

Стивен К. СЕОВ

# ПРОЕКТИРУЕМ ВРЕМЯ

ПСИХОЛОГИЯ ВОСПРИЯТИЯ ВРЕМЕНИ  
В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ



По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 978-5-93286-142-4, название «Проектируем время. Психология восприятия времени в программном обеспечении» – покупка в Интернет-магазине «Books.Ru – Книги России».

Если Вы получили данный файл каким-либо другим образом, Вы нарушили международное законодательство и законодательство Российской Федерации об охране авторского права. Вам необходимо удалить данный файл, а также сообщить издательству «Символ-Плюс» ([piracy@symbol.ru](mailto:piracy@symbol.ru)), где именно Вы получили данный файл.

# **Designing and Engineering Time**

**The Psychology of Time Perception in Software**

*Steven C. Seow*

# Проектируем время

Психология восприятия времени  
в программном обеспечении

*Стивен К. Сеов*



*Санкт-Петербург — Москва  
2009*

Серия «Профессионально»

Стивен К. Сеов

# Проектируем время. Психология восприятия времени в программном обеспечении

Перевод Н. Подольской

Главный редактор	<i>А. Галунов</i>
Зав. редакцией	<i>Н. Макарова</i>
Выпускающий редактор	<i>Л. Пискунова</i>
Научный редактор	<i>С. Миронов</i>
Редактор	<i>Е. Тульсанова</i>
Корректор	<i>Е. Бекназарова</i>
Верстка	<i>Д. Белова</i>

*Сеов С.*

Проектируем время. Психология восприятия времени в программном обеспечении. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 224 с., ил.

ISBN: 978-5-93286-142-4

Вряд ли найдется человек, которому не было бы знакомо состояние мучительного ожидания при работе на компьютере. Долгие процессы установки программ, загрузки обновлений, антивирусного сканирования... Все понимают, что сократить это время технически невозможно, но есть другие способы изменения ситуации. О них и рассказывается в этой книге. Автор подробно рассматривает особенности восприятия времени человеком и предлагает методы и приемы, способствующие повышению пользовательской толерантности и удовлетворенности. Предлагаемые методики позволяют не только вызвать у пользователя ощущение ускорения длительных операций, но и заставить его наслаждаться временем, потраченным на ожидание их выполнения.

ISBN: 978-5-93286-142-4

ISBN: 978-0-321-50918-5 (англ)

© Издательство Символ-Плюс, 2009

Authorized translation of the English edition, entitled DESIGNING AND ENGINEERING TIME: THE PSYCHOLOGY OF TIME PERCEPTION IN SOFTWARE, ISBN 0-321-50918-8, by STEVEN C. SEOW, PH.D., published by Pearson Education, Inc. Copyright © 2008. This translation is published and sold by permission of Pearson Education, Inc., the owner of all rights to publish and sell the same.

Все права на данное издание защищены Законодательством РФ, включая право на полное или частичное воспроизведение в любой форме. Все товарные знаки или зарегистрированные товарные знаки, упоминаемые в настоящем издании, являются собственностью соответствующих фирм.

Издательство «Символ-Плюс». 199034, Санкт-Петербург, 16 линия, 7, тел. (812) 3245353, www.symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Подписано в печать 25.02.2009. Формат 70x100 1/16. Печать офсетная.

Объем 14 печ. л. Тираж 1500 экз. Заказ N

Отпечатано с готовых диапозитивов в ГУП «Типография «Наука»  
199034, Санкт-Петербург, 9 линия, 12.

*Посвящается моей жене Джейм и сыну Алексу*



# Оглавление

<b>Отзывы о книге</b> .....	13
<b>Благодарности</b> .....	15
<b>Об авторе</b> .....	16
<b>Предисловие</b> .....	17
<b>Глава 1. Проектирование и конструирование времени</b> .....	21
Звучит привычно? .....	21
Странности времени .....	22
Можно ли конструировать время .....	24
Зачем проектировать и конструировать время .....	25
В отличие от денег, время переменчиво .....	27
Все стимулируется восприятием .....	28
Кто такой конструктор времени.....	29
Об этой книге .....	29
Организация глав .....	29
Терминология .....	31
Изображение взамен тысячи слов .....	32
Спуск в кроличью нору.....	32
Кроличья нора .....	32
Классические психологические опыты со временем.....	32
Содержательные статьи о восприятии времени .....	33
Научные исследования и манипулирование объективным временем .....	33
Книги о времени.....	33
Время и деньги.....	33
Восприятие времени и производительность .....	34
Восприятие времени и удовлетворение .....	34
Восприятие времени и использование компьютерных программ .....	34
Кроличьи норы .....	34

<b>Глава 2. Восприятие и толерантность</b> .....	<b>35</b>
Восприятие и толерантность.....	35
Действительность или ощущение? .....	37
Реальность: фактическое время .....	38
Точность: объективное время на основе объективных измерений .....	38
Данные: объем, изменчивость, разнообразие .....	39
Восприятие: психологическое время.....	39
Субъективность: видите ли вы то, что вижу я?.....	40
Искажение: не доверяйте своему мозгу .....	41
Толерантность: оцениваемое время.....	43
Количество или качество .....	43
Мысленный эталон .....	43
Временные факторы, влияющие на толерантность .....	45
Вневременные факторы, влияющие на толерантность.....	47
Выводы.....	51
Кроличья нора .....	51
Управление восприятием.....	51
Воспринимаемое время .....	52
Классические эксперименты с памятью .....	52
Пользовательская толерантность.....	52
<b>Глава 3. Время отклика пользователя и системы</b> .....	<b>55</b>
Беседа кремния с углеродом .....	55
Определение времени отклика.....	56
Диалог .....	58
Время отклика пользователя.....	59
Соотношение между скоростью и точностью.....	60
Время отклика системы .....	62
Промышленные стандарты на время отклика системы .....	64
Выводы.....	68
Кроличья нора .....	69
Время отклика человека.....	69
Время отклика системы .....	69
Соотношение скорость/точность.....	69
Законы Хика–Хаймана и Фиттса.....	69
Промышленные стандарты .....	70
<b>Глава 4. Реактивность</b> .....	<b>71</b>
Что такое реактивность.....	71
Реактивность определяется относительно взаимодействия .....	72
Задержка воспринимается субъективно.....	72
Язык тела воспринимается как ответ .....	73
Реактивность, основанная на ожиданиях пользователя.....	74
Мгновенная: от 0,1 до 0,2 секунды .....	75
Незамедлительная: от 0,5 до одной секунды .....	76

Не прерывающая: от двух до пяти секунд .....	78
Вынуждающая: от семи до десяти секунд.....	80
Слишком быстро? .....	81
Компьютер слишком скор.....	82
Пользователь слишком скор .....	84
Выводы.....	85
Кроличья нора .....	85
Микровыражения .....	85
Реактивность с точки зрения пользователя .....	85
Потоковое состояние пользователя .....	86
Объем внимания.....	86
<b>Глава 5. Обнаружение различий в распределении времени.....</b>	<b>87</b>
Объявление различия .....	87
D-уровни .....	89
D1: закон Вебера.....	90
Правило 20% .....	92
D2: деление надвое посредством среднего геометрического .....	95
Критерий типа «ненамного» .....	96
Выводы.....	98
Кроличья нора .....	99
Различия в распределении времени.....	99
Закон Вебера .....	99
Дифференциация .....	99
Деление надвое посредством среднего геометрического .....	99
<b>Глава 6. Индикация выполнения.....</b>	<b>101</b>
Ошибки кремния.....	101
Классификация индикации выполнения.....	102
Выбор правильного класса.....	105
Проектирование индикации выполнения.....	107
Вид отображения: текстуальный или визуальный.....	107
Единица продвижения выполнения: время или работа .....	111
Тип данных: количественный или качественный .....	116
Выводы.....	118
Кроличья нора .....	118
Виды индикации выполнения .....	118
Явление флуктуации времени .....	119
Последовательное раскрытие .....	119
<b>Глава 7. Представление времени.....</b>	<b>121</b>
Распределение во времени: прошлое, настоящее, будущее.....	121
Предварительно: Tickle-Me-Elmo .....	122
В режиме реального времени: пробный стикер.....	124
Ретроспективно: всегда худший случай .....	126

Разговор о времени .....	127
Якоря времени .....	128
Матрица якорей времени .....	129
Разговор о времени .....	130
Ситуации неопределенности .....	133
1. Предоставляйте информацию, не относящуюся ко времени ..	134
2. Таймеры и тайм-аут .....	134
Грамматика и этикет представления времени .....	135
1. Употребляйте единственное число для единичных значений .....	135
2. Ноль означает закончено! .....	135
3. Пользуйтесь единицами времени единообразно .....	136
4. Между X и Y .....	136
5. Избегайте двусмысленных фраз .....	136
Выводы .....	136
Кроличья нора .....	137
Как представлять время: предварительно или ретроспективно .....	137
Заниженная и завышенная оценки при предварительном представлении времени .....	137
Стили написания .....	137
Якоря и оценивание времени .....	137
<b>Глава 8. Потокное состояние пользователя .....</b>	<b>139</b>
Что такое потокное состояние пользователя .....	139
Потокное состояние пользователя в виде карты .....	140
Потокное состояние пользователя в виде траектории .....	141
Потокное состояние пользователя как описание его впечатления .....	142
Оптимизация потокного состояния пользователя .....	143
1. Соответствие между умениями и навыками и сложностью задачи .....	144
2. Цели и обратная связь .....	147
3. Ощущение контроля .....	149
Выводы .....	150
Кроличья нора .....	150
Психология потока .....	150
Потокное состояние пользователя в человеко-машинном взаимодействии .....	151
Потокное состояние пользователя и пользователи Интернета .....	151
Скорость чтения .....	151
<b>Глава 9. Тестирование времени .....</b>	<b>153</b>
Особенности тестирования времени .....	153
Получение надежной и достоверной информации .....	154
Выполнение замеров реальных длительностей .....	156
Тщательно определяйте реальные длительности .....	157
Выбирайте надлежащую точность .....	157

Выбирайте подходящий метод.....	159
Не используйте пользователей в качестве хронометров! .....	160
Измерение воспринимаемой длительности .....	160
Словесное оценивание .....	160
Репродуцирование.....	162
Настройка .....	162
Оценивание толерантности.....	163
Ожидаемая реактивность .....	163
Экспериментирование.....	164
Продуцирование .....	165
Метод кроссмодального сравнения.....	165
Кое-что о постановке эксперимента.....	166
Влияние последовательности: кто пришел первым .....	166
Влияние протяженности действия и тренированности пользователя .....	166
Держите кота в мешке!.....	167
Выводы.....	168
Кроличья нора .....	169
Методологии тестирования .....	169
Оценивание времени человеком.....	169
Инструментальные ограничения .....	169
<b>Глава 10. Методы.....</b>	<b>171</b>
Управление восприятием .....	171
Упреждающий запуск .....	172
Раннее завершение .....	173
Скрытая декомпоновка .....	174
Нисходящие длительности .....	175
Нелинейная индикация выполнения.....	176
Непрерывные длительности .....	178
Информирование.....	179
Целенаправленное отвлечение .....	180
Запустить-и-забыть .....	181
Управление толерантностью.....	183
Меньше обещать, больше предоставлять .....	183
Модель шкалы цен .....	184
Якоря времени.....	185
Ценность ожидания .....	186
Буферизация и предложение.....	188
Лишь в первый раз .....	189
Контекстные контрольные данные.....	190
Отсчет последних секунд .....	191
Выводы.....	192
Кроличья нора .....	193
Исследования, посвященные веб-сайтам .....	193
Ожидания и удовлетворение потребителей.....	193
Исследования и методы индустрии розничной торговли и обслуживания .....	193

---

<b>Глава 11. Нарушения</b> .....	195
Нарушение восприятия .....	195
Наблюдение за чайником .....	196
Не отпускающее ожидание .....	197
Негативная оценка .....	199
Затраченное время .....	200
Эффект Барнабуса .....	201
Перегруженность информацией .....	203
Фрагментированные длительности .....	204
Беспокойство .....	205
Нарушение толерантности .....	206
Неопределенность .....	207
Нарушение обещаний .....	208
Обещания компании кабельного телевидения .....	209
Сверхточность .....	209
Повторяющиеся циклы .....	210
Неожиданное дополнение .....	211
Задержка использования .....	213
Выводы .....	214
Кроличья нора .....	214
Оценивание с занижением и завышением .....	214
Ожидание и удовлетворенность потребителей .....	215
<b>Алфавитный указатель</b> .....	217

## ОТЗЫВЫ О КНИГЕ

Сеов написал замечательную книгу о понимании восприятия времени в практической деятельности пользователя и управлении им. Она понятная, занимательная и многогранная. Ее следует прочитать проектировщикам, разработчикам, а также всем тем, кто участвует в подготовке решений в области взаимодействия человека с компьютером.

*Сьюзен Ходжес Рамлет (Susan Hodges Ramlet),  
инженер по проектированию взаимодействий,  
член Ассоциации специалистов  
по удобству использования*

Это первое исчерпывающее руководство по очень важному аспекту удобства использования программного обеспечения. Множество реальных примеров и превосходных методов. Это будет понятно всем членам сообщества разработчиков программ и их спонсорам.

*Терренс Майкл Гардинер (Terrence Michael Gardiner),  
владелец TEAMstraps.com и главный консультант  
по дизайну, ориентированному на пользователя*

Если вы ищете книгу, объясняющую, почему пользователи так недовольны вашей программой, то вы ее нашли. Если вы хотите найти способы устранения этого недовольства, купите эту книгу.

*Тим Патрик (Tim Patrick),  
автор книги «Programming Visual Basic 2008»*

Время отклика – это характеристика, которая играет важнейшую роль в формировании удовлетворенности пользователя системой. Медленная система кажется не подчиняющейся человеку. Она провоцирует многократные щелчки мышью, а в случае крайней медлительности пользователь от нее и вовсе отказывается. Вместо того чтобы срочно приступить к улучшению ее характеристик, инженерам, разработчикам структуры системы и профессионалам в области удобства использования необходимо понять, как пользователи воспринимают время, и научиться использовать это понимание для проектирования хорошей системы. Книга Стива, являясь учебником о восприятии времени

и его применении в проектировании, тем не менее очень приятно просто для чтения на досуге. Эта книга станет классической.

*Деннис Виксон (Dennis Wixon),  
доктор философии, менеджер  
по изучению пользователей,  
Microsoft Games Studios*

Книга Сеова содержит ключевые результаты исследований в области человеко-машинного взаимодействия, относящиеся к реактивности приложений и толерантности пользователей, и расширяет рамки этих исследований, чтобы показать, как упомянутые факторы влияют на удобство использования вообще и на удовлетворение пользователя в частности. Книга предлагает четкие рекомендации в помощь разработчикам приложений для повышения степени удовлетворенности пользователей. Книга будет полезна любым разработчикам, занимающимся созданием пользовательских интерфейсов.

*Марк Фридман (Mark Friedman),  
ведущий разработчик архитектуры,  
отдел разработки и проектирования  
корпорации Microsoft*

Книга Стива Сеова о восприятии проникательна – это взгляд, постигающий сущность роли восприятия и действия; понять его будет полезно проектировщикам, исследователям и организаторам в любой области.

*Август де Лос-Рейес (August de los Reyes),  
разработчик архитектуры, Microsoft Surface*

Если ваши пользователи постоянно жалуются на недостаток, возвышающийся над всеми достоинствами, это результат неудачной реализации. Взаимодействия кажутся слишком длительными. Пользовательские интерфейсы скорее раздражающие, чем удобные. Книга Стива Сеова – единственный источник, дающий ключ к пониманию того, как проектировать быстро реагирующие интерфейсы и, что особенно важно, как создавать впечатление реактивности.

*Арнольд (Арни) М. Ланд (Arnold (Arnie) M. Lund),  
доктор философии, руководитель отдела изучения  
пользовательского восприятия корпорации Microsoft*

## Благодарности

Во-первых, хочу поблагодарить высококвалифицированный редакторский коллектив издательства Addison-Wesley, и в особенности Карен Геттман (Karen Gettman), Джоан Мюррей (Joan Murray), Криса Гузиковски (Chris Guzikowski), Эмили Фрей (Emily Frey), Мэг Шоу (Meg Shaw) и Райну Кробак (Raina Chrobak), которые направляли меня в процессе написания моей первой книги. Во-вторых, выражаю большую благодарность (в том числе от имени читателей) рецензентам – Бернару Фарреллу (Bernard Farrell), Марку Фридману (Mark Friedman), Терри Гардинеру (Terry Gardiner), Тиму Патрику (Tim Patrick) и Сьюзен Рамлет (Susan Ramlet), – чьи комментарии помогли избежать того, чтобы книга превратилась в учебник. Хочу также поблагодарить своих талантливых коллег по компании Microsoft, в частности Марка Фридмана, которые обеспечивали не только ценную обратную связь, но и проверку многих идей этой книги на практике.

## Об авторе

**Стивен К. Сеов** начал работать с компанией Microsoft в качестве специалиста по изучению пользователей (User Researcher) после получения степени доктора философии в области экспериментальной психологии в университете Брауна. В центре внимания его диссертационного исследования было планирование человеком распределения времени. До университета Брауна Стивен, уроженец Сингапура, получил степени бакалавра и магистра по судебной психологии в колледже уголовного права Джона Джея, где темой его диссертации было исследование искажения восприятия времени. Стивен постоянно обсуждает с коллегами по компании вопросы, связанные со временем и его оцениванием. С его научными интересами в области психологии тесно связана приверженность компьютерным технологиям. На досуге Стивен возится с компьютерным «железом», изучает языки программирования и создает веб-сайты. Он живет в Мейпл-Вэлли, штат Вашингтон, с женой и сыном.

За дополнительной информацией о Стивене обратитесь к его сайту [www.StevenSeow.com](http://www.StevenSeow.com).

# Предисловие

При написании этой книги я думал о некоторых специалистах-практиках и профессионалах. Первую группу составляют разработчики программного обеспечения. Не каждый имел возможность (или желание) освоить классы человеко-машинного интерфейса, принципы удобства использования или методы проектирования человеко-машинного интерфейса, и не каждая компания может пригласить на работу профессионала в области удобства использования для того, чтобы сделать веб-сайт дружественным по отношению к пользователю или обеспечить открытость новых возможностей приложения. Если вы – разработчик программного обеспечения или структуры системы, то вполне можно утверждать, что ваше мастерство и профессионализм превращают единицы и нули в решение, которое улучшает жизнь и труд людей. Если, однако, вы не можете отнести себя к профессионалам и не можете воспользоваться услугами таковых, то для вас полезно применение простых идей, способствующих повышению удобства использования и привлекательности вашего решения.

Вторую группу читателей, которую я имел в виду, составляют профессионалы в области удобства использования. Эта группа включает множество моих коллег по компании Microsoft, чья профессиональная ответственность и способности делают их для остальной компании «голосом пользователя», информирующим о том, насколько наши продукты и сервисы легко использовать, изучать и т. д. Если вы похожи на моих коллег, значит, вы беспокоитесь не только о том, как продукт или сервис используется, но и о том, как он воспринимается, какие может вызвать эмоции или действия. Эта книга содержит исследование, логические выводы и другие полезные сведения, способствующие вашему пониманию того, как работать и настраивать характеристики удобства использования, связанные со временем и оценением времени.

В третью группу читателей входят все остальные, кто несет какую-либо ответственность за качество и полезность решения. Ее составляют менеджеры по разработке и поддержке программ и продуктов, тестировщики, специалисты по маркетингу и все, кто принимает решения, связанные с представлением результата пользователю. На самом деле мы должны идти на компромиссы, искать альтернативы и обходные

маневры. Кроме того, мы вынуждены работать с теми ресурсами, которыми располагаем. В данной книге учитываются эти обстоятельства, и она представляет собой практическое руководство к тому, как выбирать приемлемые и разумные компромиссы.

Независимо от вашей подготовки, одной из особенностей этой книги, которая привлечет ваше внимание, является смешение информации, занятых историй, примеров, теорий и правил из множества научных и производственных областей, весьма далеких от сферы проектирования программного обеспечения и человеко-машинного взаимодействия. Сфера обслуживания и розничная торговля, продовольствие и напитки, кулинария, психология, социология, зоология, управление бизнесом, увеселительные мероприятия, банковское дело и связь – вот лишь некоторые из них. Такое эклектическое представление отражает сущность темы этой книги – множественность граней и универсальность восприятия времени.

Восприятие времени человеком трудно описать, и мы ограничимся сферой усовершенствования человеко-машинного взаимодействия. Здесь знания, заимствованные из других областей науки и производства, становятся жизненно важными. Например, одно из моих ранних прозрений, связанных с восприятием времени, пришло, когда я стоял в очереди в местном магазине Костко. Стоя в очереди, я наблюдал, как служащая магазина, размахивая сканирующим пистолетом, энергично «стреляет» по товарам в виртуальной тележке покупателя передо мной. Закончив это, она обратилась ко мне и попросила показать карту Костко. Я предъявил карту, и она «выстрелила» по ней в долю секунды, прежде чем начать проверять мои товары. Она, как охотник, «растреливала» товарные коды в течение нескольких секунд и после сканирования последнего товара отвернулась от меня так же, как до этого от парня, стоявшего впереди. Я решительно спросил ее: «Что же вы сделали?» Она спокойно объяснила, что пробила мои товары и все, что мне теперь остается сделать, – это заплатить за них кассиру. Испытывая скепсис по поводу того, как это работает, и страх, что могу заплатить за лишние товары, я задал ей вопрос, как кассир узнает о том, какая группа товаров является моей. Она взглянула на меня поверх своих очков, взмахнула пистолетом и ответила, что просканировала мою карту.

Через несколько секунд, когда она уже удалилась, не оставив мне времени для следующих вопросов, я понял, насколько блестящей была эта идея. Действительно, когда я подошел к контрольной стойке, кассир просто просканировал мою карту и сообщил, сколько я должен заплатить. Мой товар сошел с полки и после недолгой ручной переноски оказался в багажнике моего автомобиля. Костко эффективно реализовала высокоинтеллектуальное использование моего времени ожидания и ускорила процесс проверки!

Через некоторое время после моего опыта в Костко я осознал, что этот проверенный метод применим к проектированию компьютерных программ. Почему мы не можем задать пользователям все вопросы заранее, а уже потом начать длительный процесс установки? Почему мы не можем загрузить то, что желает пользователь, для немедленного применения, а затем продолжать загрузку других компонентов, отвечающих за менее важные возможности, в фоновом режиме? Почему мы не можем предложить инструкции о том, как пользоваться продуктом, во время установки? Можем и иногда делаем, но в недостаточной степени, поскольку продолжаем видеть, как пользователи воздевают руки в раздражении, и слышать, как они называют приложения, веб-сайты или компьютеры «тупыми». Почему так происходит?

Сейчас, когда я пишу эти строки, микропроцессоры, работающие в обычном домашнем компьютере, могут выполнять миллиарды команд в секунду. Это огромная *мозговая* энергия. Конечно, компьютер не может быть тупым. То, что пользователи нам говорят, свидетельствует не об отсутствии у компьютера *мозга*, а о нехватке *разума*. Мы обсуждаем наличие разума, достаточного для распознавания того, что важно и ценно для пользователя, достаточного для организации ясного и правильного общения с пользователем и для рассмотрения и понимания того, насколько эффективно расходуется время пользователя.

