖНЫХ ПРАВ

Статья 48. Нарушение авторских и смежных прав.

Контрафактные экземпляры произведения и фонограммы

1. Незаконное использование произведений или объектов смежных прав либо иное нарушение предусмотренных настоящим Законом авторского права или смежных прав влечет за собой гражданско-правовую, административную, уголовную ответственность в соответствии с законодательством Российской Федерации.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

2. Физическое или юридическое лицо, которое не выполняет требований настоящего Закона, является нарушителем авторских и смежных прав.
3. Контрафактными являются экземпляры произведения и фонограммы, изготовление или распространение которых влечет за собой нарушение авторских и смежных прав.
4. Контрафактными являются также экземпляры охраняемых в Российской Федерации в соответствии с настоящим Законом произведений и фонограмм, импортируемые без согласия обладателей авторских и смежных прав в Российскую Федерацию из государства, в котором эти произведения и фонограммы никогда не охранялись или перестали охраняться.

Примечание

Как видим, согласно п.1 ст. 48, за нарушение авторских прав может наступать и уголовная ответственность. Согласно ст. 146 Уголовного кодекса Российской Федерации (с 01.01.1997), незаконное использование объектов авторского права, а также присвоение авторства, если эти деяния причинили крупный ущерб, наказываются штрафом от 200 до 400 МРОТ или в размере заработной платы или иного дохода осужденного за период от 2 до 4 месяцев, либо обязательными работами на срок от 180 до 240 часов, либо лишением свободы на срок до 2-х лет. Если же эти деяния совершены неоднократно, либо группой лиц по предварительному сговору или организованной группой, то размеры штрафов удваиваются, может быть применен арест на срок от 4 до 6 месяцев либо лишение свободы на срок до 5 лет.

Статья 48.1. Технические средства защиты авторского права и смежных прав

1. Техническими средствами защиты авторского права и смежных прав признаются любые технические устройства или их компоненты, контролирурующие доступ к произведениям или объектам смежных прав, предотвращающие либо ограничивающие осуществление действий, которые не разрешены автором, обладателем смежных прав или иным обладателем исключительных прав, в отношении произведений или объектов смежных прав.
2. В отношении произведений или объектов смежных прав не допускаются:
 - 1) осуществление без разрешения лиц, указанных в п. 1 настоящей статьи, действий, направленных на снятие ограничений использования произведений или объектов смежных прав, установленных путем применения технических средств защиты авторского права и смежных прав;
 - 2) изготовление, распространение, сдача в прокат, предоставление во временное безвозмездное пользование, импорт, реклама любого устрой-

ства или его компонентов, их использование в целях получения дохода либо оказание услуг в случаях, если в результате таких действий становится невозможным использование технических средств защиты авторского права и смежных прав либо эти технические средства не смогут обеспечить надлежащую защиту указанных прав.

(Федеральный закон от 20 июля 2004 г. № 72-ФЗ)

Статья 48.2. Информация об авторском праве и о смежных правах

1. Информацией об авторском праве и о смежных правах признается любая информация, которая идентифицирует произведение или объект смежных прав, автора, обладателя смежных прав или иного обладателя исключительных прав, либо информация об условиях использования произведения или объекта смежных прав, которая содержится на экземпляре произведения или объекта смежных прав, приложена к ним или появляется в связи с сообщением для всеобщего сведения либо доведением до всеобщего сведения таких произведений или объектов смежных прав, а также любые цифры и коды, в которых содержится такая информация.
2. В отношении произведений или объектов смежных прав не допускаются:
 - 1) удаление или изменение без разрешения лиц, указанных в п. 1 настоящей статьи, информации об авторском праве и о смежных правах;
 - 2) воспроизведение, распространение, импорт в целях распространения, публичное исполнение, сообщение для всеобщего сведения, доведение до всеобщего сведения произведений или объектов смежных прав, в отношении которых без разрешения лиц, указанных в п. 1 настоящей статьи, была удалена информация об авторском праве и о смежных правах.