Эта книга, как новое шоу с известными персонажами или продолжение книги «Волшебник страны Оз», представляет собой попытку дать компьютеру разум (или, точнее, увеличить его). Данная книга имеет своей целью обеспечение впечатления быстрой работы пользователя. Это достигается не слепым применением того, что уже работает у других людей, а продвижением на шаг по пути к пониманию того, почему это работает. Я поделил все главы на две основные части. Первая организована тематически и дает специальные знания, руководства и различные рекомендации по вопросам, связанным с пониманием восприятия, пользовательской толерантности, реактивности, обнаружения различий, индикации выполнения, отображения времени и т. д. Вторая часть содержит две компиляции. Первая представляет собой совокупность проверенных методов, отобранных среди психологических принципов, правил ведения бизнеса, промышленных исследований и других источников. (Да, вы найдете здесь метод, имеющий отношение к моему опыту с Костко.) Вторая описывает совокупность нарушений, относящихся к пользовательскому опыту, на языке управления восприятием времени. Здесь вы найдете ряд печально знаменитых историй, например, о явлении флуктуации времени, замеченном Джоном К. Двораком (John C. Dvorak) из *PC Magazine*.

Я надеюсь, что эта книга вызовет у вас две реакции, которые я наблюдал, когда читал лекции по обсуждаемой теме. Первая – это *злорадство*, тип удовольствия, которое вы получаете при виде неудач других

людей. Оно обычно выражается у моих слушателей в виде смеха, когда я описываю страдания пользователей как результат плохого проектирования продуктов и услуг. Вторая – это *эффект покачивания головой*, кивки одобрения слушателей, обычно наблюдаемые перед кульминационным моментом объяснения или последними словами анекдота. И то и другое для меня как для лектора является позитивным знаком, поскольку показывает, что публика сопереживает тому, о чем слышит. Поэтому, если вы во время чтения этой книги обнаружите, что иногда становитесь смешливым, или поймаете себя на покачивании головой, значит, книга была написана не зря.

# Глава 1

## Проектирование и конструирование времени

Каждое взаимодействие с компьютером – от выполнения поиска информации в базе данных, который может длиться несколько секунд, до установки операционной системы, способной занимать четыре часа, – вынуждает пользователя тратить *время*. То, как спроектировано взаимодействие с точки зрения расхода времени пользователя, представляет собой важнейший фактор удобства использования и общего восприятия продукта или сервиса. Плохо спроектированное решение может просто привести к ненужному расходу времени пользователя, какой бы великолепной ни была архитектура. С другой стороны, решение, спроектированное как удачное для пользователя, может завоевать его расположение, несмотря на досадные задержки.

### Звучит привычно?

До обеда остается пять минут, и вы запускаете процесс установки, который, предположительно, займет больше часа. Не желая сидеть перед жидкокристаллическим экраном, непрерывно наблюдая за процессом установки, вы радостно щелкаете по кнопке с понятной надписью *Инсталлировать сейчас!* (Install Now!). Вы щелкаете по кнопке и уходите на обед. Возвращаетесь через час, сытый и довольный. Однако вместо поздравительного сообщения, объявляющего о завершении успешной установки, вы видите диалоговое окно с вопросом, действительно ли вы хотите начать установку.

Вы загружаете из Интернета явно большой файл. Индикатор выполнения в составе пользовательского интерфейса (User Interface, UI) подсказывает, что осталось всего пять секунд. Вы решаете оставаться на месте, чтобы быть свидетелем завершения загрузки. Естественно, идет отсчет оставшихся секунд: четыре, три, две, одна, нуль. Вы в восторге, поскольку долго ждали этот файл с обновлением для игры, содержащий новый набор оружия, а особенно новое фотонное... ожидание, на индикаторе отображается нуль... нуль... нуль...

Подобные мучительные опыты нередки. Мы знаем немало примеров плохого проектирования, вызывающих желание ударить кулаком по чему-нибудь или рвать на себе волосы. Найдите в Интернете короткий видеоролик «Разгневанный человек у компьютера», который гипотетически отражает то, что может сделать с пользователями плохое впечатление от общения с компьютером. К счастью, существует множество способов минимизировать вероятность того, что ваш пользователь достигнет состояния раздражения, показанного в ролике. Эта книга не содержит инструкций по построению машины времени для уменьшения задержек в вашем решении, но подходит к этому очень близко: она объединяет все знания, исследования и практический опыт для того, чтобы помочь вам так *проектировать* и *конструировать* время, чтобы досадные задержки превратились в приятные паузы, а невыносимо долгий процесс ощущался как мимолетный.

## Странности времени

Проповеди о *времени* притягивают массы людей, а специалисты толпами стекаются для изучения Туринской плащаницы: почти любая мыслимая научная дисциплина и ненаучная группа вносят свой вклад в понимание времени. Философия, к примеру, стала заниматься проблемой времени более 2500 лет назад. Обсуждался следующий вопрос: когда имеет место *настоящее*, или что такое «сейчас»? Три секунды спустя после объявления «сейчас» это «сейчас», по существу, переходит в «затем»<sup>1</sup>. Итак, когда же существует «сейчас»?

С самого начала своего существования, с конца 1800-х годов, психология фокусировала свое внимание на использовании времени реакции – времени, которое требуется для простого отклика (например, в виде нажатия клавиши пальцем) на появление раздражителя (например, возникновение точки на экране), – как меры для изучения поведения. Например, одним из ученых, впервые использовавших время, является Ф. К. Дондерс (F. C. Donders), который регистрировал время реакции людей на простой раздражитель (например, нажатие клавиши

<sup>1</sup> Этот парадокс называется *правдоподобным настоящим*. Первоначально его сформулировал Е. Р. Клей, а начал обсуждать ученый, которого многие считают отцом американской психологии, Уильям Джеймс, в своем серьезном труде «Принципы психологии».

немедленно при появлении красной точки) и время реакции выбора (например, требуется нажать клавишу немедленно при появлении красной точки, но не нажимать ее при появлении синей). Время, затрачиваемое людьми на простое принятие решения<sup>1</sup>, определялось вычитанием времени простой реакции из времени реакции выбора. В настоящее время в среде исследователей поведения измерение времени остается распространенным методом, служащим для оценивания и исследования мыслительных процессов.

Время также может служить простым и удобным для манипулирования фактором (чаще всего посредством задержек) при оценивании поведения и при исследовании познавательных процессов. Одной из ранних попыток такого использования времени является классический эксперимент с памятью, проведенный в 1885 году Г. Эббингаузом (H. Ebbinghaus)<sup>2</sup>, которого многие считают отцом исследования памяти. Эббингауз систематически изучал и описывал забывание запомненной информации с течением времени путем манипулирования периодом времени, предшествующим просьбе вспомнить бессмысленные слова (например, FQR, GYW и т. д.), став автором понятия «*кривая обучения*».

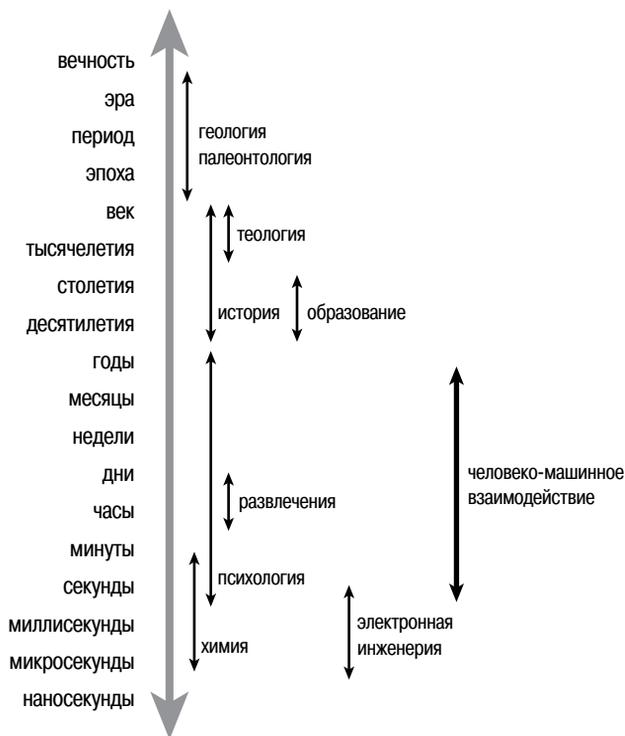
Многие другие дисциплины внесли существенный вклад в понимание времени: физика, химия, нейробиология, археология, геология и т. д. Каждая из них концентрирует внимание на специфических аспектах и временной шкале. На рис. 1.1 представлены типичные диапазоны рассмотрения времени в некоторых дисциплинах. В то время как наука и религия спорят об истоке времени, нейробиология интересуется механизмами типа «момент-за-моментом» для вневременных генов при управлении циркадными ритмами. И хотя диапазон времени в экспериментальной психологии не превышает нескольких минут, основные единицы времени для геологического времени включают тысячелетия и эры. Серьезная беседа о времени может длиться целую вечность.

Человеко-машинное взаимодействие, в основном, оперирует единицами времени от миллисекунд (например, время немедленного отклика) до часов (например, установка операционной системы), в то время как длительность других взаимодействий может превышать несколько часов. Управление еженедельно выполняемыми задачами в календарном приложении легко может оперировать неделями и месяцами. Некоторые сложные математические расчеты на компьютере могут требовать

---

<sup>1</sup> Франс Корнелис Дондерс (1818–1889) аргументированно доказал, что время, необходимое для принятия решения, линейно возрастает с увеличением количества выборов. Проще говоря, чем больше имеется возможностей выбора, тем дольше время, требуемое для принятия правильного решения. Этот факт подтвердили многие исследователи, и наиболее убедительно это сделал в 1952 году В. Э. Хик, чья работа привела впоследствии к открытию закона Хика–Хаймана.

<sup>2</sup> Эббингауз Герман (24.01.1850–26.02.1909) – немецкий психолог, один из основателей экспериментальной психологии. – *Примеч. науч. ред.*



*Рис. 1.1. Соотносительное представление и сопоставление временных шкал в некоторых дисциплинах*

для своего завершения несколько дней или недель. Эта книга фокусирует внимание на опытах работы со временем типа «день-за-днем» и на обозримых человеко-машинных взаимодействиях.

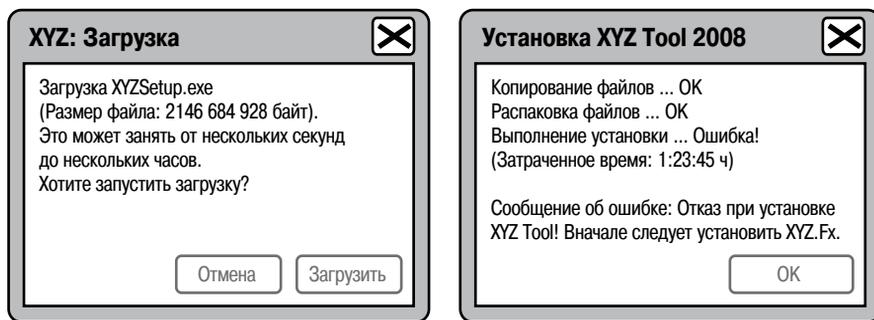
## Можно ли конструировать время

Объединение терминов «конструирование» и «время» непременно воскрешает в памяти образы из «Машины времени» Г. Уэллса: громадная машина, вся в мерцающих вольфрамовых лампочках, с непрерывно вращающимися зубчатыми колесами со всех сторон и ковшеобразным сиденьем. Это воплощение голливудского представления о конструировании времени, и, насколько нам известно, на момент написания этой книги не существует транспортного средства или механизма для путешествий во времени. Мы не можем ускорить или замедлить ход времени. Не так ли?

Хорошо... время *объективно* измеряется и при этом *субъективно* ощущается. Два человека могут ощущать трехчасовой период буквально по-разному в зависимости от интересов, обстоятельств или, к примеру,

языка каждого из них. Несмотря на то что мы не можем вернуться в прошлое, чтобы увидеть, как египтяне строят Великую пирамиду Гизы, или ускорить время, чтобы поздравить с днем рождения наших праправнуков, мы можем работать с *ощущением* времени. Ключом к конструированию субъективного времени является конструирование факторов, влияющих на ощущение времени.

Отбросим гравитационное растяжение времени<sup>1</sup>, потерю сознания, редкие неврологические заболевания, воздействие психотропных субстанций и другие подобные факторы. Простые и обычные жизненные обстоятельства могут влиять на то, как мы ощущаем время. Например, безобидные на первый взгляд опечатки и невнятные формулировки в UI могут привести к изменению ощущения времени пользователя – к лучшему или к худшему – абсолютно по-разному. Рис. 1.2 демонстрирует два гиперболизированных примера.



*Рис. 1.2. Примеры того, как безобидные на первый взгляд формулировки в UI (диалоговое окно слева) или дизайн (диалоговое окно справа) могут ухудшить впечатление пользователя до уровня ниже среднего*

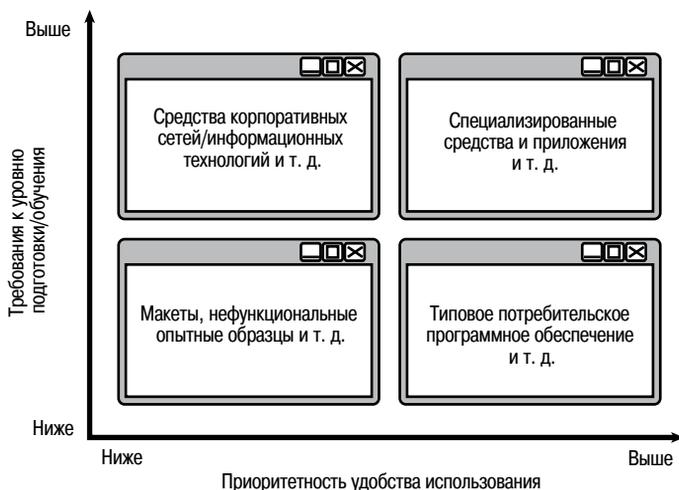
## Зачем проектировать и конструировать время

Перед изучением целесообразности проектирования и конструирования времени для оптимизации человеко-машинного взаимодействия рассмотрим следующее. В ряде случаев нам не надо беспокоиться по поводу времени и его оценивания в создаваемом продукте. Для многих инструментов и решений, не предназначенных для персонального использования и для продажи или использования другими людьми, определенного рода аспекты продукта не всегда имеют высший приоритет.

<sup>1</sup> Гравитационное растяжение времени – это одна из двух теорий Эйнштейна о возможности изменения *объективного* времени. Предположения Эйнштейна были подтверждены известными экспериментами Паунда–Ремби в 1959 году и Хафеле–Китинга в 1971 году. Теоретически предположим, что кто-то отправляется на транспортном средстве со скоростью, близкой к световой, и путешествует год. Согласно теории Эйнштейна пассажиры будут удивлены тем, что их дети стали старше на семь лет!

Это решения типа герметизирующей ленты или вешалки для пальто – решения, выполненные ради практичности и утилитарности, а не элегантности и производительности. Примеры включают любительские проекты, личные макросы Excel и частично функционирующие макеты приложений.

Другое дело, когда вопросы оценивания времени не рассматриваются (но должны рассматриваться) как высокоприоритетные. Это справедливо для тех случаев, когда решения разработаны или изменены по специальным требованиям для внутреннего применения в компании. Общее удобство использования таких собственных решений обычно очень низкое. В рамках международного исследования удобства использования внутрикорпоративных сетей, предпринятого организацией Nielsen Norman Group, к примеру, обнаружилось несколько вопиющих примеров в плане удобства использования. Средства, которые были бы сэкономлены за счет исправления и, как следствие, повышения производительности труда служащих, консультанты оценивают для компаний средних размеров примерно в 5 миллионов долларов. То, что многие собственные решения для внутреннего пользования оказываются не на высоте, можно объяснить многими причинами, но главная среди них – компромисс между *обучением* и *удобством использования* (рис. 1.3). Он состоит в том, что обычно от служащего



**Рис. 1.3.** Средства, разрабатываемые компаниями для внутреннего применения (верхний левый угол), обычно страдают от недостаточного внимания к удобству использования, поскольку предполагается, что пользователи их изучат и подготовятся к работе. Коммерческие приложения, например для записи на CD, должны в идеале располагаться в нижнем правом углу, где от пользователя требуется лишь минимальная подготовка благодаря относительно большому вниманию, которое было уделено удобству использования. Исключения попадают в верхний правый и нижний левый углы

ожидают, что он освоит особенности использования внутреннего решения или уже подготовлен к нему, и это рассматривается как часть его профессиональных обязанностей, поэтому вопросы удобства использования, такие как простота использования или понятность, в списке приоритетов стоят невысоко. Напротив, для многих коммерческих решений считается неуместным требовать профессиональной подготовки пользователей, поэтому больше внимания уделяется тому, чтобы создать решение с более высоким уровнем удобства использования.

В большинстве других случаев для обеспечения оптимального опыта работы с компьютером необходимо учитывать восприятие времени, потому что все программные продукты, по существу, требуют от пользователя временных затрат: установка, поиск, загрузка, компиляция, перемещение файлов и т. д. Некоторые из этих взаимодействий занимают небольшие интервалы времени, но другие являются «пожирателями времени», требующими для своего успешного завершения нескольких часов. Имеются и другие серьезные и специфические причины для того, чтобы производитель каких-то коммерческих или некоммерческих решений обращал внимание на то, как приложение тратит время пользователя.

## В отличие от денег, время переменчиво

Все мы знаем, что *время – деньги*<sup>1</sup>, но не всегда осознаем, что мы не очень сильны в планировании своего времени. Недавние исследования показали, что люди позволяют себе неразумный расход ресурсов ради будущих целей и задач, поскольку переоценивают время, имеющееся в их распоряжении для выполнения намеченного. По словам исследователей, это выявляется, когда «Да!» переходит в «Черт побери!». Даже для более коротких интервалов времени, которые мы будем обсуждать в следующей главе, наше восприятие времени сильно подвержено искажению.

Время и деньги, бесспорно, являются для людей парой главных ценностей. Однако, в отличие от денег, время представляет собой ценность, которая изменяется по своей величине в зависимости от многих факторов, например времени дня и обстоятельств. Исследование потребителей показало, что различное время дня стоит по-разному. Например, некоторые люди менее терпеливы во время обеда, поэтому каждая минута во время обеда более ценна, чем минута унылого времени после полудня. Даже внутри одного и того же интервала времени длительность может иметь различный вес в различных обстоятельствах. Например, исследование показало, что люди оценивают время ожидания в три раза выше, чем время, проводимое внутри транспортного средства

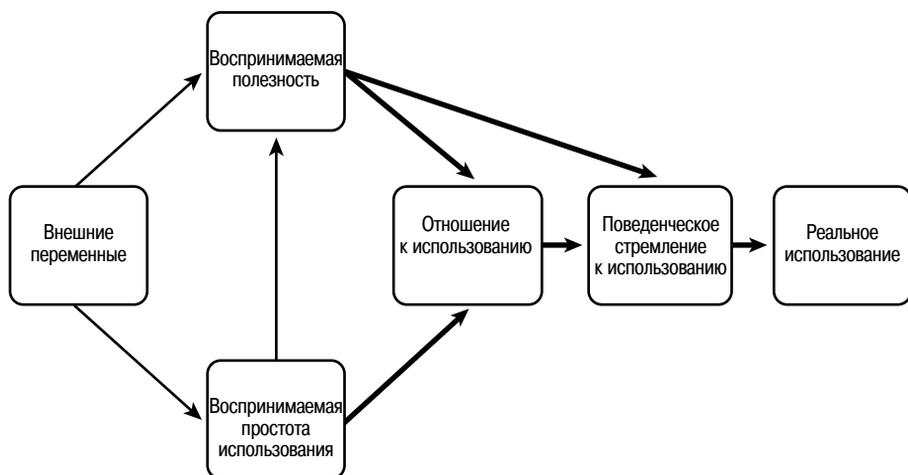
---

<sup>1</sup> Один английский профессор экономики действительно получил математическую формулу для расчета реальной денежной стоимости времени для человека.

в пути, и это может объяснить, почему некоторые из нас предпочитают медленному перемещению по магистрали объездную проселочную дорогу, даже если обе дороги потребуют одинакового времени для достижения цели.

## Все стимулируется восприятием

Не стоит сомневаться по поводу того, что удовлетворение стимулируется восприятием производительности. Все мы хотим ускорить загрузку, быстрее получить результаты, иметь интерфейсы, реагирующие быстрее, и т. д. Во многих направлениях исследований в различных дисциплинах неоднократно показано, что воспринимаемое более важно, чем объективно предлагаемое, потому что восприятие управляет поведением и принятием решений. На рис. 1.4 показана модель (модель восприятия технологии – Technology Acceptance Model, TAM), связывающая, среди других факторов, воспринимаемую полезность и воспринимаемую простоту использования с конечным реальным одобрением и продолжительным использованием технологии. Фактически изменение восприятия для имитации реальности является основой многих форм художественного выражения и увеселительных мероприятий. Многие другие факторы также стимулируют удовлетворение, среди них – удобство использования, эстетика и экономичность. Но то, что заставляет все эти факторы служить к общему удовлетворению, – так это восприятие. Не имеет значения, насколько быстро выполняется загрузка, насколько удобна конструкция продукта, насколько основательно мы организуем представительский уровень и насколько рациональна цена, которую мы назначаем; если восприятие в каком-то пункте искажено, все преимущества не работают.



*Рис. 1.4. Модель восприятия технологии по Фреду Дэвису (Fred Davis) демонстрирует важность восприятия в прогнозировании одобрения и использования технологии*

Что касается восприятия производительности, то исследование потребителей неоднократно подтверждало простой факт: если производительность воспринимается как превосходящая ожидания, то степень удовлетворения высока. И наоборот, если производительность воспринимается как низкая по сравнению с ожидаемой, то степень удовлетворения низка. Этот принцип известен как *первый закон сервиса Майстера*, который устанавливает, что удовлетворение является функцией *несогласованности*, или разницы между тем, что мы воспринимаем, и тем, что мы ожидаем. Проще говоря, если восприятие не соответствует ожидаемому, неудовлетворенность возрастает.

## Кто такой конструктор времени

Итак, время можно конструировать, но кто же такой конструктор времени? Служащая магазина Костко, которая, пока вы стоите в очереди, предварительно сканирует ваши товары для ускорения процесса контроля, является конструктором времени. Служащий клиники, который предлагает вам заполнить длинный, но необходимый документ, пока вы ожидаете доктора, является конструктором времени. Дизайнер парка развлечений, устанавливающий ларьки с интересными товарами среди популярных аттракционов, является конструктором времени. Общие для этих примеров факторы очевидны. В каждом случае изменение ситуации приводит к ускорению событий для покупателя, пациента и постоянного клиента. Следовательно, конструктор времени – это тот, кто применяет принципы восприятия времени и реализующие их методы к существующему решению для достижения наилучшего восприятия у людей, которые взаимодействуют с решением.

## Об этой книге

### Организация глав

Цель данной книги, вообще говоря, состоит в том, чтобы объединить знания, исследования и опыт и сделать их достаточно согласованными для приложения к практическим задачам, которые могут встретиться вам при разработке решений. Для достижения этой цели книга после данной главы делится на две части. Первая часть содержит следующие восемь глав. Они охватывают специальные темы, касающиеся времени и распределения времени в UI.

#### *Глава 2 «Восприятие и толерантность»*

В этой главе обсуждается важность дифференциации *воспринимаемой длительности* и *реальной длительности*, а также роль *толерантности пользователя* и факторов, влияющих на нее.

### *Глава 3 «Время отклика пользователя и системы»*

В этой главе вводятся понятия времени отклика пользователя и системы и обсуждаются промышленные нормативы для максимально приемлемого времени отклика.

### *Глава 4 «Реактивность»*

В этой главе определяется реактивность как функция ожиданий пользователя и вводится четыре класса реактивности: мгновенная, незамедлительная, не прерывающая и вынуждающая.

### *Глава 5 «Обнаружение различий в распределении времени»*

Эта глава знакомит с основными концепциями обнаружения человеком изменений в распределении времени. Ее цель – содействие в определении требуемых эксплуатационных характеристик, например в задании целевой производительности и определении допустимой регрессии.

### *Глава 6 «Индикатор выполнения»*

В этой главе описывается подход к классификации видов индикации выполнения и объясняется, как он может помочь в определении наиболее подходящего типа индикации.

### *Глава 7 «Представление времени»*

В этой главе обсуждается, когда и как вам следует обеспечивать представление времени в своем продукте, и приводятся сведения об использовании якорей времени.

### *Глава 8 «Потоковое состояние пользователя»*

В этой главе обсуждается, как создать оптимальное потоковое состояние пользователя с учетом идей, заимствованных из исследования психологии потокового состояния.

### *Глава 9 «Тестирование времени»*

В этой главе обсуждаются различные подходы к сбору данных о реальной длительности, воспринимаемой длительности и толерантности пользователя.

Многие сложности, ограничения и препятствия, с которыми вы встретитесь при разработке своего решения, могут быть преодолены путем перестройки базовой архитектуры, модернизации аппаратной части или наращивания вычислительной мощности. К сожалению, такие подходы зачастую требуют больших усилий, времени и денег.

Вторая часть содержит две заключительные главы, в которых представлены некоторые альтернативы дорогостоящих решений.

### *Глава 10 «Методы»*

В этой главе описываются методы управления восприятием и управления толерантностью, на которые следует обратить внимание при проектировании UI.

## Глава 11 «Нарушения»

В этой главе описываются некоторые из встречающихся нарушений и обсуждаются способы предотвращения и исправления нарушений.

## Терминология

Идеи и практические опыты, упоминаемые в этой книге, носят обобщенный характер. Это означает, что они применимы не только к конкретным компьютерным приложениям, функционирующим под управлением тех или иных операционных систем или компьютерных систем другого рода. Поэтому я старался использовать общую терминологию, применимую для большинства приложений и предмета рассмотрения данной книги.

### Решение

Термин *«решение»* применяется в этой книге к компьютерным приложениям, службам, веб-сайтам, устройствам, аппаратному обеспечению и любым другим продуктам, содержащим UI для отправки, получения или обмена какой-то информацией с пользователем. Примеры включают веб-сайт, предназначенный для электронной коммерции, антивирусную компьютерную программу, обучающую компьютерную программу и простую программу электронной почты.

### Процесс

В контексте данной книги термин *«процесс»* относится к операции, задаче, функции или другому активному объекту (либо к набору таких объектов), приводимому в исполнение решением. Как правило, один или несколько аспектов решения, связанных со временем, например время запуска, длительность и время завершения, значимы для пользователя. Некоторые процессы характеризуются длительным выполнением, например установка, загрузка информации, запросы к базам данных, другие – относительно малой длительностью выполнения, например поведение средств управления UI, изменение масштаба и размеров окон.

### Пользователь

Термин *«пользователь»* подразумевает конечного пользователя, потребителя, оператора или какого-то человека, непосредственно или опосредованно взаимодействующего с вашим решением. Кроме очевидных пользователей, сидящих перед экраном стационарного или переносного компьютера, это может быть клиент банка, взаимодействующий с программным обеспечением, установленным в банкоматах, работник, доставляющий покупки на дом и пользующийся портативным устройством для управления доставкой, или медсестра, использующая сложное медицинское оборудование для наблюдения за здоровьем пациента.

## Изображение взамен тысячи слов

Для повышения доходчивости повествования о теории и опыте, о страдании и блаженстве и о методах и нарушениях я разместил в этой книге около 100 схем. Многие из этих схем представляют UI продуктов одной из двух вымышленных торговых марок: ABC и XYZ. UI из набора решений ABC иллюстрируют примеры хорошего проектирования и методов, что является залогом произведения на пользователя хорошего впечатления. Решения, принадлежащие XYZ, характеризуются, напротив, ошибками и просчетами проектирования, которые подлежат устранению.

## Спуск в кроличью нору

По всей книге представлен широкий спектр знаний, от психологических теорий с педантичным упоминанием имен до сведений общего характера, от исследований выдающихся ученых нобелевского уровня до диссертационных проектов начинающих аспирантов и от методических руководств профессионалов в области удобства использования до эффективных методов, используемых служащими магазинов. Эта книга обращена к практике и применению. На случай если вы хотите «спуститься глубже в кроличью нору», я предлагаю в конце каждой главы список исследований, прочитанных лекций и литературы. Список по каждой теме сделан коротким, чтобы «Кроличья нора» не стала полномасштабным библиографическим списком. По всей вероятности, следование за одним ресурсом приведет вас к большему числу источников – в этом назначение раздела «Кроличья нора»! Большинство этих ресурсов согласуется с изложением в главе, а некоторые, по обыкновению, представляют противоположные теории, контраргументы или альтернативные подходы. Читатель может ознакомиться со всей теоретической и фактической информацией, соответствующей его проблеме, и принять окончательное решение о подходе к решению.

## Кроличья нора

### Классические психологические опыты со временем

Donders, F. C. Die Schnelligkeit psychischer Prozesse (On the speed of mental processes) Archiv für Anatomie und Physiologie und wissenschaftliche Medizin, 1868. 657–681.

Ebbinghaus, H. Über das Gedächtnis: Untersuchungen zur experimentellen Psychologie. Leipzig: Duncker and Humblot. Trans, 1885.

H. A. Ruger and C. E. Bussenius. Memory: A Contribution to Experimental Psychology. New York: Teachers College, Columbia University, 1913. Reprint 1964. New York: Dover.

## Содержательные статьи о восприятии времени

Allan, L. G. The perception of time. *Perception & Psychophysics*, 26, 1979. 340–354.

Fraisse, P. *The Psychology of Time*. New York: Harper & Row, 1963.

James, W. *Principles of Psychology*. New York: Holt, 1890. (Примечание автора: доступна онлайн-версия по адресу <http://psychclassics.yorku.ca/James/Principles/>. См. раздел «Chapter 15».)

Woodrow, H. Time perception. In *Handbook of Experimental Psychology*, (Ed.) S. S. Stevens. New York: Wiley, 1951.

## Научные исследования и манипулирование объективным временем

Hafele, J. and R. Keating. Around the world atomic clocks: Predicted relativistic time gains. *Science*, 177, 1972. 166–170.

Pound, R.V. and G. A. Rebka, Jr. Gravitational red-shift in nuclear resonance. *Physical Review Letters*, 3, 1959. 439–441.

## Книги о времени

Hawking, S. *A Brief History of Time*. Bantam Books, 1988<sup>1</sup>.

McCready, S. *The Discovery of Time*. Naperville, IL: Sourcebooks, Inc., 2001.

Whitrow, G. J. *What Is Time? The classic account of the nature of time*. New York: Oxford Press, 1972.

## Время и деньги

Leclerc, F., B. H. Schmitt and L. Dube. Waiting time and decision making: Is time like money? *Journal of Consumer Research*, 22, 1995. 110–119.

Time is Money, Professor Proves. CNN.com/SCI-TECH. May 29, 2002. (Примечание автора: доступна онлайн-версия по адресу <http://archives.cnn.com/2002/TECH/science/05/29/time.money/>.)

Zauberman, G. and J. G. Lynch. Resource slack and propensity to discount delayed investments of time versus money. *Journal of Experimental Psychology: General*, 134, 2005. 23–37.

---

<sup>1</sup> Стивен Хокинг «Краткая история времени от большого взрыва до черных дыр». – Пер. с англ. – СПб: Амфора, 2001.

## Восприятие времени и производительность

Barber, R. E. and H. C. Lucas. System response time, operator productivity, and job satisfaction. *Communications of the ACM*, 26, 1983. 972–996.

Miller, R. B. Response time in man-computer conversational transaction. Fall Joint Comp. Conf. U.S.A., 1968. 267–277.

Shneiderman, B. Response time and display rate in human performance with computers. *Computing Survey*, 16, 1984. 265–285.

## Восприятие времени и удовлетворение

Davis, M. M. and J. Heineke. How disconfirmation, perception and actual waiting times impact customer satisfaction. *International Journal of Service Industry Management*, 9, 1998. 64–73.

Davis, M. M. and T. E. Vollman. A framework for relating waiting time and customer satisfaction in a service operation. *The Journal of Services Marketing*, 4, 1990. 61–69.

Durrande-Moreau, A. Waiting for service: Ten years of empirical research. *International Journal of Service Industry Management*, 10, 1999. 171–189.

Maister, D. H. The psychology of waiting lines. In Czepiel (Ed.), *The Service Encounter*. Lexington, MA: Lexington Books, 1985. 113–123.

Tom, G. and S. Lucey. Waiting time delays and customer satisfaction in supermarkets. *Journal of Services Marketing*, 9, 1995. 20–29.

## Восприятие времени и использование компьютерных программ

Davis, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13, 1989. 319–340.

Morris, M. G. and A. Dillon. How user perceptions influence software use. *IEEE Software*, 14, 1997. 58–65.

## Кроличьи норы

Кэрролл Л. Приключения Алисы в Стране Чудес. Зазеркалье: про то, что увидела там Алиса // Пер. А. А. Щербакова. – М.: Художественная литература. – 1977. (Примечание автора: из этой книги взят термин «кроличья нора». Доступна онлайн-версия по адресу [http://lib.ru/CARROLL/alisa\\_zah.txt](http://lib.ru/CARROLL/alisa_zah.txt) (пер. Б. Заходера).)

# Глава 2

## Восприятие и толерантность

Помимо фактического и объективного оценивания времени, важно также понять, что же воспринимается пользователями и каков их уровень толерантности. Например, трехминутная загрузка может восприниматься как пятиминутная, но вполне возможно, что пользователи согласны ждать лишь одну минуту. В этой главе обсуждается важность дифференциации воспринимаемой длительности и фактической длительности и описывается роль пользовательской толерантности.

### Восприятие и толерантность

Ваши пользователи недовольны тем, что какой-то процесс в вашем решении работает невыносимо медленно. Вы запускаете тест производительности и определяете, что все функционирует в соответствии со спецификацией. Тем не менее вы с готовностью посвящаете несколько дней дополнительной настройке своего решения и делаете процесс работающим несколько быстрее. Вы с оптимизмом выпускаете новую, улучшенную версию своего решения. Спустя несколько недель выносится вердикт... по-прежнему недопустимо медленно.

Критики из известного компьютерного журнала опубликовали обзор, где сравнили ваше решение с решением конкурента. По реактивности пользователи дают решению конкурента пять из пяти звездочек, а вашему решению – жалкие две. В недоумении, вы подвергаете тестированию на реактивность оба решения. Результат: оба решения одинаково реактивны. Что происходит?

Основной контингент пользователей компьютеров обычно не хронометрирует свои взаимодействия с компьютером, выполняя повседневные задачи, и предпочитает игнорировать оценивание времени, если только время не привлекает их внимание, например, в случае, когда задержка значительно превышает привычные. Пользователи, оценивая длительности, вероятнее всего, будут руководствоваться своими внутренними представлениями, а не объективными измерениями. Поскольку восприятие в высокой степени подвержено искажению, то, что пользователь воспринимает, может сильно отличаться от того, что в действительности происходит.

Рассмотрим следующие вопросы:

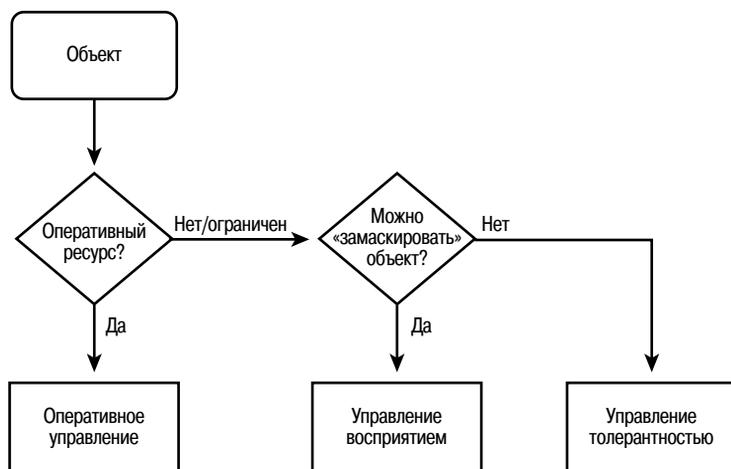
- Как мы можем добиться того, чтобы процесс загрузки стал быстрее?
- Как мы можем добиться того, чтобы процесс загрузки казался быстрее?
- Как заставить пользователей относиться к загрузке терпеливо?

Несмотря на то что вопросы кажутся одинаковыми, на самом деле они разные и требуют различных ответов. В зависимости от обстоятельств, ограничений и других стесняющих вас условий может оказаться, что на тот или иной вопрос ответить легче, чем на другие. Ответ на первый вопрос, вероятно, будет чисто техническим, требующим изменения реального процесса, протокола или вида обработки. Такие подходы, восходящие к индустрии розничной торговли и обслуживания, могут быть охарактеризованы как *оперативные*. В индустрии розничной торговли и обслуживания *оперативное управление* предполагает существенные изменения в способе ведения бизнеса. Возвращаясь к первому вопросу, можно сказать, что если время и ресурсы позволяют реализовать новые технологии для ускорения загрузки (например, использование улучшенных алгоритмов сжатия, дополнительного аппаратного обеспечения и т. д.), то это хороший вопрос для того, чтобы его задать и на него ответить.

В то время как ответ на первый вопрос, как правило, требует исправления или реконструкции, ответ на второй больше связан с созданием нового облика, потому что он предполагает подход, свободный от глубоких оперативных изменений. В контексте конструирования программного обеспечения при наличии технических ограничений лучше задать этот второй вопрос и ответить на него. Отвлечение внимания людей от длительного ожидания – очевидный вариант. Соответствующие методы применяются в той сфере, которая в розничной торговле и индустрии обслуживания носит название «*управление восприятием*». Например, управляющий банком может принять решение увеличить численность персонала для снижения времени ожидания (оперативное управление) или поставить рядом с очередью телевизор (управление восприятием).

Наконец, могут сложиться обстоятельства, когда действуют оперативные ограничения (ограничения бюджета, недоступность аппаратуры,

план и т. д.) и задержки «замаскировать» слишком трудно: например, проводятся сложные исследования, обещающие принести большую выгоду, но непременно требующие времени для своего завершения. Тогда критически важным становится приложение усилий к увеличению *пользовательской толерантности*. Такие ситуации делают последний вопрос наиболее актуальным. *Управление толерантностью* не стремится «замаскировать» реальную длительность, оно, напротив, концентрируется на том, чтобы сделать пользователей более терпимыми к неизбежным ожиданиям или задержкам. На рис. 2.1 приведена структурная схема, где представлены подходы к методам управления и показано, когда следует применять тот или иной подход. Больше информации о методах управления восприятием и толерантностью, а также о нарушениях можно найти в главе 10 «Методы» и в главе 11 «Нарушения».



*Рис. 2.1. Структурная схема, помогающая принять решение о том, когда следует использовать оперативное управление, управление восприятием и управление толерантностью*

## Действительность или ощущение?

Неудовлетворенность характеристиками решения, та, о которой рассказано в начале этой главы, аналогична неудовлетворенности медленным обслуживанием в ресторане; медленность может быть *фактической* или *кажущейся*. Люди *знают*, что нечто происходит медленно, или *чувствуют* это? Обедаящий может забеспокоиться из-за того, что официант, возможно, забыл передать заказ повару, или из-за того, что заметил, как другой столик обслужили раньше. Аналогично, в компьютерном контексте пользователь может обнаружить некое бездействие по множеству причин. Загрузка может в самом деле быть медленной из-за необычайно высокого уровня сетевого трафика, или она может восприниматься как медленная из-за того, что пользователю срочно нужна информация.

Заметим, что установление различия между действительностью и ощущением представляется важным, но то, что говорят пользователи, не следует оспаривать. Если вы хотите вместе с пользователем проверить, действительной или кажущейся является длительность, о которой он говорит, надо проявить осторожность. В этом случае лучше спросить «Можете ли вы дать мне больше информации о задержке?», чем «А вы замеряли время выполнения?».

## Реальность: фактическое время

Несмотря на то что эта книга фокусирует внимание на управлении восприятием и толерантностью, имеется несколько важных соображений, связанных с объективным оцениванием времени. Во-первых, в контексте человеко-машинного взаимодействия объективное оценивание времени, например времени отклика, должно выполняться в «системе координат» пользователя. Во-вторых, объективное оценивание времени должно выполняться объективными средствами. Наконец, сбор всей возможной информации всегда дает преимущество. Первое положение мы обсудим в следующей главе. Давайте пристальнее рассмотрим два последних.

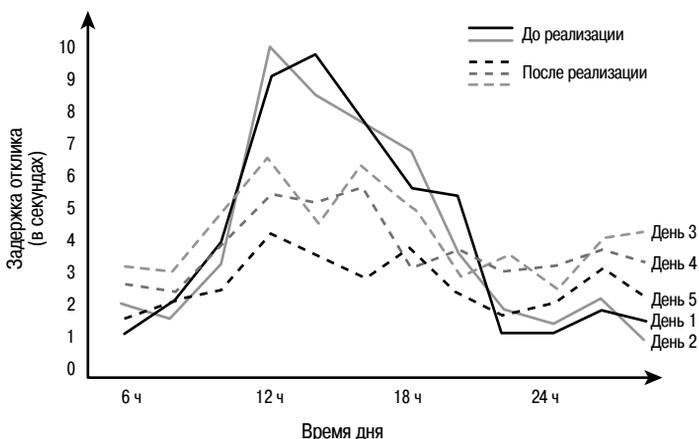
## Точность: объективное время на основе объективных измерений

Очень важно установить надежные и достоверные значения фактических длительностей, потому что они представляют единственный объективный показатель в исследовании времени, заявленного пользователем, и в задачах, связанных с оцениванием времени. Установление надежных и достоверных данных является решающим при сравнении с воспринимаемой длительностью. Существует множество методов определения фактических длительностей. Иногда числа можно получить при неофициальных измерениях, из системных журналов с метками времени, технической спецификации или путем расчетов по формулам, в частности запрограммированных. Во всех случаях, в силу того что реальные длительности отражают *объективное* время, оно должно измеряться *объективными* средствами. Мы отложим более подробное обсуждение этого вопроса до главы 9 «Тестирование времени».

Если данные о реальных длительностях получены из вторичных источников (не измерены непосредственно вами), например от пользователя или из документации, будет хорошо, если вы проверите, как эти данные были получены, и попытаетесь воспроизвести их получение. Сомнительные оценки не могут адекватно заменить собой фактические длительности. Если процесс измерен необъективно, такие измерения более уместно считать воспринятыми в ощущении.

## Данные: объем, изменчивость, разнообразие

Наряду с точностью, решающим является также объем данных. При любой возможности выполняйте множественные измерения искомой длительности, применяя разнообразные методы измерений, для того чтобы получить и проанализировать представительный набор результатов. Например, углубленный анализ данных, такой как анализ временных рядов, может выявить влияние внушения, к примеру, эффект бутылочного горлышка, который изменяет длительность в сторону увеличения всякий раз, когда процесс стартует или устанавливает контакт. Аналогично, простое вычерчивание распределения данных по частотам может показать, где данные центрируются и как они рассеяны, а это может помочь диагностировать проблему. На рис. 2.2 показаны гипотетические объективные данные о задержке отклика при загрузке веб-сайта до и после выполнения новой компьютерной программы. Такой анализ был бы невозможен без сбора массы данных.

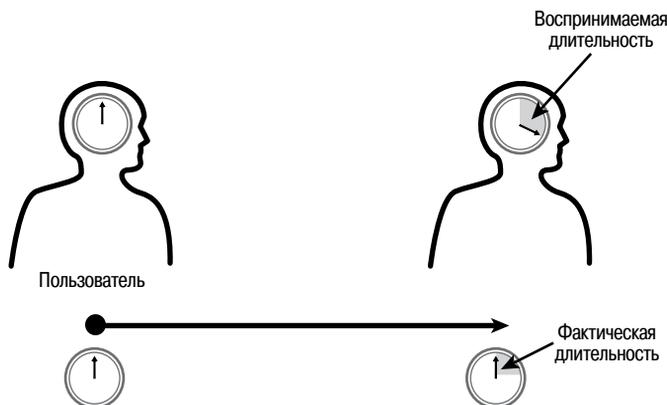


**Рис. 2.2.** Гипотетические объективные данные о задержке отклика при загрузке веб-сайта до (сплошная линия) и после (штриховая линия) реализации новой компьютерной программы. Сбор множества данных обеспечивает возможность более информативных видов анализа

## Восприятие: психологическое время

В то время как объективная реальность соотносится с фактами (независимо от того, ощущаем ли мы их), восприятие зависит от того, как мы ощущаем объективную реальность (независимо от того, является ли она действительной). Мы проживаем каждый день, управляемые нашим мозгом, полным идей и теорий о том, как и почему предметы таковы, каковы они есть. Обычно имеет значение, точны ли мы в своем восприятии ежедневных событий. В общем, мы не задаем вопросов своим ощущениям. Если кто-то считает, что некоторое событие заняло

«примерно час», это говорит лишь о том, какова длительность события по мнению этого человека (рис. 2.3). Хороший способ составить мнение об ощущаемой длительности состоит в том, чтобы представить себе воспринимающего в виде секундомера, сообщающего значение длительности события, пусть и неточно. Такое время многие психологи называют *психологическим временем*.

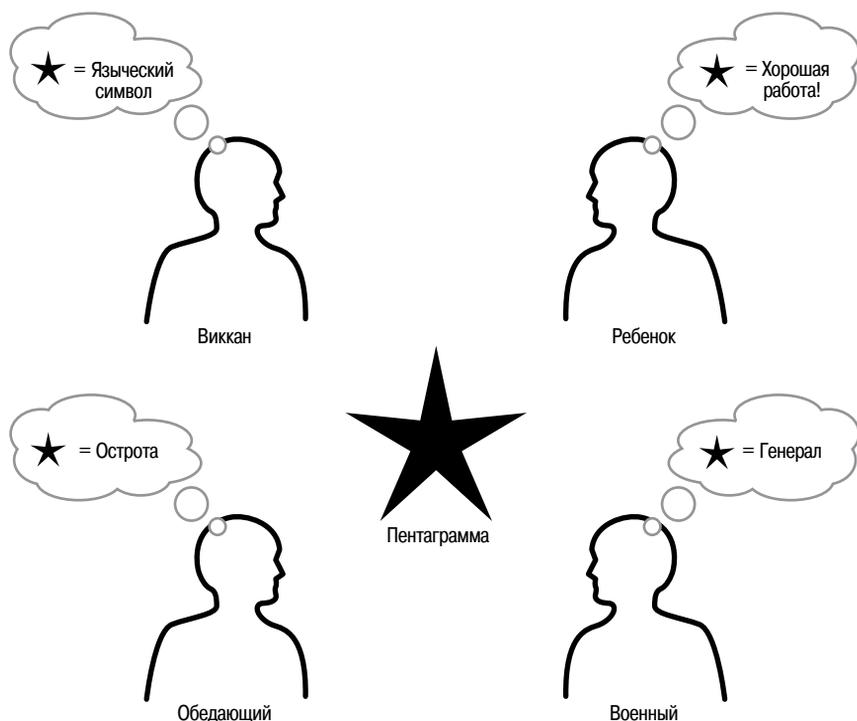


*Рис. 2.3. Воспринимаемая длительность никогда не должна рассматриваться как истинная и в точности равная фактической объективной длительности*

## Субъективность: видите ли вы то, что вижу я?

Строго говоря, восприятие связано тем, что делает мозг с информацией, полученной от органов чувств. Предположим, что человеку показывают пятиконечную звезду (рис. 2.4). Визуальный раздражитель достигает глаза, и крошечные фоторецепторы на сетчатке обнаруживают и преобразуют визуальную информацию в нервные сигналы, которые направляются для обработки в различные области мозга. Начиная с этого момента многие факторы, например возраст, образование, культура и даже убеждения и опыт, будут определять значение символа для конкретного человека. Мозгом первоклассника звезда воспринимается как символ награды за хорошо выполненную работу. В мозгу виккана<sup>1</sup> она, вероятно, будет ассоциироваться с культом языческой веры. В мозгу обедающего символ вполне может указывать на пряность блюда! В мозгу военного этот символ обозначает ранг офицера. А если так, то ощущение органами чувств физического мира никогда не может считаться истинным и точно соответствовать тому, что складывается в мозгу воспринимающего.