(Федеральный закон от 20 июля 2004 г. № 72-ФЗ)

Статья 49. Гражданско-правовые способы защиты авторского права и смежных прав

1. Автор, обладатель смежных прав или иной обладатель исключительных прав вправе защищать свои права способами, предусмотренными Гражданским кодексом Российской Федерации.
2. Обладатели исключительных прав вправе требовать по своему выбору от нарушителя вместо возмещения убытков выплаты компенсации:
 - 1) в размере от 10 тысяч рублей до 5 миллионов рублей, определяемом по усмотрению суда, арбитражного суда или третейского суда исходя из характера нарушения;

2) в двукратном размере стоимости экземпляров произведений или объектов смежных прав либо в двукратном размере стоимости прав на использование произведений или объектов смежных прав, определяемой исходя из цены, которая при сравнимых обстоятельствах обычно взимается за правомерное использование произведений или объектов смежных прав.

Обладатели исключительных прав вправе требовать от нарушителя выплаты компенсации за каждый случай неправомерного использования произведений или объектов смежных прав либо за допущенные правонарушения в целом.

Компенсация подлежит взысканию при доказанности факта правонарушения независимо от наличия или отсутствия убытков.

1. Авторы и исполнители в случае нарушения их личных неимущественных прав или имущественных прав также вправе требовать от нарушителя возмещения морального вреда.
2. Автор, обладатель смежных прав или иной обладатель исключительных прав в установленном законом порядке вправе обратиться для защиты своих прав в суд, арбитражный суд, третейский суд, органы прокуратуры, органы дознания, органы предварительного следствия в соответствии с их компетенцией.
3. Организация, управляющая имущественными правами на коллективной основе, в порядке, установленном законом, вправе обращаться в суд от своего имени с заявлениями в защиту нарушенных авторских прав и (или) смежных прав лиц, управление имущественными правами которых осуществляется такой организацией.

(в редакции Федерального закона от 20 июля 2004 г. № 72-ФЗ)

Статья 49.1. Конфискация контрафактных экземпляров произведений или фонограмм

1. Контрафактные экземпляры произведений или фонограмм, а также материалы и оборудование, используемые для воспроизведения контрафактных экземпляров произведений или фонограмм, и иные орудия совершения правонарушения подлежат конфискации в судебном порядке в соответствии с законодательством Российской Федерации.
2. Конфискованные контрафактные экземпляры произведений или фонограмм подлежат уничтожению, за исключением случаев их передачи обладателю авторских прав или смежных прав по его просьбе.

(Федеральный закон от 20 июля 2004 г. № 72-ФЗ)

**Статья 50. Способы обеспечения иска по делам
о нарушении авторских и смежных прав**

1. Суд или судья единолично, а также арбитражный суд может вынести определение о запрещении ответчику либо лицу, в отношении которого имеются достаточные основания полагать, что оно является нарушителем авторских и смежных прав, совершать определенные действия (изготовление, воспроизведение, продажу, сдачу в прокат, импорт или иное предусмотренное настоящим Законом использование, а также транспортировку, хранение или владение с целью выпуска в гражданский оборот экземпляров произведений и фонограмм, в отношении которых предполагается, что они являются контрафактными).
2. Суд или судья единолично, а также арбитражный суд может вынести определение о наложении ареста и изъятии всех экземпляров произведений и фонограмм, в отношении которых предполагается, что они являются контрафактными, а также материалов и оборудования, предназначенных для их изготовления и воспроизведения.

При наличии достаточных данных о нарушении авторских или смежных прав орган дознания, следователь, суд или судья единолично обязаны принять меры для розыска и наложения ареста на экземпляры произведений или фонограмм, в отношении которых предполагается, что они являются контрафактными, а также на материалы и оборудование, предназначенные для изготовления и воспроизведения указанных экземпляров произведений или фонограмм, включая в необходимых случаях меры по их изъятию и передаче на ответственное хранение.

Президент
Российской Федерации
Б. Ельцин
Москва, Дом Советов России
9 июля 1993 года
№ 5351-I

Вопросы и ответы

Вопрос: Допустим, я выпустил свой скрипт. Как мне его защитить? В смысле прав.

Ответ:

Авторские права, причем и личные, и имущественные, принадлежат вам с момента создания вами произведения, к которому относится и скрипт. Вы можете специально ничего не предпринимать по защите авторских прав на ваш скрипт, так как закон автоматически защищает ваше произведение с момента его возникновения. Если вы хотите специально зарегистрировать ваш скрипт как объект авторского права, то можете обратиться либо к нотариусу, либо зарегистрировать ваше произведение в Российском Авторском Обществе (РАО).

Другое дело, что если вы разрабатывали скрипт не в тиши собственного кабинета, а сидя на рабочем месте в какой-то конторе, то имущественные права в этом случае автоматически принадлежат конторе, в том случае, если у вас есть с этой организацией трудовой договор.