<sup>1</sup> Викканство – западноевропейский неоязыческий культ. – *Примеч. перев.*



*Рис. 2.4. Восприятие – это то, что мозг делает с информацией, которую получает от органов чувств. Пятиконечная звезда может представлять языческий символ, показывать степень пряности кушанья, быть оценкой хорошо выполненной работы или символизировать генеральский чин*

## Искажение: не доверяйте своему мозгу

Распространено неверное мнение, что то, что мы вызываем из памяти, надежно и непогрешимо. Многолетние исследования памяти показали нам, что наша память не абсолютно надежна. Воспоминания о времени и его оценках, подобно любой другой информации, которую мы можем держать в памяти, не являются неподдающимися искажению. Память обычно рассматривается как трехстадийный процесс – кодирование, хранение и извлечение, – и на каждой стадии информация восприимчива к искажению (рис. 2.5). Во время кодирования физические недостатки или физиологические ограничения (например, близорукость) могут привести к неточному свидетельствованию о событии. В процессе хранения информация подвержена забыванию (или затуханию, как это называют исследователи памяти) со временем, а также другим искажениям. В отношении извлечения из памяти запомненной информации,



*Рис. 2.5. Обычно считается, что процесс запоминания состоит из трех стадий: кодирование, хранение и извлечение. На каждой стадии информация о пережитом подвергается искажению*

работа Э. Лофтус (E. Loftus) показала, что человеческие воспоминания и рассказы о вспоминаемых событиях сильно подвержены тонким воздействиям, таким как формулировка вопроса, используемая для сигнала к воспоминанию<sup>1</sup>.

Типы искажения времени по числу не уступают типам обычно задаваемых вопросов о времени: *длительность* (сколько времени нечто продолжалось), *момент времени* (когда нечто произошло) и *последовательность* (что случилось вначале и т. д.). Последовательность включает два или больше моментов времени, и эти значения (например, когда нечто началось или закончилось) являются, по существу, типом длительности, потому что вы в действительности сравниваете истекшее время или оставшееся время между моментом времени и другой значимой временной точкой. Рассмотрим следующие вопросы: когда был выключен компьютер? Когда будет выпущена бета-версия приложения? Хотя они оба выглядят как вопросы о единственной временной точке, ответы на них неизбежно содержат сравнение с другой временной точкой, обычно с текущим временем. Всякий раз при сравнении двух временных точек вы в действительности имеете дело с длительностью. Длительности могут претерпевать искажения только двух типов: получать *завышенную* или *заниженную* оценку.

<sup>1</sup> В классическом эксперименте с памятью Лофтус показывала студентам фильм об автомобильной катастрофе и позднее просила их оценить скорости автомобилей. Оценки скоростей автомобилей различались в зависимости от способа задания вопроса. Для группы, у которой спросили: «Насколько быстро двигались автомобили, когда они столкнулись друг с другом?», средняя оценочная скорость составила 34 мили в час. Для группы, у которой спросили «Насколько быстро двигались автомобили, когда они разбились друг о друга?», оценка составила 40,5 миль в час.

## Толерантность: оцениваемое время

Исследования показывают, что, когда людей просят вспомнить длительность недавно происходившего, малые длительности обычно переоцениваются, в то время как большие получают заниженную оценку. Этот эффект в литературе об оценивании времени известен как закон *Фирордта*. Однако большая часть опытов по изучению восприятия времени не имеет отношения к такому состоянию пользователей, когда они активно выполняют задания или работают с UI. Большинство связанных со временем экспериментов должны исследовать состояние, когда пользователи не имеют возможности выполнять свои задания или продолжать взаимодействовать с компьютером. Оно, разумеется, лучше известно как состояние *ожидания* и до боли знакомо каждому из нас. Исследование показывает, что трое из четырех человек обычно завышают оценку времени ожидания и что это завышение может достигать 25%.

Но что такое «долго» и «коротко»? Что такое «медленно» и «быстро»? Почему одни и те же десять секунд кажутся лишь мгновением одному человеку и досадно тянутся для другого?

### Количество или качество

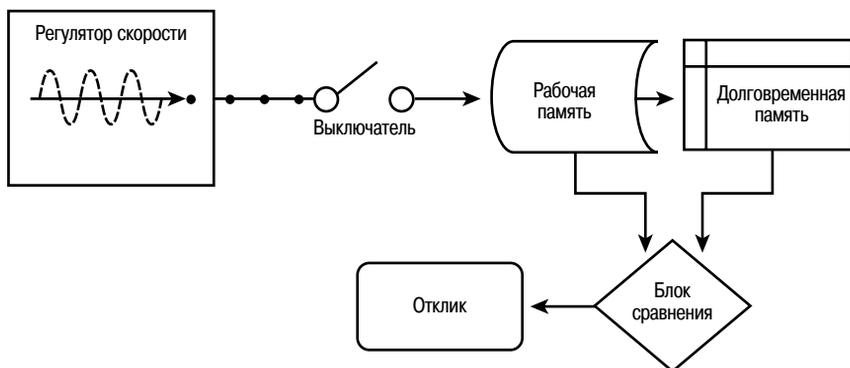
Как указано в главе 1 «Проектирование и конструирование времени», значение времени колеблется и изменяется в зависимости от многих факторов. Десятиминутное ожидание для человека, опаздывающего на 15 минут на важную встречу, мучительно. Такое же десятиминутное ожидание для человека, который уже три дня ждет доставки багажа, безразлично. Следовательно, терпимость к некоторой конкретной длительности формируется субъективной значимостью длительности. Во многих случаях мозг и секундомер функционируют одинаково, подобно часам, но важно помнить, что часы просто сообщают время и не судят о скорости или удовлетворении. То есть воспринимаемая длительность является строго *количественной* оценкой и должна быть свободной от каких бы то ни было видов *качественных* оценок. При этом воспринимаемая длительность имеет количественное выражение («Установка заняла час...»), а толерантность определяется качественно («Это слишком медленно, потому что она должна была занять 15 минут!»).

### Мысленный эталон

Картина является неполной без привязки воспринимаемой длительности к толерантности. Вспомним, что основной контингент пользователей обычно специально не хронометрирует операции и процессы в используемом компьютерном продукте, но с пристрастием относится к реальным или кажущимся задержкам. (Существует, конечно, подмножество пользователей, по разным причинам желающих оценить

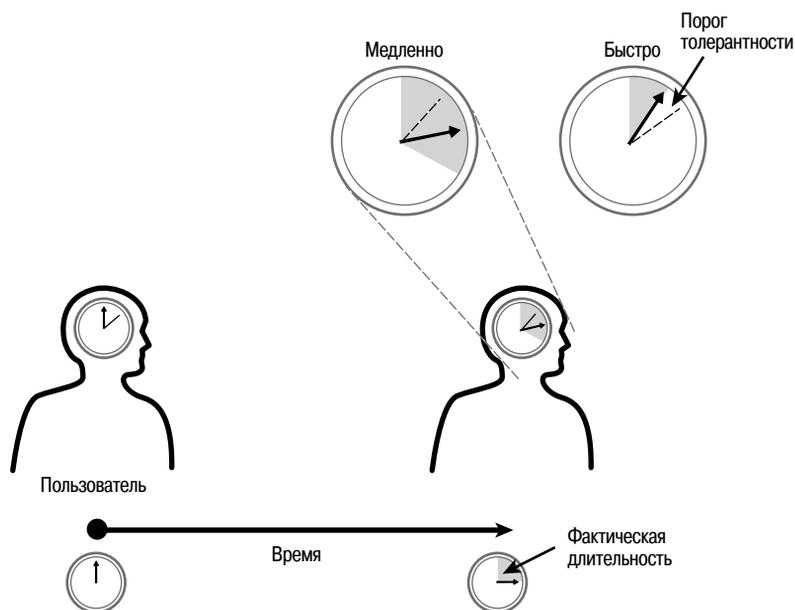
производительность и поэтому ставящих объективные эксперименты.) В любом случае для обоих подмножеств можно обнаружить общий фактор: все длительности не несут никакой смысловой нагрузки в отсутствие предмета для сравнения. Например, простое утверждение, что теннисный мяч имеет диаметр 2 дюйма, является фактическим, количественным и описательным и при этом лишенным иного смысла. Если я добавляю, что, поскольку мяч имеет диаметр 2 дюйма, я не могу использовать для теннисного корта ограждение с отверстиями размером 3 дюйма, то знание диаметра приобретает смысл.

Аналогично, какой-то вид эталона требуется, чтобы значение реальной длительности обрело смысл. В этом нет отличия от воспринимаемой длительности: в отсутствие сопоставления с предшествующим опытом или аналогами воспринимаемое значение не несет никакой смысловой нагрузки. Фактически количественные модели чувственного оценивания содержат компонент, представляющий долговременную память. Одна такая модель, предложенная Гиббоном (Gibbon), Черчем (Church) и Меком (Meck), показана на рис. 2.6. В компьютерном контексте, если пользователь считает, что нечто является медленным или быстрым, вы всегда можете сопоставить текущее восприятие с чем-то, хранящимся в памяти, например с предыдущим опытом, ожиданиями и т. д. Эти эталоны восприятия дают пользователю возможность решить, как охарактеризовать воспринимаемую длительность. Например, если пользователь помнит, что конкретное компьютерное приложение обычно загружается полностью в течение примерно пяти секунд, но однажды для запуска потребовалось удвоенное время, пользователь замечает разницу и подозревает что-то нехорошее.



**Рис. 2.6.** Многие модели оценивания времени, например модель Гиббона, Черча и Мека, включают компонент, который представляет долговременную память, сохраняющую запомненную длительность. В этой модели содержимое долговременной памяти сравнивается с тем, что воспринимается в режиме реального времени в рабочей памяти, и отклик создается на основании сравнения

То, что присутствует в памяти в качестве ожидаемого значения, устанавливает *порог толерантности*, и если воспринимаемая длительность его превышает, то нечто трактуется как медленное. Напротив, если воспринимаемая длительность оказывается ниже порога, то нечто трактуется как быстрое (рис. 2.7). В отличие от сравнения фактической и воспринимаемой длительности, здесь речь идет о том, как пользователь *ощущает* длительность по отношению к тому, о чем он думает или что знает. Эта идея порога толерантности аналогична идее *порога негодования*, упомянуемого П. Саффо (P. Saffo) в работе «Bringing Design to Software». Вы вполне можете рассматривать первую как временной вариант последней, которая определяется Саффо как «максимальный поведенческий компромисс, на который мы идем, чтобы задача была выполнена».



**Рис. 2.7.** Воспринимаемая длительность несет смысловую нагрузку только тогда, когда она сопоставляется с порогом толерантности. Если воспринимаемая длительность меньше порога толерантности, пользователь интерпретирует происходящее как быстрое. Напротив, если длительность воспринимается как превышающая порог толерантности, пользователь интерпретирует происходящее как медленное

## Временные факторы, влияющие на толерантность

Очевидными факторами, влияющими на толерантность, являются те, которые имеют временные измерения, – те, которые описываются явными показателями времени. Благодаря тому что большинство этих факторов достаточно очевидно, они легко определяются несложными наблюдениями, измерениями или исследованиями.

## Использование и опыт

Несмотря на то что основной контингент пользователей обычно не хранит подробные данные о том, насколько быстро компьютер выполняет те или иные задачи, память о длительностях, сформировавшаяся благодаря неоднократному использованию и опыту, может быть хотя и не очень точной, но достаточно надежной для практических целей. Возьмем, к примеру, загрузку компьютера. Нередко пользователи нажимают кнопку питания для включения компьютера, уходят, переключают внимание на другое и затем, некоторое время спустя, возвращаются, надеясь, что компьютер готов к работе. Ввиду того что такие задачи выполняются часто, задержки при загрузке, превышающие привычные (например, после модернизации аппаратного обеспечения, загрузки другого компьютера с такой же конфигурацией), начнут снижать толерантность. В случае другой крайности недостаточное использование или опыт также могут нанести ущерб толерантности.

## Стандарты и эталоны

Когда конкретные длительности не соответствуют опубликованным или заявленным показателям, это, естественно, оказывает негативное влияние на порог толерантности. Более важно осознавать, что эти показатели могут быть установлены *любым* источником и для обычного пользователя доверие к этим числам основывается в большей степени на общей популярности, чем на науке и технике. Например, если популярный источник (например, журнал *Wired*) сообщает, что, по результатам исследования, большинство хороших веб-сайтов загружаются полностью менее чем за две секунды, то возможно, что эти числа станут фактическими стандартами для многих читателей. То, что показатели получены эмпирическим путем, основной контингент пользователей обычно не ставит под сомнение. В таком случае становится очень важным понять, что это за стандарты или эталоны, как они определены, насколько интересующий пользователя продукт или сервис им соответствует и, при необходимости, какие корректирующие измерения можно провести.

## Объекты для сравнения

Как было отмечено ранее, пользователи не могут ощутить, является ли некий процесс быстрым или медленным, пока не сопоставят его с объектом сравнения. Для большинства пользователей объектами сравнения, вероятнее всего, будут служить одна и та же характеристика для различных версий и вариантов (например, веб-версия и версия для настольного компьютера) продукта, различные характеристики того же продукта, та же характеристика другого продукта или различные характеристики других продуктов. Табл. 2.1 содержит несколько примеров. Установить, что пользователи используют в качестве объектов сравнения, можно различными способами (фокус-группы, наблюдения,

интервью и т. д.), однако главной целью должно быть определение того, что это за объекты для сравнения, а затем, при необходимости, принятие мер противодействия, если пользователи оценивают характеристику, службу или продукт «недобросовестно», производя сравнение с *несравнимым* объектом сравнения.

Таблица 2.1. Простые примеры объектов сравнения

	Тот же продукт или служба	Сравнимый продукт или служба
<b>Та же характеристика или функция</b>	Добавление Избранного в Internet Explorer 6 и Добавление Избранного в Internet Explorer 7	Выполнение интернет-поиска в Google и Выполнение интернет-поиска в Yahoo!
<b>Другая, но сравнимая характеристика или функция</b>	Копирование и вставка текста в документ и Копирование и вставка изображения в документ	Просмотр потокового видео в Интернете и Просмотр видео из загруженного из Интернета видеофайла

## Индикация в пользовательском интерфейсе

Иногда пользователи теряют толерантность и терпение просто потому, что отсутствует ясное указание на то, сколько будет продолжаться процесс. Например, UI показывает, что некоторый процесс займет не более пяти минут, но уже прошло семь. Для того чтобы понять, какова длительность процесса, пользователи, кроме текстовой информации, опираются на графический индикатор выполнения (progress bar). Когда длительность превышает то значение, которое отображается, толерантность стремительно падает. Недостаточная информация заставляет людей быть нетерпеливыми, но и ясная информация, которая оказывается неточной, также оказывает неблагоприятное влияние на толерантность пользователей.

## Вневременные факторы, влияющие на толерантность

Общепризнано, что существуют вневременные факторы, влияющие на толерантность. Это значит, что они не содержат временных показателей, но управляют тем, с какой длительностью процесса пользователи могут согласиться. При сопоставлении с факторами, упоминаемыми в предыдущем разделе, эти факторы, которые могут представлять комфортно размеченный контекст, являются менее объективными и, тем не менее, важными при оценивании толерантности пользователей.

## Повторяющиеся попытки, неоднократные сбои

Большинство пользователей не сдаются при первом сбое выполнения задачи, потому что склонны не придавать значения случайной ошибке в работе системы и предпринять несколько новых попыток выполне-

ния. Возьмем, к примеру, Интернет. Обычно пользователи не воздевают руки к небу и не сдаются сразу после того, как браузер сообщает об ошибке загрузки номер 404. Вероятно, они попробуют открыть веб-сайт заново, повторить ввод URL или щелкнуть по кнопке Обновить или Перезагрузить. Если бы пользователи не были толерантны к ошибкам системы, то они бы закрывали браузер и заявляли, что продукт неполноценный или плохо функционирующий, сразу же, при первой ошибке номер 404. Благодаря тому что основной контингент пользователей компьютеров толерантен и адаптивен, те, кто создает программное обеспечение, могут выйти сухими из воды, включив в веб-браузер кнопку Обновить или Перезагрузить, а те, кто создает аппаратное обеспечение, могут достигнуть того же при помощи кнопки Сброс!

Однако следует сделать предостережение. Такая исключительная «мягкость пользователя», по-видимому, держится по крайней мере на двух факторах. Первый связан с воспринимаемой надежностью и стабильностью продукта. Некоторые продукты воспринимаются как высоконадежные и считаются стабильными, и включение в их состав кнопки или функции, осуществляющей сброс, может показаться странным. Например, телевизор поступает к конечному пользователю без кнопки сброса или обновления, потому что считается бытовым электроприбором, который обязан нормально работать всегда. Аналогично, определенные компьютерные продукты воспринимаются как более стабильные и надежные, чем другие, то есть менее предрасположенные к ошибкам. Например, пользователи более толерантны к ошибкам в процессе перехода к веб-странице, чем при вводе текста в текстовом редакторе. Аналогично, пользователи более толерантны к сбоям в работе бета-версии продукта, чем в законченном продукте. По существу, мягкость пользователя варьируется в функции от предполагаемой стабильности, поэтому пользователи менее требовательны к процессам, от которых не ожидается высокой стабильности.

Вторым фактором, который может влиять на мягкость пользователя, является воспринимаемая уверенность пользователя в том, что он знает, почему нечто не работает. Например, пользователи более толерантны, если недавно купленный мобильный аппарат не синхронизируется с их персональным компьютером сразу же, поскольку пользователь, вероятно, подумает, что существует множество причин отказа синхронизации: устройство не сконфигурировано, устройства Bluetooth недоступны, на персональном компьютере не установлено необходимое программное обеспечение и т. д. Однако при возрастании его уверенности в некоторых фактах (устройства Bluetooth доступны, поскольку голубой индикатор мигает, программное обеспечение установлено на персональном компьютере в соответствии с инструкцией или все работало с тех пор, как было включено месяц назад, и т. д.) толерантность к сбоям быстро убывает по мере повторения неудачных попыток, поскольку каждая из них ухудшает мнение об аппарате или приложении.

## Время дня, день недели

Для многих современных и индустриальных сообществ решающую роль в регулировании толерантности пользователей могут играть время дня и день недели (и, возможно, месяц года). Например, одно исследование показало, что многие посетители ресторанов быстрого питания менее терпеливы во время ланча, чем в обеденное время. Для них ограничение времени ланча (например, из-за необходимости возврата на работу), очевидно, играет первостепенную роль. День недели также может оказывать большое влияние на толерантность, что, к примеру, отражает крылатая фраза «Понедельник – день тяжелый». Есть основания предположить, что многие офисные работники имеют определенные приоритеты, относящиеся к возвращению в офис в начале недели. Для многих, кто организует свои дела и задачи с помощью компьютера (например, составление расписания и отслеживание встреч с помощью календаря в приложении электронной почты или сохранение информации о прохождении заказов на покупки во внутреннем приложении), восприятие соответствующей информации имеет высший приоритет в начале недели, и поэтому толерантность к ожиданию возможности работы с информацией невысока, хочется получить ее как можно скорее. Если приложение связано с последовательностью процессов (загружаемых всякий раз), отдаляющих пользователя от получения информации (или, еще хуже, сообщает пользователю, что требуется время для выполнения каких-то непонятных регламентных работ), толерантность будет убывать с каждой секундой. В сущности, воспринимаемая срочность и приоритет задачи изменяют порог толерантности.

## Эмоциональное состояние

Исследования неоднократно подтверждали, что в особой эмоциональной или стрессовой обстановке восприятие сильно подвержено искажению. Например, некоторые программные продукты созданы для использования в средах, которые заведомо являются стрессовыми в большей степени, чем остальные. Так, использование текстового процессора для написания отчета и сенсорного экрана терминала розничной торговли в ресторане в часы пик представляют собой весьма различные занятия для студента колледжа, подрабатывающего в ресторане. Аналогично, поиск в Интернете свежих слухов о знаменитостях и получение информации в полицейском автомобиле о подозрительном водителе – это задачи, очень по-разному воспринимаемые одним и тем же полицейским, желающим быть в курсе голливудских новостей. Поэтому имеет смысл присмотреться к сценариям использования продукта или сервиса и определить, могут ли быть произведены какие-то модификации, для того чтобы улучшить впечатление пользователя. Приложение, созданное для представления информации, к примеру, может быть оптимизировано для использования во время ланча путем исключения утомительных и ненужных действий. Приложение, предназначенное

для получения информации и установленное на портативном компьютере в полицейском автомобиле, должно предоставлять полицейскому возможность быстро получать существенную и точную информацию без препятствий, создаваемых бессмысленными диалогами и вопросами.

## Предубеждение

По различным причинам пользователь может обладать низкой толерантностью к процессу независимо от того, насколько процесс оптимизирован. Рассерженные офисные работники, проинформированные сотрудниками о том, что новое приложение, размещенное в Интернете, заменило старое, размещенное на стороне клиента (которое каждый получил в личное пользование), могут, естественно, иметь низкую толерантность к новой реализации – типична реакция «Я ненавижу эту новую систему». Репутация и имена торговых марок некоторых продуктов также могут вызвать раздражение пользователей. Иногда репутация продукта может опередить сам продукт, например, благодаря обзорам, отзывам потребителей, одним словом, слухам. В других случаях может сыграть роль классическое приписывание вины тому производителю, чья продукция предвзято считается ущербной. В таких случаях некоторые пользователи могут пытаться мысленно усилить недостатки и ограничения продуктов для поддержания своего предубеждения. Обратная сторона медали заключается в противоположном *эффекте ореола*, когда пользователи полагают, что продукт обладает желаемым качеством, потому что имеет общую с их любимым продуктом торговую марку.

## Культура, мода, прихоти

В то время как предубеждение работает на уровне индивидуального пользователя, факторы, действующие в более широком масштабе, также могут влиять на толерантность отдельного человека. А именно, важную роль в формировании толерантности к продукту могут играть мода и прихоти. Вполне обоснованные претензии к продукту (сложно пользоваться, громоздкий, слишком мелкий, слишком крупный, слишком медленный и т. д.) могут быстро стать спорными, если сообщество сверстников утверждает, что этот продукт – последний писк моды и им обязательно надо пользоваться. Прекрасным примером может служить Newton, один из ранних карманных компьютеров, потерпевший на рынке полный провал. Через несколько лет после его кончины стало модно иметь именно его, вопреки тому факту, что размеры и базовая функциональность Newton и новых поколений карманных компьютеров были одинаковыми. Воздействие моды и прихотей тесно связаны с культурой, потому что некоторые культуры представлены не только потребителями, вдумчиво сравнивающими качество продуктов, но и многими молодыми приверженцами, желающими испробовать новые технологии. В Сингапуре, например, обмен текстовыми сообщениями по

мобильному телефону был одной из наиболее распространенных среди младшего поколения прихотей начиная с начала 1990-х годов, несмотря на очевидную неуклюжесть и медленный ввод символов с помощью цифровой клавиатуры телефона – кошмар, который ужаснет любого дизайнера, заботящегося об удобстве использования.

## Выводы

Никогда нельзя полагать, что восприятие фактической длительности является точным и правильным. В то время как фактическая длительность отражает объективное время, воспринимаемая длительность отражает субъективное психологическое время, подверженное искажениям различного уровня. Помимо различий фактической и воспринимаемой длительности, следует учитывать толерантность пользователя, на которую воздействуют как временные (например, эталоны и стандарты), так и вневременные факторы (например, предубеждение и мода).

## Кроличья нора

### Управление восприятием

Geelhoed, E., P. Toft, S. Roberts and P. Hyland. To influence time perception. CHI'95 Mosaic of Creativity, 1995. 272–273.

Hui, M. K. and D. K. Tse. What to tell consumers in waits of different lengths: An integrative model of service evaluation. *Journal of Marketing*, 60, 1996. 81–90.

Katz, K., B. Larson and R. Larson. Prescriptions for the waiting-in-line blues: entertain, enlighten, and engage. *Sloan Management Review*, Winter, 1991. 44–53.

Kellaris, J. J. and R. J. Kent. The influence of music on consumers' temporal perceptions: Does time fly when you're having fun? *Journal of Consumer Psychology*, 1, 1992. 365–376.

Leclerc, F. How should one be told to hold? *Advances in Consumer Research*, 28, 2002. P. 78.

North, A. C., D. J. Hargreaves and J. McKendrick. Music and on-hold waiting time. *British Journal of Psychology*, 90, 1999. 161–164.

Oakes, S. Musical tempo and waiting perceptions. *Psychology & Marketing*, 20, 2003. 685–705.

Sasser, W. E., M. Olsen and D. D. Wychoff. *The Management of Service Operations*. Boston: Allyn & Bacon, 1978. (Примечание автора: зеркала, установленные около эскалаторов, сокращают воспринимаемое время ожидания!)

## Воспринимаемое время

Allan, L. G. The perception of time. *Perception & Psychophysics*, 26, 1979. 340–354.

Eisler, H. Subjective duration and psychophysics. *Psychological Review*, 82, 1975. 429–450.

Fraisse, P. Perception and estimation of time. *Annual Review of Psychology*, 35, 1984. 1–36.

Hicks, R. E., G. W. Miller and M. Kinsbourne. Prospective and retrospective judgments of time as a function of amount of information processed. *The American Journal of Psychology*, 89, 1976. 719–730.

Loehlin, J. C. The influence of different activities on the apparent length of time. *Psychological Monographs*, 73, 1959. 1–27.

Loftus, E. F. Time went by so slowly: Overestimation of event duration by males and females. *Applied Cognitive Psychology*, 1, 1987. 3–13.

Merikle, P. M. and S. Joordens. Parallels between perception without attention and perception without awareness. *Consciousness and Cognition*, 6, 1997. 219–236.

Thomas, E. A. C. and W. B. Weaver. Cognitive processing and time perception. *Perception & Psychophysics*, 17, 1975. 363–367.

## Классические эксперименты с памятью

Loftus, E. F. and J. C. Palmer. Reconstruction of automobile destruction. *Journal of Verbal Learning and Verbal Behavior*, 13, 1974. 585–589.

Miller, G. A. The magical number seven, plus or minus two: Some limits on out capacity for processing information. *The Psychological Review*, 63, 1956. 81–97. (Примечание автора: одна из самых выдающихся работ на тему памяти. Имеется в Сети по адресу [www.musanim.com/miller1956/](http://www.musanim.com/miller1956/).)

## Пользовательская толерантность

Dellaert, B. G. C. and B. E. Kahn. How Tolerable Is Delay? Consumers' evaluations of internet websites after waiting. *Journal of Interactive Marketing*, 13, 1999. 41–54.

Goodin, R. E., J. M. Rice, M. Bittman and P. Saunders. The time-pressure illusion: Discretionary time vs. free time. *Social Indicators Research*, 73, 2005. 43–70.

Hancock, P. A. and J. L. Weaver. On time distortion under stress. *Theoretical Issues in Ergonomics Science*, 6, 2005. 193–211.

Osuna, E. E. (1985). The psychological cost of waiting. *Journal of Mathematical Psychology*, 29. 82–105.

Pruyn, A. and A. Smidts. Customers' reactions to waiting: Effects of the presence of "fellow sufferers" in the waiting room. *Advances in Consumer Research*, 26, 1999. 211–216.

Saffo, P. The consumer spectrum. In *Bringing Design to Software*. T. Winograd (Ed.). Reading, MA: Addison-Wesley, 1996. 87–99. (Примечание автора: прочтите раздел «Threshold of Indignation».)



# Глава 3

## Время отклика пользователя и системы

Задолго до того, как персональные компьютеры завоевали популярность, исследователи знали, что распределение времени между командами пользователя и откликом системы может быть решающим фактором в формировании наилучшего пользовательского впечатления. Во многих случаях человеко-машинное взаимодействие похоже на беседу. В этой главе вводятся понятия времени отклика как пользователя, так и системы и обсуждаются различные промышленные стандарты на максимально допустимые значения времени отклика системы.

### Беседа кремния с углеродом

Дэйв хотел, чтобы его компьютер всего-навсего открыл дверь, но компьютер не подчинился. «Извини, Дэйв, боюсь, я не смогу этого сделать», – сухо сказал компьютер по имени ХАЛ. Озадаченный Дэйв потребовал объяснений: «В чем проблема?» Ответ ХАЛа прозвучал более зловеще: «Думаю, ты знаешь, в чем проблема, не хуже меня».

Это была незабываемая сцена из фильма 1968 года «2001: Космическая Одиссея». Несмотря на то что мы все еще не можем создать компьютер, способный функционировать подобно ХАЛу (и, вероятно, не хотим этого), мы находимся в постоянном взаимодействии с компьютером. Иногда мы сообщаем информацию компьютеру, иногда компьютер предоставляет ее нам. Благодаря тому что поток информации при человеко-машинном взаимодействии является двунаправленным, мы имеем взаимодействие, *диалоговое* по своей природе.

Как и в беседе человека с человеком, здесь существуют правила этикета. Особенно большое значение в диалоге человека и компьютера играет распределение времени. Например, в беседе человека с человеком пауза при ответе на простой вопрос может иметь разнообразные интерпретации и подтекст. Рассмотрим следующий диалог между Гомером и Мардж Симпсон в эпизоде из культового комикса «Симпсоны»:

**Гомер:** Мардж, ты уважаешь мой интеллект?

(четырьмя секундами позже)

**Мардж:** Да.

**Гомер:** Хорошо!.. Подожди-ка. Почему ты так долго не отвечала?

(четырьмя секундами позже)

**Мардж:** Просто так.

**Гомер:** Хорошо!.. Подожди-ка. Ты надо мной смеялась?

(четырьмя секундами позже)

**Мардж:** Да.

**Гомер:** Хорошо!.. Подожди-ка. Это плохо.

Четырехсекундные паузы могут быть интерпретированы как неуверенность, потому что Мардж не должна была тратить так много времени для ответа на вопрос Гомера. Аналогично, когда компьютер тратит несколько секунд для того, чтобы дать отклик на простую команду, пользователь, естественно, ощущает, что с компьютером что-то не так.

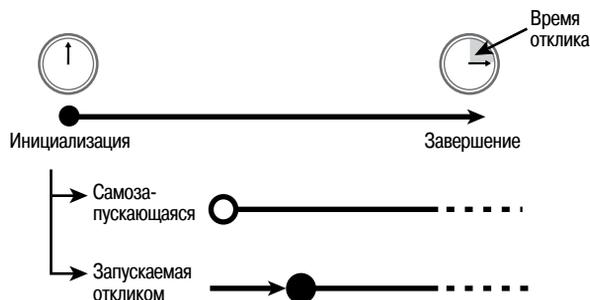
## Определение времени отклика

Где-то в конференц-зале в Силиконовой долине происходит процесс принятия решения о том, насколько быстро должна выполняться одна из функций некоего решения. После долгих дебатов и споров кто-то замечает, что для времени реакции человека, как учили на курсе психологии колледжа, справедливо значение 200 мс. Эта крупница знания, кажется, приведет дебаты к благополучному завершению, о чем можно судить по многочисленным одобрительным кивкам. Две сотни миллисекунд. Вопрос закрыт.

Что неверно в этом сценарии и выводе? Ответ короток: здесь, возможно, произошла путаница между двумя весьма различными понятиями о распределении времени. Если здесь не обсуждается зависящее от времени или критически важное решение, то вопрос, насколько быстро должна выполняться некоторая функция, никак не связан с тем, как быстро реагирует пользователь.

На простейшем уровне время отклика можно определить просто исходя из времени начала и завершения процесса или взаимодействия. Начало любого процесса или взаимодействия может быть инициировано автоматически, например плановым извещением или в ответ на такую команду, как нажатие клавиши пользователем. В любом случае при

этом как бы «запускается секундомер». Как правило, сам отклик является наблюдаемым действием или результатом, который останавливает секундомер, отмечая время окончания отклика (рис. 3.1). В некоторых случаях более точной является фиксация окончания процесса по завершении формирования результата. Например, начальную загрузку текста веб-страницы можно принять за момент окончания загрузки веб-страницы лишь с натяжкой. Завершение процесса наступает, когда страница оказывается загруженной целиком и воспринимается пользователем как готовая к просмотру.



*Рис. 3.1. Время отклика может быть установлено автоматически или в ответ на команду. В любом случае время отклика характеризуется временем начала и временем завершения*

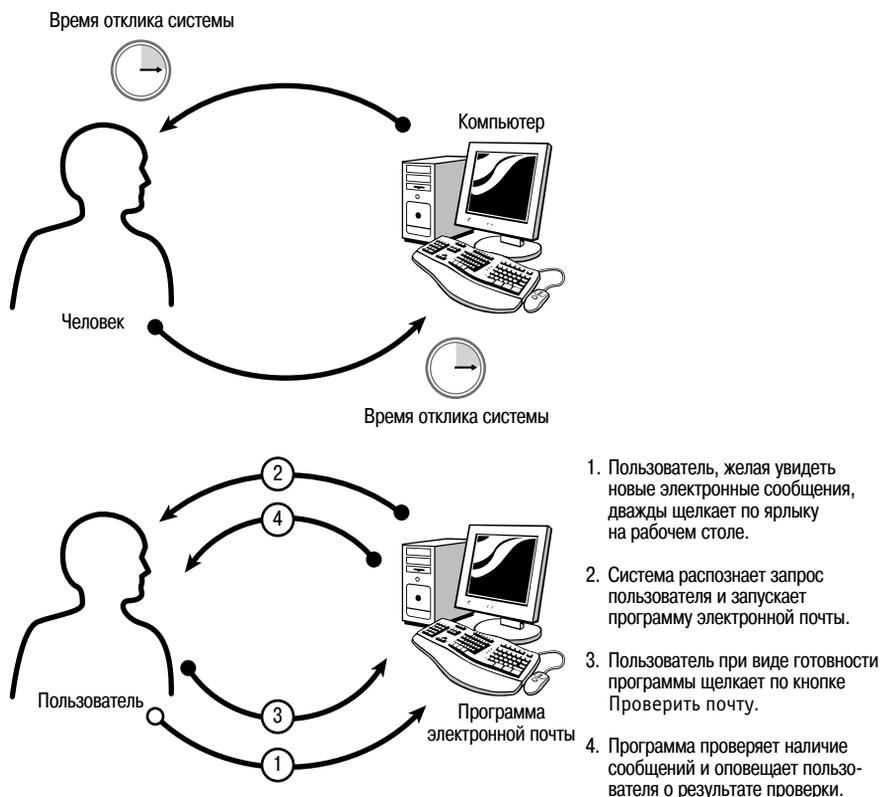
Если пользователь является тем, кто запускает или инициирует процесс (например, нажатием клавиши), то время от ожидания пользователем возможности запуска процесса до момента, когда будут выполнены поддающиеся наблюдению действия, называется *временем отклика пользователя*. Как уже отмечалось, такая инициализация может быть либо самозапускающейся (например, включение компьютера по утрам), либо запускающейся в ответ на внешние факторы (например, выбор элемента отображаемого на экране списка). Определение точного времени старта для самозапускающейся инициализации является произвольным, потому что имеется множество мыслительных процессов, не поддающихся наблюдению, – принятие решения, распознавание и т. д., – которые участвуют в подготовке результата, реально наблюдаемого. В большинстве случаев вы будете иметь дело с тем, что инициализация проявляется как ответ на что-то, представляемое компьютером.

В других случаях имеют место процессы, инициируемые системой. Опять-таки, они могут быть самозапускающимися (например, при поступлении планового требования о загрузке обновлений) или, гораздо чаще, запускаемыми пользователем (например, щелчком кнопкой мыши для запуска приложения). Между инициализацией и завершением компьютер тратит определенное время на обнаружение пользовательского ввода (щелчок кнопкой мыши), затем выполняет некоторые функции или демонстрирует некоторое поведение (поиск в базе данных) и наконец достигает состояния, когда процесс завершен (вывод результатов).

Время, прошедшее между крайними моментами, чаще всего называют *временем отклика системы*. Это согласуется с определением Б. Шнайдермана (B. Shneiderman)<sup>1</sup>: «*Временем отклика системы* является количество секунд, прошедших от момента инициализации действия пользователем (обычно нажатие клавиши Enter или Return) до момента, когда компьютер начинает представлять результаты на экране или принтере».

## Диалог

Для основного контингента пользователей типичное взаимодействие с компьютером включает сочетание ввода информации в компьютер (ввод данных, команд, выбор опций и т. д.) и получения последовательности ответов от компьютера (отображение результатов, анимация, решение числовых задач и т. д.). Это иллюстрирует рис. 3.2. По всей



**Рис. 3.2.** Вверху: человеко-машинное взаимодействие как процесс диалога. Внизу: пример того, как происходит типичное взаимодействие

<sup>1</sup> Бен Шнайдерман – ученый, профессор университета Мериленд, специалист по компьютерным интерфейсам. – *Примеч. науч. ред.*

вероятности, эти действия перекрываются и пользователи уже обдумывают и планируют ряд последующих шагов, в то время как компьютер еще отвечает на первую команду.

Представьте, что вам надо отправить сообщение с помощью своей программы электронной почты. Вероятно, прежде чем вы прочитали первое слово этого предложения, в вашем мозгу уже сложился список надлежащих шагов для выполнения поставленной задачи. В то время как начинающий пользователь может применять безопасный подход «шаг-за-шагом» по принципу «рецепта из поваренной книги», что обеспечивает безошибочное выполнение последовательности шагов, приводящее к выполнению задачи (как, например, при использовании возможности уменьшения эффекта красных глаз в программе редактирования фотографий), опытный пользователь, скорее всего, мысленно сосредоточится на конечной цели и будет идти к ней, пользуясь информацией о шагах, которую он запомнил ранее. Большинство пользователей принадлежат к этой последней группе.

## Время отклика пользователя

Вернемся к дебатам в Силиконовой долине. Людям требуется около 200 мс, для того чтобы дать простой отклик на представление простого раздражителя. Это означает следующее. Предположим, что некоего человека попросили нажимать клавишу сразу, как только возникает точка на экране. Среднее время, которое проходит между появлением точки и нажатием клавиши, составит около пятой доли секунды. Эта характеристика широко известна как *время простой реакции* и обычно измеряется в лаборатории с применением простейших видов отклика, например нажатия кнопки пальцем.

Несмотря на то что более сложные отклики требуют больше движений, вследствие чего время реакции увеличивается, некоторые сложные отклики при тренированности могут быть выполнены за время, близкое к пятой части секунды. Чемпион по быстрой стрельбе, например, может вынимать свое оружие из кобуры и делать выстрел примерно за 200 мс. Для простых смертных 200 мс – это то время, за которое мы можем выполнить простое наблюдаемое действие в ответ на появление какого-то раздражителя (звук, объект и т. д.) во внешнем мире. Это объясняет, почему участникам телевизионных игр лучше держать руки на автоматическом прерывателе, чем в карманах!

Если задача требует большего обдумывания, например требуется дать отклик на появление красной точки, но не давать его на появление синей, то реакция замедляется. Утверждение, что *время реакции выбора* возрастает с увеличением числа возможностей выбора, было впервые исследовано в 1800-х годах, а позднее было научно промоделировано и сформулировано как закон Хика–Хаймана, который, по существу, описывает модель поведения человека, где люди рассматриваются как

информационные процессоры. Как и другие виды процессоров, информационные процессоры имеют ограничения. Проще говоря, когда имеется большой объем информации, которую надо обработать, процессор тратит больше времени. Упрощенное представление об этой концепции можно получить при сравнении наполнения бассейна водой с помощью садового и пожарного шлангов. Первый имеет маленькую пропускную способность, и следовательно, потребуется больше времени для наполнения бассейна. Эксперимент Хика<sup>1</sup> показывает, что при возрастании числа возможностей выбора от двух до десяти<sup>2</sup> время реакции увеличивается примерно с 200 до 500 мс.

## Соотношение между скоростью и точностью

Одна из причин важности понимания смысла времени отклика пользователя выражается в *соотношении между скоростью и точностью*. Когда люди стараются справиться с задачей как можно быстрее, они делают массу ошибок. И наоборот, когда люди пытаются сделать как можно меньше ошибок, страдает скорость. Это соотношение хорошо изучено в психологии. В 1952 году в исследовании Хика, к примеру, в одной серии испытаний участникам следовало давать ответы быстро, насколько это возможно, а в другой серии им надо было давать как можно более точные ответы. В первой серии испытаний участники были относительно проворны, но делали много ошибок. Во второй общая точность была выше, но ответы давались относительно медленно.

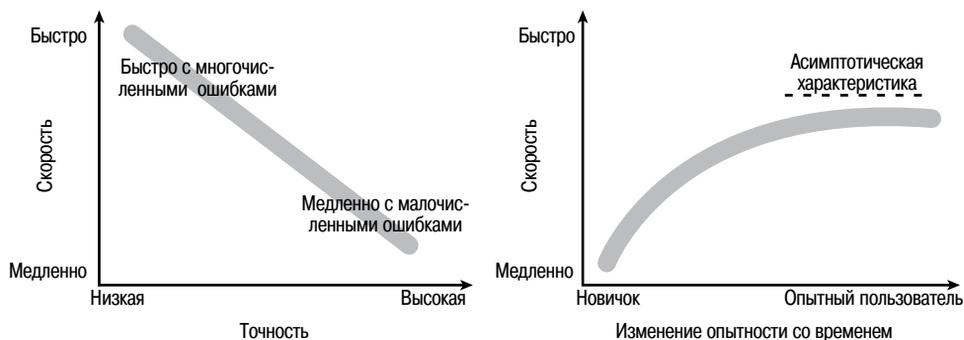
Уменьшению времени человеческого отклика (проявлению более быстрой реакции) способствует множество факторов. Обычная практика уменьшает время человеческого отклика. Это уменьшение времени отклика обычно наблюдается как повышение производительности и описывается *степенным законом научения*, согласно которому люди медленно выполняют новые для себя задачи, но постепенно, по мере их выполнения, улучшают свои временные характеристики. Это значит, что если вы изобразите временную характеристику на графике, то сможете увидеть кривую начального обучения, за которой следует плоский участок, свидетельствующий о стабилизации навыка. Возвращаясь к соотношению скорость/точность, вероятно, можно утверждать,

---

<sup>1</sup> Хик Вильям Эдуард (1912–1975) – английский врач, психотерапевт, психолог. Работал в Кембриджском университете по проблемам экспериментальной психологии. В 1952 г. совместно с Р. Хайманом вывел закон времени реакции – *Примеч. науч. ред.*

<sup>2</sup> В исследовании Хика информация измерялась в битах. Пользователи должны были делать выбор в условиях возрастающего количества возможностей выбора. Испытание с двумя возможностями выбора представляло один бит, так как пользователь делает выбор из двух элементов. Испытание с четырьмя возможностями выбора представляло два бита и т. д.

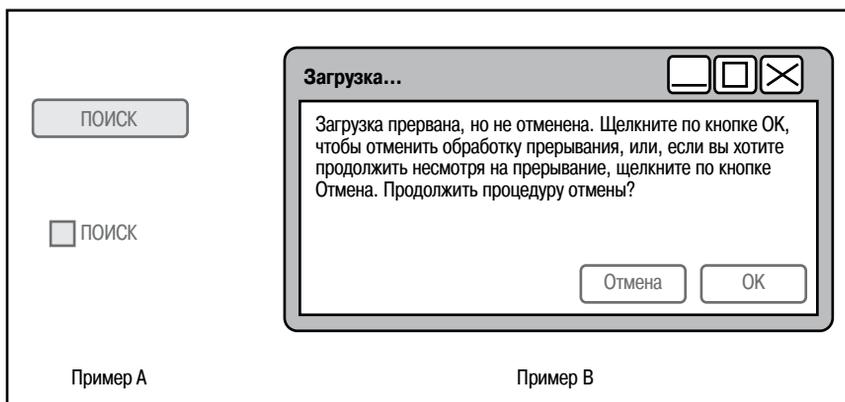
что точка выравнивания представляет область комфорта, где люди ощущают свои успехи в выполнении определенных задач как по скорости, так и по точности. Рис. 3.3 иллюстрирует соотношение скорость/точность и степенной закон научения.



**Рис. 3.3.** Слева: соотношение скорость/точность. Когда внимание сосредоточено на скорости, страдает точность, и наоборот. Справа: степенной закон научения. Со временем длительность выполнения нового задания уменьшается

Если от пользователей требуют выполнять некоторую задачу за определенное время, используя UI, то начинают появляться ошибки. Если только это не игры (где установка временных ограничений делает игру увлекательной) и не приложения, критические по времени (например, медицинские или военные системы), будьте осторожны, устанавливая пользователям временные ограничения, особенно в ситуациях, когда им надо принимать решения. С другой стороны, если компьютерные программы требуют от пользователя точной и правильной информации, то пользователь снизит скорость. Например, многие графические средства управления, такие как кнопки и шкалы с ползунком, имеют довольно обширные интерактивные области, где пользователь может щелкнуть для ввода команды. Представим себе, что эти средства управления уменьшились в размерах до нескольких пикселей. При таком сценарии причина изменения времени отклика пользователя очевидна. Это простое явление сформулировано в виде широко известного закона Фиттса<sup>1</sup>. Аналогично, если система требует от пользователя принятия решения, основанного на некоторой неопределенной информации, то он неизбежно замедлит темп работы – пример закона Хика–Хаймана (рис. 3.4).