Вопрос: Я написал скрипт. Надо ли мне где-то регистрировать свой продукт?

Ответ:

Если вы имеете в виду, нужно ли его сертифицировать, то — нет. Сертифицировать продукт приходится тогда, когда он применяется, скажем, в оборонке, в банковском деле, на серьезных промышленных объектах и т. д. И то — если банк или государство в лице какой-нибудь ВЧ ваш продукт купит без сертификации — то вас это совершенно не касается, и никакой ответственности вы за это не понесете. Просто в тех местах, где на ПО нужен сертификат, у вас без сертификата его не купят. Но РНР-скрипты обычно в таких "серьезных" делах не используются. Я знаю только 3 случая, когда понадобилась сертификация РНР-скриптов.

Вопрос: Здравствуйте, мне интересен следующий вопрос: как правильно заимствовать материалы другого сайта (изображения или контент), чтобы избежать недоразумений? Часто владелец сайта требует поставить ссылку на свой ресурс — здесь все понятно. Материалы сайта публикуются под лицензией GPL — ясно. На сайте-источнике имеется предупреждающее сообщение с запретом копировать материалы — яснее некуда. А что делать, если никаких сообщений и предостережений нет? Существуют ли какие-нибудь неписанные (или писанные) правила на этот счет?

Ответ:

В любом случае данные материалы (как почти любые, за некоторыми небольшими исключениями) однозначно защищаются законом об авторском праве. Это значит, что в суде вы в случае чего проиграете, если что-то скопируете с чужого сайта без указания авторства. Даже если автор сам напишет, "копируйте, берите и используйте где хотите", а потом вдруг подаст в суд — суд будет на его стороне. Потому что в любом случае нарушается авторское право. Таким образом, если что-то надо скопировать, во избежание недоразумений в любом случае нужно указывать авторство. Исключениями (то, на что не распространяется авторское право), к примеру, являются, государственные законы, сообщения о фактах (типа "Этот дом стоит на улице такой-то") и т. д.

Кроме авторских прав есть еще имущественные права. К примеру, авторские права на наши книги принадлежат нам (авторам), а имущественные права — издательству. Мы передаем ему эти имущественные права, подписывая авторский договор заказа. В том случае, когда договоров о передаче имущественных прав не подписывается, они принадлежат тем же лицам, которым принадлежит авторство. Но дело в том, что авторские права не предусматривают коммерческого использования произведения, а имущественные предусматривают. То есть при нарушении имущественных прав нарушителю вполне можно впаять выплату "упущенной выгоды", размер которой определяет суд. В общем, лучше либо все делать самому, либо честно сослаться на "субъектов авторских прав".

Вопрос: Дело вот в чем: у меня в голове есть несколько проектов сайтов историко-географической направленности. Текстовое содержание я сделаю сам, а вот с иллюстрациями будут проблемы. Допустим, мне понравилась какая-нибудь фотография, скорее всего, любительская (скорее всего, она будет размещена во франко/австро/германоязычном домене), могу я безболезненно ее разместить на своем сайте, указав сайт-источник? Не окажусь ли я в положении нарушителя закона об авторском праве?

Ответ:

С юридической точки зрения, скорее всего, все будет нормально на основании ст. 19 закона об авторском праве и смежных правах (далее — АП и СП), который говорит о праве на цитирование без выплаты авторского вознаграждения, но с обязательным указанием имени автора. То есть цитировать произведения можно, но с обязательным указанием автора произведения, из которого взята цитата. То есть если вы берете материал с сайта, на котором фотография, вы должны указать и имя автора фотографии, и "взято с сайта ...". Почему надо указывать первое — понятно, а второе — по двум причинам.

Во-первых, авторы сайта сами могут быть субъектами авторского права — к примеру, они в вашем случае могут иметь "право составителя" (ст. 11 закона об АП и СП). (По этой причине в ссылках на сборники всегда указывается либо составитель, либо под чьей редакцией это было сделано). Во-вторых, согласно ст. 19 закона об АП и СП цитировать можно "с обязательным указанием имени автора, произведение которого используется, и источника заимствования". Имя автора вы можете не указывать только в том случае, если оно неизвестно. Тогда указывается только "источник заимствования" — в нашем случае это сайт.

Теперь немного относительно цитирования. Тут закон более хитер и сказано, что цитировать можно "в объеме, оправданном целью цитирования".