<sup>1</sup> Пол Морис Фиттс (1912–1965) – американский психолог, занимался вопросами экспериментальной и организационной психологии, пионер инженерной психологии – *Примеч. науч. ред.*

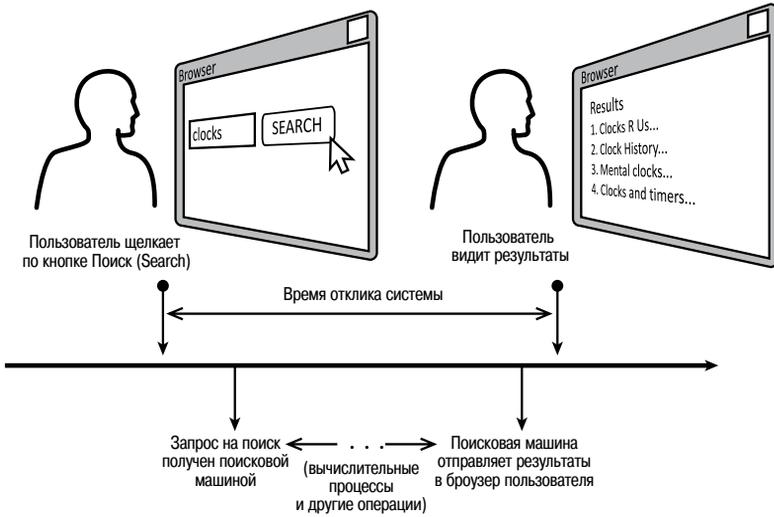


*Рис. 3.4. Слева: пример закона Фиттса. При уменьшении интерактивной области кнопки Поиск (Find), предназначенной для щелчка, время отклика пользователя увеличивается. Справа: пример закона Хика–Хаймана. При возрастании умственной нагрузки время отклика пользователя также увеличивается*

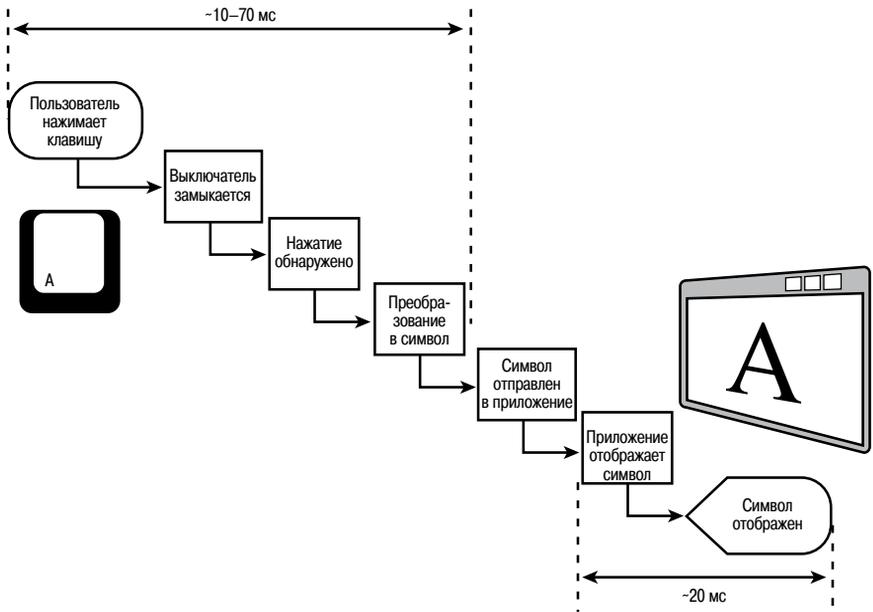
## Время отклика системы

Время отклика системы дополняет время отклика пользователя. В контексте человеко-машинного взаимодействия, время отклика системы должно определяться с точки зрения пользователя. В качестве примера представим, что пользователь вводит слово для поиска в поисковой машине и щелкает по кнопке Поиск. Для заинтересованного пользователя щелчок по кнопке означает начало процесса, а последовательно отображаемые результаты свидетельствуют о завершении процесса (рис. 3.5). Процессы на сервере и другие операции, составляющие поиск, для обычного пользователя ничего не значат. Важно лишь то, что пользователь испытывает.

При определении и вычислении времени отклика системы вы должны учитывать время, затраченное на выполнение действия, и время, необходимое пользователю для обнаружения отклика компьютера. Это относится не только к относительно заметной длительности, пример которой приведен на рис. 3.5, но и к процессам с очень малыми длительностями. Рассмотрим пример, показанный на рис. 3.6. Исследование показывает, что задержки клавиатуры могут варьироваться в диапазоне от 11 до 73 мс, поэтому вводимые символы могут отображаться не сразу. В зависимости от скорости обновления экрана монитора также возможна некоторая задержка отображения символа на экране. Поэтому учет одного только времени работы программы может оказаться недостаточным.



**Рис. 3.5.** В контексте человеко-машинного взаимодействия время отклика системы должно измеряться от момента, когда пользователь производит подающиеся наблюдению действия, до момента, когда он видит результат



**Рис. 3.6.** Исследование показало, что задержки клавиатуры варьируются в диапазоне примерно от 10 до 70 мс. В зависимости от скорости обновления экрана монитора также возможна некоторая задержка отображения символа на экране. Что происходит между этими двумя процессами, обычно определяется программным кодом

## Промышленные стандарты на время отклика системы

Время отклика системы в значительной степени определяется аппаратным обеспечением. Например, наращивание вычислительной мощности может ускорить реальную операцию поиска и время отклика системы снизится. Увеличение потребляемых программой ресурсов аппаратуры также оказывает влияние на время отклика системы. Однако и само программное обеспечение способно обеспечить повышение быстродействия. Усовершенствованные алгоритмы сжатия, рассматриваемые в качестве первого примера, даже в отсутствие форсированной аппаратной части могут ускорить передачу информации, что тоже снижает время отклика системы. Другой подобный, но не слишком заметный фактор, влияющий на время отклика системы, – это качество проектирования. Даже располагая новейшими программными технологиями, водруженными на пьедестал из мощнейшего аппаратного обеспечения, плохо спроектированная система с бесполезными излишествами и непроизводительными издержками может характеризоваться плохим временем отклика.

Кроме того, что время отклика системы определяется аппаратурой, зависит от программ и оптимизируется проектированием, на него в контексте человеко-машинного взаимодействия также влияют физиологические и психологические ограничения пользователя. Например, известно, что многие англоговорящие пользователи не могут прочесть более четырех-пяти слов в секунду – это отражение как физиологических, так и психологических ограничений. Так что не стоит представлять поток текстовой информации, которую приходится читать слишком быстро, или текстовый блок при недостаточном для чтения времени. В свете сказанного было разработано несколько промышленных стандартов на время отклика системы, которые следует применять при проектировании UI.

### Миллер (1968)

В статье 1968 года под названием «Response Time in Man-Computer Conversational Transactions» Р. Б. Миллер (R. B. Miller) описал несколько сценариев человеко-машинных взаимодействий, представив «исчерпывающее перечисление и определение различных классов действий человека и результатов на терминалах различных видов». Это одни из первых, если не первые, нормативы на время отклика системы. Язык, который Миллер использует в описаниях сценариев, например «Система, ты можешь работать со мной?», не должен восприниматься буквально, он целиком предназначался Миллером для «упрощения общения с читателями статьи». В табл. 3.1 представлено подмножество сценариев, описанных Миллером.

Таблица 3.1. Подмножество нормативов по Миллеру (1968)

Тема	Время отклика (в секундах)
Отклик на активацию средства управления	От 0,1 до 0,2
Отклик на запрос «Система, ты слушаешь?»	3
Отклик на запрос «Система, ты можешь работать со мной?»	2 (простой) 5 (сложный)
Отклик на запрос «Система, ты меня понимаешь?»	От 2 до 4
Отклик на идентификацию	От 0,4 до 0,5 (обратная связь) 0,2 (подтверждение)
Отклик на запрос «Я здесь. Что я должен теперь делать?»	От 10 до 15
Отклик на простой запрос вывода информации в виде списка	2
Отклик на простой запрос статуса	От 7 до 10
Отклик на сложный запрос табличной информации	4
Отклик на запрос следующей страницы	1
Отклик на запрос «Сейчас решай мою задачу»	15
Отклик на задержку, следующую за вводом запроса с клавиатуры или с помощью стилуса	3 (клавиатура) 2 (стилус)
Графический отклик на действие стилуса	0,1
Отклик на сложный запрос в графическом виде	От 2 до 10

## MIL-STD 1472

Подготовленный Министерством обороны США, MIL-STD 1472F, известный также под названием *Department of Defense Design Criteria Standard: Human Engineering (revision F)*, представляет собой 219-страничный документ, подробно описывающий нормативы, от тонких и грубых настроек для средств управления до цветового кодирования обычных световых индикаторов. Официально документ предназначен для «представления инженерно-психологических критериев, принципов и инструкций по проектированию для достижения успешной интеграции человека с системами, подсистемами, оборудованием и средствами и достижения эффективности, простоты, результативности, надежности и безопасности работы системы, обучения и обслуживания». Временные показатели в документе объединены для удобства читателя в таблицу XXII, и их подмножество представлено здесь, в табл. 3.2.

Таблица 3.2. Подмножество нормативов из MIL-STD 1472

Действие	Определение	Время (в секундах)
Отклик клавиши	От отжатия клавиши до положительного отклика (например, щелчка)	0,1
Печать клавиши	От отжатия клавиши до появления символа	0,2
Перелистывание страницы	От конца запроса до видимости нескольких первых строк	1,0
Просмотр страницы	От конца запроса до начала прокручивания текста	0,5
Ввод координат	От выбора поля до визуальной верификации	0,2
Функция	От выбора команды до отклика	2,0
Указание	От выбора точки до отображения точки	0,2
Вычерчивание	От выбора точки до отображения линии	0,2
Локальное изменение	Изменение изображения с использованием локальной базы данных (например, новый список пунктов меню из дисплейного буфера)	0,5
Изменение на хосте	Изменение на хосте, где данные хранятся в виде, доступном для чтения (например, изменение масштаба существующего изображения)	2,0
Изменение файла	Изменение изображения требует доступа к файлу хоста	10,0
Запрос (простой)	От команды до отображения обычного сообщения	2,0
Запрос (сложный)	Ответное сообщение требует редко используемых вычислений в графическом виде	10,0
Обратная связь относительно ошибок	От ввода до появления сообщения об ошибке	2,0

## ESD/MITRE

Эти нормативы 1986 года, опубликованные корпорацией MITRE, были подготовлены и поддержаны ВВС США. Однако авторы определен-

но ожидали, что нормативы будут полезны преподавателям, студентам, практикам в сфере инженерной психологии, исследователям, системным аналитикам и проектировщикам программного обеспечения. Формально названный ESD-TR-86-278, документ *Guidelines for Designing User Interface Software* включает 944 норматива, которые охватывают тематику ввода данных, отображения данных, управления очередностью, инструкций пользователя, передачи данных и защиты данных. Нормативы основаны на других существующих нормативах, включая раннюю редакцию MLT-STD 1472. В табл. 3.3 собраны некоторые нормативы, относящиеся к времени отклика системы.

Таблица 3.3. Подмножество нормативов из ESD/MITRE

Раздел	Действие/тема	Время (в секундах)	Норматив
1.0/4	Быстрый отклик	0,2	Максимальное время задержки при отображении сообщений обратной связи в условиях нормальной работы
1.1/5	Быстрое подтверждение ввода	0,2	Максимальное время подтверждения ввода в указанной позиции
1.1/7	Отклик курсора	0,5	Максимальное время перемещения курсора от одной позиции к другой
2.7.1/6	Быстрый отклик на запрос отображения	От 0,5 до 1,0	Отклик системы на простой запрос отображения данных
3.0/28	Приемлемое время отклика компьютера	От 0,5 до 1,0 2,0	Отклик системы на ввод с помощью средства управления
3.0/19	Доступность средства управления	0,2	Максимальное время задержек или блокировок средства управления
4.3/11	Приемлемое время отклика для сообщения об ошибке	От 2,0 до 4,0	Отображение сообщения об ошибке

## TAFIM

*Technical Architecture Framework for Information Management* (TAFIM) – это восьмитомный документ, также разработанный Минис-

терством обороны США в 1996 году для «обеспечения развития технической инфраструктуры Министерства обороны, включая определение служб, стандартов, концепций проектирования, компонентов и конфигураций, которые могут использоваться в качестве инструкции к разработке технических структур, удовлетворяющих конкретным требованиям заданий». Конкретные нормативы на человеко-машинный интерфейс содержатся в последнем томе, и нормативы, касающиеся времени, приведены в табл. 3.4.

Таблица 3.4. Подмножество нормативов из TAFIM

Раздел	Действие/ тема	Время (в секундах)	Норматив
6.6.2	Окно выполняемой работы	5	Для простых запросов, которые могут быть обработаны за это время, обеспечивается простая визуальная обратная связь в виде короткого сообщения. Если время отклика превышает указанное, приложение должно отобразить окно, информирующее о выполняемой работе
8.3.1.14	Средство управления	От 5 до 200 мс	Время отклика системы
8.3.1.15	Обратная связь	15	Если пользователь ожидает дольше этого времени, обеспечивается периодическая индикация нормального выполнения операции

## Выводы

Человеко-машинное взаимодействие является диалоговым по своей природе, и поэтому роль временных параметров как для времени отклика пользователя, так и для времени отклика системы является решающей. В отношении времени отклика пользователя важно помнить, что, если пользователь должен сконцентрировать внимание на скорости, страдает точность, и наоборот. Что касается времени отклика системы, то существует несколько промышленных стандартов, содержащих нормативы на максимально приемлемое время отклика. Несмотря на то что эти нормативы дают конкретные численные показатели, с которыми следует работать, они основаны на производительности аппаратного и программного обеспечения, доступного во время публикации документа. Поэтому пользоваться этими нормативами следует с осторожностью. В следующей главе мы рассмотрим альтернативный, ориентированный на пользователя подход к параметрам времени отклика системы.

## Кроличья нора

### Время отклика человека

Card, S. K., J. D. Mackinlay and G. G. Robertson. The Information Visualizer: An Information Workspace. ACM Conference on Human Factors in Computing Systems (CHI '91), 1991. 181–188.

Luce, R. D. Response Times: Their role in inferring elementary mental organization. New York: Oxford University Press, 1986.

Ulrich, R. and M. Giray. Time resolution of clocks: Effects on reaction time measurement – Good news for bad clocks. *British Journal of Mathematical & Statistical Psychology*, 42, 1989. 1–12.

### Время отклика системы

Beringer, J. Timing accuracy of mouse response registration on the IBM microcomputer family. *Behavior Research Methods, Instruments, & Computers*, 24, 1992. 486–490.

Miller, R. B. Response time in man-computer conversational transaction. Fall Joint Comp. Conf. U.S.A., 1968. 267–277.

Shimizu, H. Measuring keyboard response delays by comparing keyboard and joystick inputs. *Behavior Research Methods, Instruments, & Computers*, 34, 2002. 250–256.

Shneiderman, B. Response time and display rate in human performance with computers. *Computing Survey*, 16, 1984. 265–285.

### Соотношение скорость/точность

Fitts, P. M. Cognitive aspects of information processing: III. Set for speed versus accuracy. *Journal of Experimental Psychology*, 71, 1966. 849–857.

Pachella, R. G. and R. W. Pew. Speed-accuracy tradeoff in reaction time: Effect of discrete criterion time. *Journal of Experimental Psychology*, 76, 1968. 19–24.

### Законы Хика–Хаймана и Фиттса

Fitts, P. M. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47, 1954. 381–391.

Hick, W. E. On the rate of gain of information. *Quarterly Journal of Experimental Psychology*, 4, 1952. 11–26.

Hyman, R. Stimulus information as a determinant of reaction time. *Journal of Experimental Psychology*, 45, 1953. 188–196.

Seow, S. C. Information Theoretic Models of HCI: A comparison of Hick-Hyman Law and Fitts' Law. *Human-Computer Interaction*, 20, 2005. 315–352.

## Промышленные стандарты

Department of Defense Design Criteria Standard: Human Engineering. MIL-STD 1472F. (Примечание автора: доступен в Сети по адресу <http://www.hfetag.com/docs/hfs/mil-std-1472f.pdf>.)

Department of Defense Technical Architecture Framework for Information Management (TAFIM). Volume 8: DoD Human Computer Interface Style Guide.

Smith, S. L and J. N. Mosier. Guidelines for Designing User Interface Software: ESD-TR-86-278. Bedford, MA: The MITRE Corporation, 1986.

# Глава 4

## Реактивность

Реактивность обычно выступает как характеристика решения, но если решением пользуется человек, то без учета данного факта надлежащее определение этого понятия дать нельзя. В то время как аппаратное и программное обеспечение постоянно совершенствуется, человеческие возможности восприятия остаются неизменными. Эта глава знакомит вас со способом классификации реактивности в соответствии с общим восприятием того, что представляется мгновенным, незамедлительным, не прерывающим и вынуждающим.

### Что такое реактивность

Определение *реактивности* в контексте работы с компьютером иногда может оказаться столь же неуловимым, как определение красоты в поэзии и стиля в одежде, – вы понимаете это, только когда видите. Для проектирования реактивных компьютерных систем вам требуются более формальные параметры. В предыдущей главе понятие о человеко-машинном взаимодействии основывалось на представлении о взаимодействии человек-человек в силу того, что оба взаимодействия являются диалоговыми по своей природе. Если такая аналогия правомочна, значит, можно определить реактивность программного обеспечения по подобию реактивности человеческого диалога. Существует три важнейших характеристики реактивности: она определяется *относительно* типа взаимодействия в запросе; она *субъективна* в том отношении, что пользователи могут обладать различными уровнями толерантности и по-разному ее интерпретировать; наконец, она *неисключительна* по своей форме, так что любые косвенные признаки могут служить средством измерения реактивности.

## Реактивность определяется относительно взаимодействия

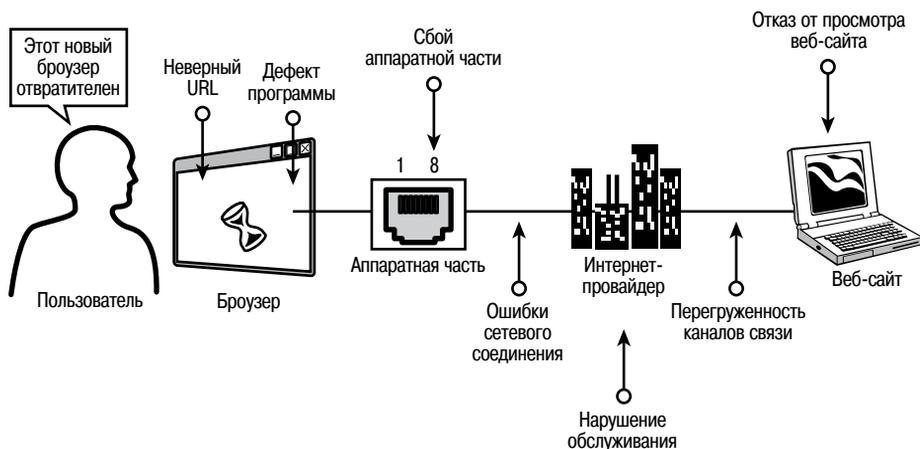
Пять секунд – слишком ли это долго для отклика? Ясно, что это зависит от того, каков был запрос. Представьте, что вы на приеме замечаете знакомое лицо, решаете подойти к человеку и спрашиваете: «Вы Джон Сперлинг, не так ли?» Скорее всего, пятисекундная задержка ответа этого человека будет явно странной. Однако, если предположить, что человек застигнут врасплох незнакомцем, задающим неожиданный вопрос (например, «Вы управляли счетом Мицубиси шесть лет назад вместе с Майком, не правда ли?»), пятисекундная задержка ответа не будет столь неожиданной, потому что для того, чтобы вспомнить, может потребоваться некоторое время.

То же самое относится к контексту человеко-машинного взаимодействия. Различные виды взаимодействий характеризуются разными приемлемыми диапазонами времени отклика. Например, время между нажатием клавиши и появлением символа следом за точкой ввода (мигающая вертикальная черта) должно быть намного меньше, чем время перехода на какой-то веб-сайт и полной загрузки веб-страницы браузером при щелчке по кнопке. Поэтому оценивание всех форм реактивности по одной шкале (от очень быстрой до очень медленной) или задание пороговых значений ( $< 20$  мс = быстро) не учитывает *относительную* природу реактивности.

## Задержка воспринимается субъективно

Вернемся к знакомому лицу на приеме. Предположим, что вы спросили человека, является ли он кем-то, кого вы знаете, а он молчит гораздо дольше пяти секунд. Что могло заставить его так долго не отвечать вам? Возможно, у него плохой слух и он даже не услышал вопроса. Может быть, он услышал ваш вопрос, но мучительно пытается вспомнить, кто вы такой. Возможно также, что он не Джон Сперлинг или вообще не понимает по-английски. Может даже оказаться, особенно с учетом направлений развития современных технологий, что Джон Сперлинг не может ответить вам, потому что в данный момент беседует по мобильному телефону, используя неприметный наушник Bluetooth.

Аналогично, в контексте человеко-машинного взаимодействия может существовать множество причин того, что система не дает отклик так быстро, как должна была бы. Рассмотрим типичный пример попытки перейти на веб-сайт (рис. 4.1). Прежде всего, может вообще отсутствовать интернет-соединение. Введенный адрес URL может быть некорректным. Веб-сайт может принимать большой поток информации именно в это время. Возможно, интернет-провайдер по каким-то причинам отключил соединение. Для типичного пользователя, особенно если он не сильно технически подготовлен, интерпретация, в общем, ограничена несколькими обычными причинами, такими как неполадки с веб-сайтом или браузером.



*Рис. 4.1. Может быть несколько причин того, что конкретная веб-страница не загружается в браузере, но для обычного пользователя вероятно лишь небольшое количество субъективных интерпретаций того, чем вызван сбой*

Вынудить пользователя ломать голову о том, почему система не отвечает, – это то же самое, что заставить человека доискиваться до того, почему в ходе беседы не прозвучал вежливый ответ. Если UI неадекватно доносит до пользователя информацию о ходе функционирования, скорее всего, у него появится желание быстренько возложить на кого-то вину. Когда дело доходит до конкретного указания на кого-то пальцем, работает принцип лезвия Оккама – простейшее объяснение всегда является лучшим: новый браузер виноват в медленном соединении, информационная система управления плохо спроектирована, потому что она настроена на отображение только внутрикорпоративной информации, и т. д. Следовательно, интерпретации задержек *субъективны*, ведь опытный пользователь может быть более благожелательным и толерантным, чем неопытный.

## Язык тела воспринимается как ответ

Если Джон действительно беседовал по мобильному телефону и использовал при этом устройство Bluetooth, то он мог бы сообщить, что «откладывает» свой ответ, посредством каких-то невербальных знаков, например поднятием вверх пальца, что означало бы «подожмите минутку». При общении обычным является сочетание словесного выражения и невербальных знаков. Иногда можно пожать плечами, для того чтобы показать, что мы не знаем ответа, или потереть лоб для выражения состояния задумчивости. В других случаях это едва различимо и менее осознанно, как то, что психолог П. Экман (P. Ekman)<sup>1</sup> называет «микровыражением», – выражения лица, которые наблюдаются около четверти секунды.

<sup>1</sup> Пол Экман – автор таких известных работ, как «Почему дети лгут?», «Психология лжи». – *Примеч. науч. ред.*

В человеко-машинном взаимодействии ясные знаки UI (линейные индикаторы выполнения и т. п.) являются не единственным способом отклика; любые формы индикации могут интерпретироваться как отклик. Рассмотрим, как компьютер включается. В разные моменты загрузки экран монитора может становиться пустым или оставаться не реагирующим в течение нескольких секунд. Однако большинство пользователей не беспокоятся по поводу такого отсутствия реакции, потому что они могут слышать шум жесткого диска или видеть мигание светодиода. Следовательно, отсутствие реакции является *неисключительным*, потому что «язык тела» системы или какой-то артефакт либо признак при выполнении системой задачи может быть интерпретирован пользователем как показатель уровня реактивности. Как и при взаимодействии человек-человек, такая индикация может и помогать, и наносить ущерб взаимодействию.

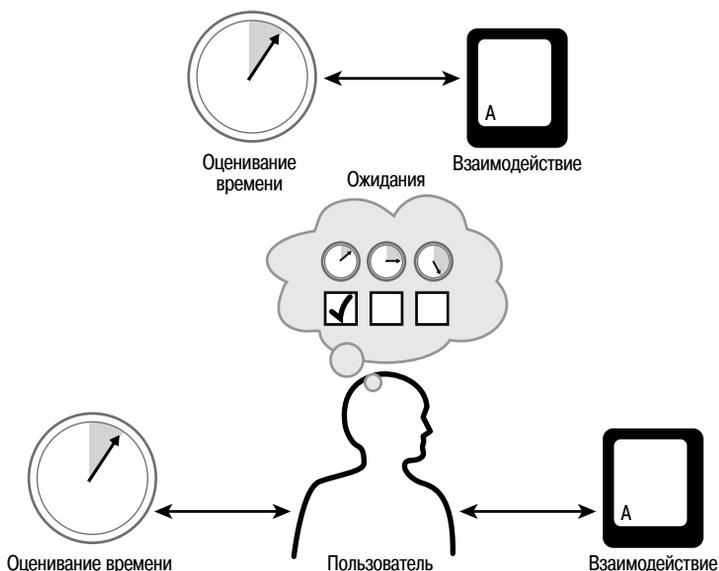
## Реактивность, основанная на ожиданиях пользователя

По мере развития компьютерных технологий наши общие способности к восприятию остаются относительно неизменными. Поэтому подход, обеспечивающий возможность распространения понятия реактивности на другие области техники и придания ему применимости по прошествии длительного времени, заключается в том, чтобы выработать показатели, *ориентированные на пользователя*, в противовес показателям, *ориентированным на технологию*. Иными словами, при определении показателей нужно основываться на наших знаниях о пользователях, а не о технологиях.

Наиболее популярные сведения о показателях времени, ориентированных на пользователя, восходят к работе С. Карда (S. Card) и его коллег по Исследовательскому центру Пало-Альто. Выдвинутое Я. Нильсеном (J. Nielsen) базовое утверждение гласит, что существует три важных *временных константы* человеко-машинного взаимодействия. Первая, называемая *константой воспринимаемого времени обработки* (*perceptual processing time constant*) и равная 0,1 с, говоря словами Нильсена, связана с «ощущением пользователей, что система реагирует мгновенно». Вторая, *константа незамедлительного времени отклика* (*immediate response time constant*), составляющая одну секунду, устанавливает «временной предел того, что неразрывность мысли остается ненарушенной». Последняя, *константа времени элементарной задачи* (*unit task time constant*), составляющая десять секунд, относится к вниманию пользователя, сосредоточенному на отдельной конкретной задаче.

Работа Карда и его коллег предлагает альтернативную возможность взглянуть на реактивность. В последующих разделах мы рассмотрим соответствующую ориентированную на пользователя систему воззрений,

базирующуюся на ожиданиях пользователя. Она основана на том, что вместо связывания конкретного взаимодействия с фиксированным значением (например, *время отклика: 0,1 с*) для каждой ожидаемой временной величины используется диапазон значений (основанный на сведениях, уже известных и установленных эмпирически) (*мгновенно: от 0,1 до 0,2 с*), для того чтобы задать диапазон наиболее приемлемых значений времени отклика. Затем конкретные взаимодействия связываются с ожиданиями пользователей (например, *отклик на нажатие клавиши: мгновенный*). При этом система воззрений оказывается согласованной с новыми формами взаимодействий по мере их появления. Рис. 4.2 иллюстрирует этот подход.

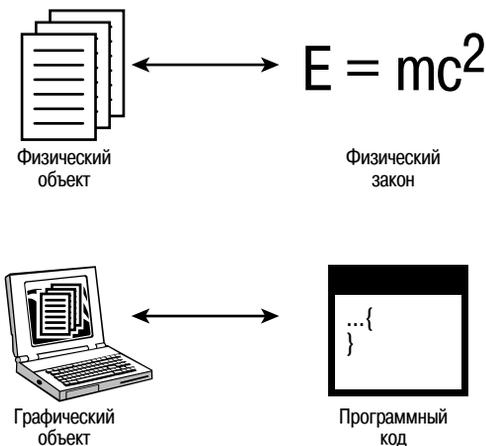


*Рис. 4.2. Модель сверху связывает взаимодействие непосредственно с временным показателем. Модель внизу связывает взаимодействие с ожиданием пользователя (мгновенно, незамедлительно и т. д.)*

## Мгновенная: от 0,1 до 0,2 секунды

Когда вы щелкаете выключателем от лампы, вы ожидаете включения или выключения света. Мгновенная (Instantaneous) реакция и другие характеристики выключателя не только обеспечиваются материалом и механическими деталями, из которых он состоит, но и базируются на физических законах. С другой стороны, графическая кнопка в программном приложении не подчиняется тем же самым физическим законам (например, касающимся силы, тяготения и т. д.), которые управляют поведением выключателя. Например, с помощью нескольких строк программного кода можно отжать графическую кнопку через три секунды после щелчка пользователя даже без его вмешательства.

Такое поведение не является характерным для реальных физических кнопок или выключателей, потому что «законы», управляющие поведением графической кнопки, а также то, как она выглядит и ощущается, являются искусственными и произвольно определенными в программном коде. Рис. 4.3 иллюстрирует эту аналогию.



*Рис. 4.3. Физические объекты подчиняются физическим законам, в то время как объекты, генерируемые компьютером, подчиняются законам, заданным программным кодом*

Очевидно, что, помимо востребованности и удобства использования программ, важно то, насколько удачно те или иные их свойства имитируют объекты и модели поведения, наблюдаемые в физическом мире. Например, для того чтобы кнопки походили на реальные, принято делать их откликающимися звуком на щелчки мышью и придавать им визуальные эффекты, к примеру затенение. Другим важным аспектом «похожести» является реактивность кнопки. Во многих случаях все графические средства управления или весь пользовательский ввод должны функционировать как откликающиеся мгновенно. Вообще, все виды взаимодействия, копирующие какие-то объекты физического мира, которые действуют мгновенно, должны демонстрировать мгновенную реактивность. Их перечень включает множество видов пользовательского ввода и их основные действия, но ими не ограничивается. Для таких взаимодействий максимальное время отклика – от наблюдаемого действия, инициированного пользователем, до наблюдаемого отклика компьютера – для обычного ввода (например, нажатия клавиши) должно быть меньше 0,1 с, а для более сложных действий, например вызова раскрывающегося меню, – меньше 0,2 с.

## Незамедлительная: от 0,5 до одной секунды

В отличие от только что описанной мгновенной реактивности, этот следующий класс реактивности предполагает незамедлительное (Immediate) подтверждение получения информации или команд и рефлексивное

реагирование на эту информацию. Это то же самое, что начало диалога человека с человеком, когда одна сторона инициирует беседу привлечением внимания другой стороны посредством обращения к ней или к нему по имени:

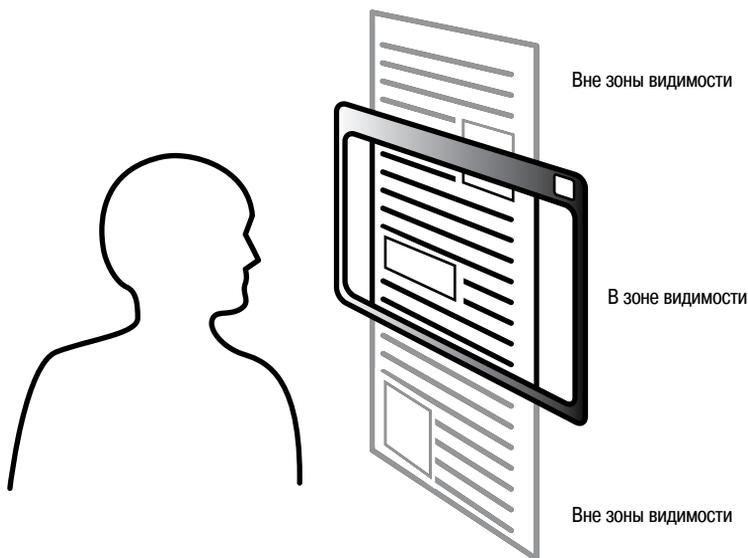
**Джон:** Мэри!

(менее чем через одну секунду)

**Мэри:** Да, Джон?

Если вы замените «одну секунду» на «три секунды», будет казаться, что диалог какой-то неправильный. Как и в том, что мы обсуждали раньше, появятся основания интерпретировать задержанный ответ как неуверенность, невнимательность или даже виновность. Аналогично, при замене величины задержки на «0,2 с» вы тоже начнете искать в диалоге что-то необычное. В этом последнем случае можно интерпретировать быстрый ответ как тревогу, гиперактивность или угодничество. Многие люди подсознательно испытывают неловкость при затянувшихся паузах в беседе и непроизвольно стараются их заполнить. Кард и его коллеги пишут следующее:

*При паузе более секунды либо слушающая сторона дает отклик по каналу обратной связи для демонстрации своего состояния прослушивания (например, «так-так»), либо говорящая сторона дает отклик (например, «так...»), чтобы показать, что она сейчас обдумывает новое высказывание.*



**Рис. 4.4.** Пользователь видит только то, что показывает ему приложение, а другие части документа находятся вне зоны видимости. Для их отображения потребуются лишь прокрутить страницу вверх или вниз, что должно выполняться незамедлительно

В контексте человеко-машинного взаимодействия класс незамедлительной реактивности устанавливает стандарт для выдачи пользователю подтверждения того, что его информация или команды получены, а если запрос, указание или команда просты – то и для предоставления полного ответа. Например, если пользователь запрашивает информацию, которая воспринимается как уже готовая (к примеру, некоторый текст на веб-странице, не уместившийся в окне браузера), то он ожидает, что запрашиваемое им уже здесь, рядом, и его отображение требует лишь прокрутки страницы (рис. 4.4). В таких случаях максимальный предел времени отклика – опять-таки, от наблюдаемого действия, инициированного пользователем, до наблюдаемого отклика компьютера – составляет от 0,5 до одной секунды.

## Не прерывающая: от двух до пяти секунд

Неразрывность – весьма желательное качество диалога. В обычной беседе человека с человеком распределение времени обмена информацией между обеими сторонами никогда не является постоянным (если только стороны не декламируют строки сценария), длительность пауз в речи изменяется. Эти переменные паузы являются нормальными и отражают время, затраченное каждой стороной на обдумывание и подготовку ответа. При слишком быстром ответе есть основания предположить, что он был подготовлен заранее или не вполне адекватно обдуман. Если ответу, напротив, предшествует длинная неловкая пауза, течение беседы тоже может пострадать. Р. Миллер пишет следующее:

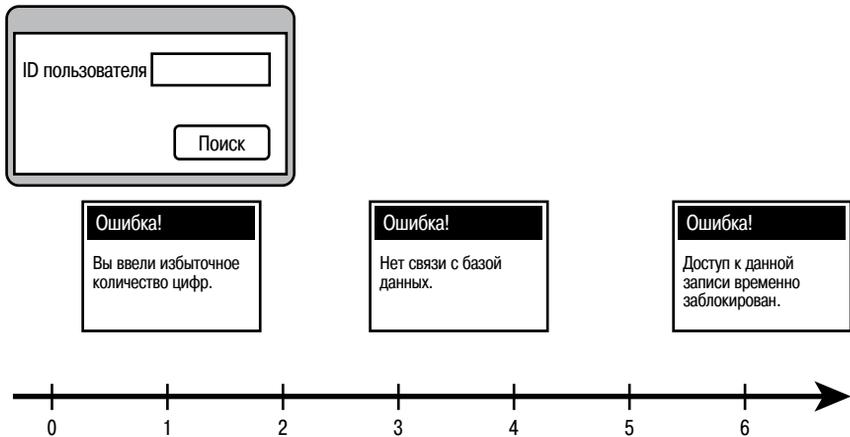
*Если вы обращаетесь к другому человеку, то ожидаете получить какой-то коммуникативный отклик в пределах x секунд – возможно, от двух до четырех секунд... В диалоге любого рода между людьми молчание, длящееся более четырех секунд, становится неловким, потому что оно вызывает прерывание потока общения.*

До известной степени пользователи понимают, что компьютеру требуется время для «обдумывания». Однако для обеспечения *непрерывности* система должна выдать *какую-то* разновидность ответа в пределах допустимого диапазона времени. Миллер дает ясную рекомендацию: «Пользователь всегда сохраняет непрерывность мысли, когда он делает сложные запросы. Этот конкретный запрос требует завершающего отклика в пределах четырех секунд».

Следует помнить о двух важных вещах:

- Во-первых, это требование сохранения непрерывности при человеко-машинном взаимодействии обычно не относится к пользователю. Это значит, что пользователь не обязан отвечать компьютеру в пределах интервала от двух до пяти секунд.
- Во-вторых, поддержание непрерывности может включать случаи, когда система сообщает, что она не может начать или продолжить обработку запроса пользователя. Иногда это случается по причине

того, что пользователь ввел ошибочные данные, например содержащие опечатки или синтаксические ошибки. В тех случаях, когда ошибки *воспринимаются* как простые, система должна ответить сообщением об ошибке в пределах двух секунд. Если же система принимает пользовательский запрос, но не может выполнить его по той или иной причине, она должна ответить сообщением об ошибке менее чем через пять секунд. Аналогично, для команд или действий, *воспринимаемых* пользователем как простые, система должна ответить по прошествии примерно двух секунд. На те же, которые *воспринимаются* как сложные, система должна ответить менее чем через пять секунд. На рис. 4.5 показаны гипотетические ошибки различных уровней сложности и приемлемое время отклика для них.



**Рис. 4.5.** Гипотетический пример распределения времени, связанного с уровнем сложности ошибки. Простые ошибки, например неверно введенная строка, должны быть быстро «выявлены» и инициирована выдача сообщения. Более сложные ошибки, например показанная справа, потребуют достаточно большого времени обработки

Нижняя и верхняя границы класса не прерывающей (Continuous) реактивности также служат удачными ориентирами того, какой уровень обратной связи следует обеспечить во время ожидания пользователем завершения обработки его действия. В течение двух секунд никакая обратная связь не нужна, потому что двухсекундная или меньшая задержка обычно воспринимается терпеливо (кроме тех случаев, когда пользователи ожидают мгновенной или незамедлительной реакции). Однако при превышении двух секунд может быть рассмотрено обеспечение простой обратной связи, свидетельствующей о функционировании. При превышении пяти секунд такая обратная связь абсолютно необходима. Это означает, что периодически с интервалом от двух до пяти секунд должен демонстрироваться факт функционирования, для

того чтобы пользователь оставался уверенным, что система не зависла. Дополнительную информацию по этой теме вы найдете в этой книге в главе 6 «Индикация выполнения».

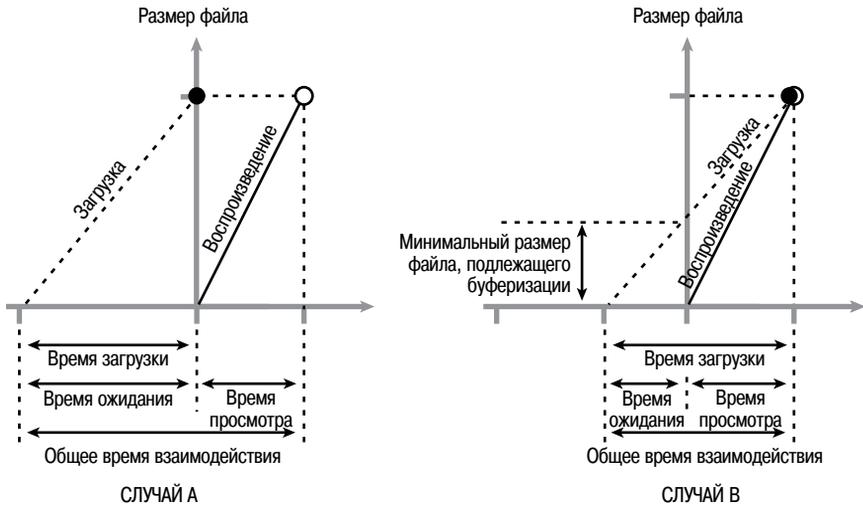
## Вынуждающая: от семи до десяти секунд

Исследования показали, что люди склонны отказаться от обращения к неотвечающим веб-сайтам после ожидания примерно от восьми до десяти секунд. Этот диапазон содержит то девятисекундное значение, которое названо Т. Селкером (T. Selker), сотрудником Массачусетского технологического института, «объемом внимания золотых рыбок», и то десятисекундное значение, которое названо Кардом и его коллегами «константой времени элементарной задачи». Эти и другие исследования показывают, что диапазон времени внимания пользователя, по грубой оценке, составляет от семи до десяти секунд. В течение этого времени должны быть выполнены действия, необходимые для удержания внимания пользователя. При превышении указанного времени пользователь склонен отвлечься или переключить внимание на другие задачи. Очевидно, что некоторые процессы продолжаются более десяти секунд. Даже при наличии периодической обратной связи в ожидании отклика системы дольше десяти секунд терпение пользователя истощается. Для случая же, когда общее время отклика превышает 15 с, Миллер советуется в своей работе 1968 года:

*Если он занятый человек, вынужденное ожидание в течение более 15 с, даже если информация для него существенна, может вызвать нечто большее, чем раздражение и гнев. Он может потерять самоконтроль... Поэтому, если случаются задержки отклика, превышающие 15 с, лучше спроектировать систему так, чтобы освободить пользователя от физического и умственного плена, чтобы он мог заняться чем-то другим и получил отображение ответа тогда, когда ему это удобно.*

Практический способ состоит в том, чтобы предоставлять пользователю полезную информацию о прохождении процесса примерно через каждые десять секунд. Этот прием используется во многих приложениях потокового видео, где приложение вначале сохраняет определенный объем информации в буфере и затем начинает воспроизведение буферизированного видео, а в это время загружаются оставшиеся части (рис. 4.6). Если нет возможности предоставить пользователю существенную информацию в течение периода менее десяти секунд, воспользуйтесь советом Миллера и предоставьте пользователю возможность переключить внимание на другие задачи.

Важно помнить, что приведенные выше значения представляют нижний и верхний пределы *максимально приемлемого* времени отклика. Например, простое действие, такое как нажатие клавиши, должно завершиться не позже нижнего порога класса мгновенной реактивности, равного 0,1 с. Существует, однако, опасность сделать что-либо слишком быстро, и это обсуждается в следующем разделе. На рис. 4.7 обобщенно показаны четыре уровня реактивности.



**Рис. 4.6.** Два опыта просмотра видео: В случае А пользователь загружает видео целиком до начала воспроизведения. Общее время, затрачиваемое при таком взаимодействии, равно сумме времени загрузки и времени воспроизведения. В случае В воспроизведение начинается сразу, как только становится доступным определенный объем информации в буфере. Несмотря на то что время загрузки остается прежним, общее время, затраченное на такое взаимодействие, меньше, поскольку время загрузки перекрывается с временем получения информации



**Рис. 4.7.** Максимальное время отклика, выраженное в виде диапазонов, для четырех классов реактивности

## Слишком быстро?

Могут ли отклики быть слишком быстрыми? Интуитивно кажется, что во всем, чем занимается компьютер, — чем *быстрее*, тем *лучше*. Ведь такая тенденция очевидна со времени наступления эпохи современных

компьютеров. Как сказано ранее, компьютерная техника продолжает развиваться и совершенствоваться, однако познавательные способности и возможности восприятия обычного человека относительно постоянны. Такое «несоответствие» порождает проблемы удобства использования, когда компьютер слишком скор для пользователя или когда пользователь слишком скор для компьютера.

## Компьютер слишком скор

Интуитивно не очевидно, но, тем не менее, возможно, что решение или система будут *слишком скоры* для пользователей. Это необязательно предполагает медленного пользователя, работающего на превосходном компьютере, потому что понятие «слишком быстро» относительно. Компьютер может демонстрировать приемлемую скорость реакции, но пользователь по различным причинам испытывает трудности в том, чтобы поспеть за ним. Это часто бывает с пользователями, медленно работающими с устройствами ввода, а также с теми, кому необходимо больше времени для планирования своего отклика из-за слабого знания UI, или для принятия решений, или для других когнитивных процессов.

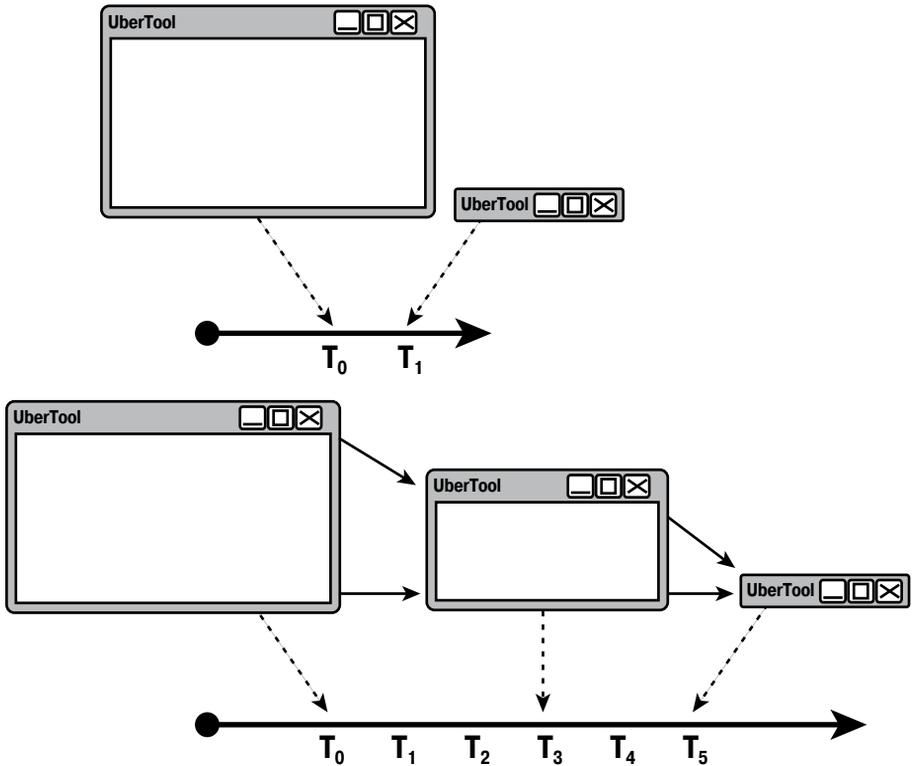
Яркий пример: двойные щелчки в среде операционной системы компании Майкрософт. Представим, что пользователь делает двойной щелчок по ярлыку, расположенному на рабочем столе. В зависимости от скорости щелчков (время между отпусканием кнопки мыши и повторным ее нажатием в первом и втором щелчках) компьютер ведет себя так или иначе. В среде Windows по умолчанию двойной щелчок должен выполняться не дольше 500 мс, если целью является запуск приложения или открытие документа. Однако если два щелчка выполнены с более чем 500-миллисекундным интервалом между ними, то активируется режим редактирования имени ярлыка, предназначенный для его переименования. При этом вместо открытия приложения пользователь вынужден будет заняться переименованием ярлыка.

Исследования показывают, что на то, насколько быстро люди могут совершать простые повторяющиеся удары пальцем, существенно влияет возраст. Средняя максимально возможная скорость таких движений для большинства взрослых составляет примерно 200 мс. Неудивительно, что скорость ударов пальцем с возрастом падает. Поэтому, если для двойного щелчка установлена слишком высокая скорость, определенная категория пользователей не может выполнить простую задачу запуска приложения посредством двойного щелчка. К счастью, Windows предоставляет возможность изменения параметра скорости двойного щелчка.

Простое увеличение времени отклика компьютера для подгонки ко времени отклика человека не является эффективной мерой, потому что, в общем, для системы предпочтительна, в плане производительности, быстрая реакция в ответ на действия пользователя. Лучшее решение

заключается в такой минимальной настройке реактивности компьютера, чтобы пользователи могли испытывать удовлетворение от своего прогресса и система отзывалась бы на этот прогресс.

Преднамеренное замедление системы или приложения не является распространенной практикой, но в некоторых случаях оно необходимо. Прекрасным примером незначительного увеличения отклика системы для обеспечения максимального удобства пользователя является анимация при перемещении окон. Вы можете сворачивать и раскрывать окна немного быстрее, отключив анимационные эффекты, наглядно иллюстрирующие изменения окон, чтобы окна немедленно принимали требуемое состояние. Однако при этом пользователи, особенно начинающие, возможно, не вполне будут понимать, что происходит с окнами. Р. Рао (R. Rao) выразил эту мысль так: «Анимирование переходов между одним расположением информации и другим позволяет пользователю связать до и после. При хорошо спроектированном



*Рис. 4.8. Вверху: мгновенное сворачивание окна без визуальных эффектов приводит к тому, что пользователю, особенно новичку, окно кажется исчезнувшим. Внизу: если затрчено чуть больше времени для того, чтобы анимировать сжатие окна до минимизированного состояния, пользователи лучше понимают, что произошло с окном и куда оно делось*

переходе пользователь не будет озадачиваться тем, что происходит». Несмотря на то что такие мелкие эффекты визуализации в UI не добавляют никакой функциональности, низкая стоимость реализации повышает применимость решения. На рис. 4.8 показана минимизация окна при наличии анимированных переходов и без них.

Для справки отметим, что для создания иллюзии движения можно использовать так называемый *фи-феномен* (*phi phenomenon*) (известный также как иллюзорное движение или кажущееся движение). Впервые описанный психологом М. Вертгеймером (M. Wertheimer)<sup>1</sup>, фи-феномен заключается в визуальной иллюзии движения одного объекта, в то время как два стационарных экземпляра объекта в различных позициях быстро сменяют друг друга. Исследования показали, что 40 мс (с момента исчезновения первого экземпляра до момента появления второго) – это оптимальная длительность для обеспечения эффекта иллюзии при малых расстояниях.

## Пользователь слишком скор

Другой сценарий, где несоответствие между временем отклика человека и системы вызывает проблемы удобства использования, заключается в том, что пользователь выполняет действия слишком быстро для компьютера. Б. Шнайдерман пишет: «Имеется также опасность слишком быстрой работы. Когда пользователи повышают скорость своих действий в последовательности взаимодействий, они могут начать хуже обучаться, читать со сниженным пониманием, принимать плохо обдуманное решение и делать больше ошибок при вводе». Прочтения, указанные Шнайдерманом, могут быть объяснены теорией соотношения скорость/точность, описанной в главе 3 «Время отклика пользователя и системы».

Нанесение «ударов по скорости», для того чтобы замедлить работу пользователей, имеет негативные последствия для удобства использования, потому что для пользователей естественно повышать скорость работы по мере накопления опыта выполнения конкретной задачи. Как было отмечено ранее, производительность выполнения нового задания со временем и с опытом повышается (закон научения). В целом, это полезное и естественное явление, поскольку очевидно, что пользователи обучаются и совершенствуются. Однако, когда пользователи достигают своей «конечной скорости», риск сделать ошибку возрастает. Примером может служить то, как пользователи Windows, выбрав несколько файлов, нажимают клавишу Delete и мгновение спустя – клавишу Enter, чем подтверждают намерение удаления. Из-за того что операционная система «недостаточно быстрая» для пользователя, подтверждение намерения удаления она воспринимает как требование открытия всех выбранных файлов, что и выполняет.

<sup>1</sup> Макс Вертгеймер – немецкий психолог, основатель гештальтпсихологии. В 1912 г. провел экспериментальное исследование феномена кажущегося движения, положившее начало гештальтпсихологии. – *Примеч. науч. ред.*

Одни ошибки незначительны, другие могут быть существенными. В любом случае ошибка может оказывать негативное влияние на производительность труда пользователей и удобство использования, если работу приходится постоянно исправлять или переделывать. Для того чтобы справиться с этой неизбежной уязвимостью, можно применять общеизвестный подход, воплощенный в стратегии «стоп-сигналов». Он характеризуется тем, что пользователей, по существу, просят проверять свои действия, особенно критические, например удаление файла, форматирование диска или передачу важной информации. Этот подход следует реализовывать в минимальной степени. Другой общепринятый практический способ заключается в объединении возможностей, которые помогают пользователю возвратиться назад по времени, для того чтобы избавиться от результатов неверного образа действий. Прекрасными примерами могут служить временное удаление и возможность отмены действий.

## Выводы

При моделировании человеко-машинного взаимодействия, основываясь на взаимодействии человек-человек, мы определяем реактивность для первого так же, как мы это делаем для второго. Максимально приемлемое время, затраченное на подготовку отклика, зависит от природы отклика, следовательно, реактивность определяется *относительно* типа взаимодействия. Несмотря на то что задержки, которые превышают обычные, вызываются многими причинами, вероятно, возникнет лишь несколько *субъективных* интерпретаций того, что происходит. Наконец, реактивность *неисключительна* и может характеризоваться признаками любой формы. С учетом этих факторов была представлена теория реактивности, ориентированная на пользователя и основанная на общем представлении о том, что является мгновенным, незамедлительным, не прерывающим и вынуждающим.

## Кроличья нора Микровыражения

Ekman, P. *Emotions Revealed: Recognizing faces and feelings to improve communication and emotional life*. New York: Times Books, 2003.

## Реактивность с точки зрения пользователя

Card, S. K., J. D. Mackinlay and G. G. Robertson. *The Information Visualizer: An Information Workspace*. ACM Conference on Human Factors in Computing Systems (CHI '91), 1991. 181–188.

Dix, A. J. *The myth of the infinitely fast machine*. People and Computers III – Proceedings of HCI'87. D. Diaper and R. Winder (Eds.). Cambridge University Press, 1987. 215–228.

Geist, R., R. Allen and R. Nowaczyk. Towards a model of user perception of computer system response time. CHI + GI 1987, 1987. 249–253.

Miller, R. B. Response time in man-computer conversational transaction. Fall Joint Computer Conference U.S.A., 1968. 267–277.

Shneiderman, B. Response time and display rate in human performance with computers. *Computing Survey*, 16, 1984. 265–285.

## Потоковое состояние пользователя

Csikszentmihalyi, M. Flow: The Psychology of Optimal Experience. New York: Harper and Row, 1990.

Ghanij, A. and S. P. Deshpandes. Task characteristics and the experience of optimal flow in human-computer interaction. *Journal of Psychology*, 128, 1994. 381–391.

Rao, R. See and Go Manifesto. *Interactions*, September–October 1999.

## Объем внимания

Christakis, D. A., F. J. Zimmerman, D. L. DiGiuseppe and C. A. McCarty. Early television exposure and subsequent attentional problems in children. *Pediatrics*, 113, 2004. 708–713.

Turning into digital goldfish. <http://news.bbc.co.uk/2/hi/science/nature/1834682.stm>.

Web “turns people into goldfish”. *The Inquirer*, February 22, 2002.

Zimmerman, F. J. and D. A. Christakis. Associations between content types of early media exposure and subsequent attentional problems. *Pediatrics*, 120, 2007. 986–992.

# Глава 5

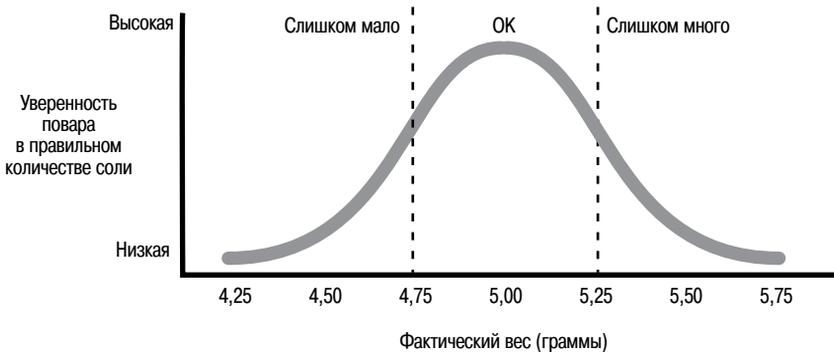
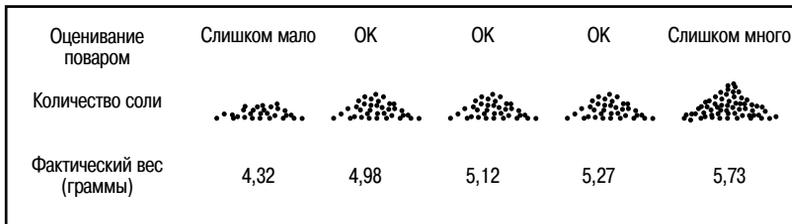
## Обнаружение различий в распределении времени

Осмысление возможности обнаружения различий в распределении времени является решающим условием совершенствования многих аспектов человеко-машинного взаимодействия. К примеру, ниже определенного порога различия становятся слишком малыми, для того чтобы человек мог их обнаружить. Затраты времени, усилий и ресурсов для создания различия, которое не может быть замечено, окажутся бесполезными. Эта глава знакомит вас с основными концепциями определения человеком изменений в распределении времени, для того чтобы помочь в установлении требуемых эксплуатационных характеристик, например в задании целевой производительности и определении допустимой регрессии.

### Объявление различия

Вы пишете спецификацию для следующей версии вашей программы записи на компакт-диск. Конкретная функция текущей версии программы для своего завершения требует 20 с. Очень способные разработчики в вашем коллективе могут снизить это время с 20 до 17 с, но для написания кода им нужны две недели. Полагая, что три секунды представляют существенную величину, вы решаете, что эти улучшения стоят времени и усилий, и официально включаете их в спецификацию. Какие риски вы не приняли во внимание? Читайте дальше.

Врожденная способность обнаруживать малейшие различия столь же полезна при использовании компьютеров, как и при решении повседневных задач, потому что множество измерений в жизни производится не посредством объективного физического инструментария, а при помощи субъективной мысленной аппроксимации. К примеру, опытному повару не требуются измерительные устройства при смешивании продуктов для приготовления блюда. Скорее всего, он на глаз может оценить, сколько надо взять каждого ингредиента, ведь приготовление блюда – в большой степени практическая задача. Не пользуясь измерительной ложкой или стаканом, повар знает, когда взято слишком мало или слишком много соли. Всякий раз повар использует различное количество соли, и приемлемое количество на самом деле лучше представляется в виде диапазона (рис. 5.1). Если количество ниже нижней или выше верхней границы этого диапазона, повар определит, что взято слишком мало или слишком много соли.



*Рис. 5.1. Необходимое количество соли лучше задавать не точно, а в некотором диапазоне*

В контексте человеко-машинного взаимодействия понимание способности пользователей определять различия в распределении времени важно во многих отношениях. При установлении требуемых эксплуатационных характеристик, например, в вымышленном сценарии, описанном в начале этой главы, вы должны рассмотреть, сможет ли пользователь заметить конкретное улучшение характеристики. Если усовершенствование даже не может быть обнаружено, затраты становятся бессмысленными. С другой стороны, зная диапазон, в котором

изменения распределения времени незаметны, можно определить, каково приемлемое снижение характеристики, – это часто называют *допустимой регрессией*.

## D-уровни

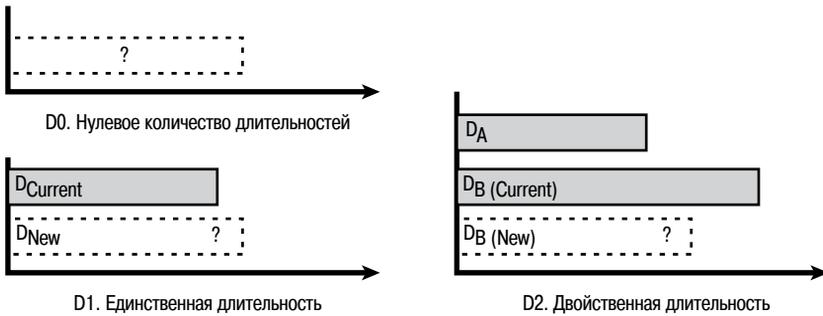
В главе 4 «Реактивность» описывается деление различных видов взаимодействия на классы в зависимости от ожиданий пользователей. Например, большинство объектов UI, таких как кнопки и другие средства управления, в которых воплощено подобие физическим объектам реального мира, должны при проектировании наделяться свойством немедленного отклика. С другой точки зрения, процесс определения подходящего распределения времени для конкретного взаимодействия может рассматриваться как сценарий с нулевым количеством длительностей *D0* (zero-duration), в рамках которого вы просто определяете подходящее распределение времени вне зависимости от других измерений времени, эталонов и иных мер.

В некоторых случаях вы не работаете со сценариями типа *D0*, потому что необходимо модифицировать существующую длительность. Например, надо определить, оправдает ли предполагаемое улучшение распределения времени цену своей реализации. В ситуации, описанной в начале этой главы, результат планируемой модификации проекта заключается в трехсекундном ускорении процесса, который обычно занимает 20 с. При таком сценарии вам следует определить рентабельность инвестиций в снижение 20-секундной длительности до 17-секундной. По причине того, что это связано с модификацией единственной длительности, подобные сценарии попадают в разряд сценариев с единственной длительностью *D1* (single-duration).

Если имеют место две сосуществующие длительности, в особенности в конфликтных ситуациях, то одна длительность обязательно сравнивается с другой. Например, функция поиска в приложении возвращает результаты за три секунды, при этом ваша текущая эквивалентная функция то же самое делает за шесть секунд. Хорошо бы по возможности сделать время отклика меньше трех секунд, но иногда минимально приемлемым является повышение планки (в данном примере уменьшение времени) до уровня, позволяющего, по выражению Дж. Мура (G. Moore)<sup>1</sup>, *нейтрализовать* конкуренцию так, чтобы исключить *дифференциацию*. Мысль Мура мы поясним позже, а сейчас ситуации, при которых две длительности противостоят друг другу, отнесем к категории сценариев с двойственной длительностью *D2* (dual-durations), потому что цель должна достигаться путем рассмотрения двух установленных длительностей. Рис. 5.2 иллюстрирует три рассмотренных D-уровня.

---

<sup>1</sup> Джеймс Ф. Мур – американский психолог, эксперт по проблемам управления. – *Примеч. науч. ред.*



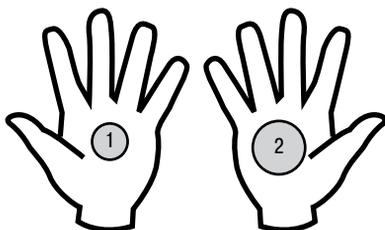
*Рис. 5.2. Иллюстрация трех D-уровней. Вверху слева длительность не зависит от других длительностей в сценарии D0. Внизу слева текущая длительность ( $D_{Current}$ ) подлежит модификации ( $D_{New}$ ) в сценарии D1. В последнем сценарии новая длительность ( $D_{B(New)}$ ) определяется для текущей длительности ( $D_{B(Current)}$ ) с учетом другой длительности ( $D_A$ )*

Рекомендации для сценариев типа D0 были разработаны в главе 4. Для других двух сценариев зададимся целью осмыслить обнаружение и восприятие различий распределения времени и то, как это можно применить для оптимизации человеко-машинных взаимодействий. Два психологических принципа – закон Вебера и деление надвое посредством среднего геометрического – имеют прямое отношение к сценариям типов D1 и D2. Первый принцип относится к обнаружению различия между величинами, характеризующими некоторое свойство. В компьютерном контексте он относится к сравнению целевой и относительно фиксированной длительностей, таких как время отклика приложения до и после реализации некоторых улучшений. Второй принцип также относится к паре величин, но концентрирует внимание на определении зоны отсечения с учетом двух существующих длительностей.

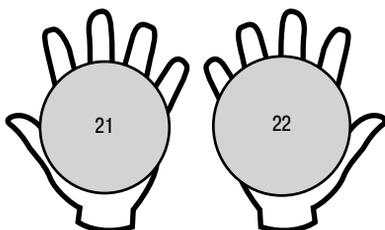
## D1: закон Вебера

Представьте себе, что вы держите в руках два предмета: весом 1 фунт в одной руке и весом 2 фунта в другой. По всей вероятности, вы сможете определить, какой из них тяжелее, просто потому, что их держите. Представим себе повторение того же эксперимента с другой парой предметов. На этот раз первый весит 21 фунт, а другой – на фунт больше, 22 фунта (рис. 5.3). Несмотря на то что в обоих парах предметы различаются по весу на один фунт, вероятно, во втором случае вы не сможете столь же уверенно сказать, какой предмет тяжелее. Не огорчайтесь, мышцы обеих ваших рук работают отлично.

Это явление впервые наблюдалось Э. Г. Вебером (E. H. Weber), немецким физиком и пионером экспериментальной психологии. Работая с Г. Т. Фехнером (G. T. Fechner), Вебер сформулировал закон, который сначала назывался его именем, а позднее – законом Вебера–Фехнера.



Сравнение А. Простое обнаружения различия в один фунт



Сравнение В. Сложное обнаружения различия в один фунт

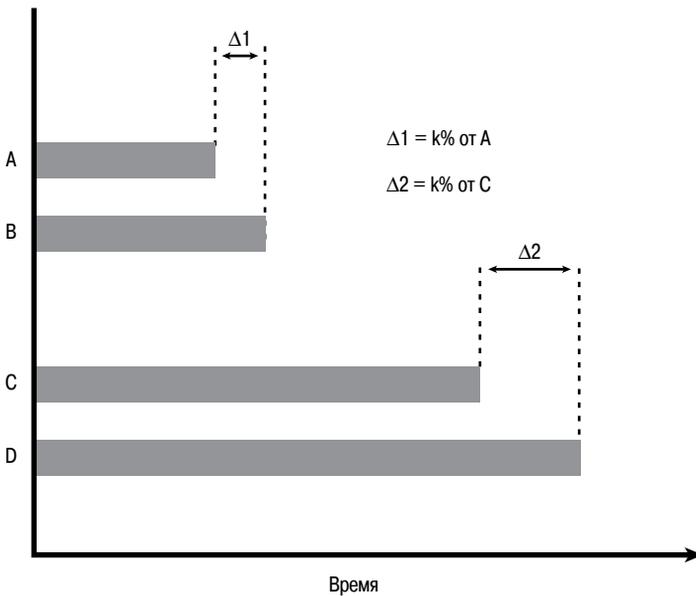
**Рис. 5.3.** Совсем не сложно обнаружить различие между 1 и 2 фунтами в сравнении А (вверху). Однако, несмотря на то же различие в 1 фунт в сравнении В, обнаружить различие между 21 и 22 фунтами труднее

Закон Вебера базируется на ключевой концепции о *наименьшем заметном различии*, обычно обозначаемом аббревиатурой *jnd* (*Just Noticeable Difference*). Как следует из названия, *jnd* – это минимальное увеличение или уменьшение величины свойства раздражителя (яркость гирлянды лампочек, громкость постоянного жужжания и т. д.), которое может быть обнаружено, или, как гласит название, замечено. В примере со сравнением весов вы можете определить *jnd* посредством увеличения веса второго объекта до тех пор, пока различие не станет заметным<sup>1</sup>. Разумеется, в паре приведенных примеров с 1 и 2 фунтами и 21 и 22 фунтами *jnd* различны. Это означает, что во втором опыте вам, вероятно, потребуется разница более чем в 1 фунт, для того чтобы стало возможным определить, какой предмет тяжелее. Вебер открыл закономерность роста *jnd* при возрастании сравниваемых величин<sup>2</sup>.

В контексте восприятия времени закон Вебера применим во всех случаях сравнения двух длительностей. Исследователи распределения времени обычно нормируют стандартное отклонение для получения так называемой *дроби Вебера*, или *отношения Вебера* (рис. 5.4).

<sup>1</sup> Формально говоря, минимальная разница, которая может быть замечена в 50% случаев, и будет составлять *jnd*. Например, если проведено десять опытов и в шести из них участники смогли обнаружить различие, когда добавляется третий фунт, то *jnd* равно 3 фунтам.

<sup>2</sup> Согласно теории, *jnd* возрастает как линейная функция логарифма раздражителя.

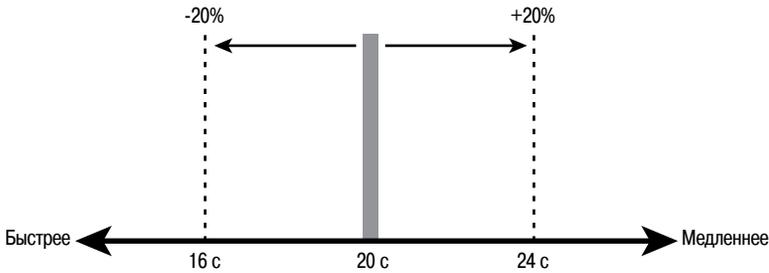


*Рис. 5.4. Пусть установлено, что длительность В имеет отличие от длительности А, минимально возможное для того, чтобы его можно было распознать, и то же самое справедливо для длительностей С и D. Закон Вебера говорит о том, что отношение разности (1 и 2) к меньшей длительности в обеих парах одно и то же*

Можно говорить, пользуясь языком дилетанта, что возможность распознать наименьшее отличие между двумя произвольными длительностями согласно теории описывается постоянным отношением (например, между 10 и 12, между 20 и 24 или между 3000 и 3600). Это отношение может быть рассчитано посредством использования среднего значения и стандартного отклонения, которые характеризуют набор чисел из эксперимента, где участникам необходимо обнаружить отличие между несколькими парами длительностей. Исследования показывают, что отношение Вебера для распознавания временных отличий (различения двух длительностей) находится в диапазоне от 7 до 18% для длительностей до 30 с.

## Правило 20%

Есть примеры непосредственного применения закона Вебера и концепции  $\ln k$  к человеко-машинному взаимодействию. На основании данных исследования оценивания времени человеком получено практическое правило о применении отношения Вебера, равного 20%, к интересующей вас длительности. Вы можете иметь в виду значения, либо не достигающие, либо превышающие границы зоны отсека, в зависимости от контекста (рис. 5.5). В следующих разделах обсуждаются два основных применения *правила 20%*; первое относится к значению, на 20% ниже нижней, а второе – к значению выше верхней границы зоны отсека.



*Рис. 5.5. Правило 20% дает нижнюю и верхнюю границы интересующей вас длительности (на рисунке 20 с) при рассмотрении улучшения характеристик и допустимой регрессии*

## Задание требуемой целевой производительности

Существует множество причин, по которым какие-то из реально существующих длительностей требуется сократить. Потребители могут жаловаться на вялое выполнение конкретной функции поиска в вашем продукте. Возможно, тестирования удобства использования подтвердили, что пользователи покидают веб-сайт из-за слишком медленной загрузки первой страницы. В подобных сценариях типа D1 вы имеете дело с существующей длительностью, которую следует сократить для повышения удобства использования и удовлетворенности. В таких случаях следует установить планку улучшения распределения времени по крайней мере на уровне 20% – тогда улучшение будет заметно. Например, если функция поиска доставляет результаты в течение десяти секунд, то будет разумным добиваться нового значения времени отклика меньше восьми секунд.

Отметим, что это общее практическое правило повышения вероятности именно обнаружения. Отношение Вебера как раз дает сведения о величине разности, достаточной для того, чтобы различие между двумя длительностями было *замечено*. Однако тот факт, что различие *может быть замечено*, не является показателем того, что оно будет *понято и оценено*. Предположим, что новый метод сжатия ускоряет загрузку с 30 до 24 с, обеспечивая уменьшение затрачиваемого времени на 25%, что превышает указанные 20%. Несмотря на то что пользователи смогут заметить различие, они необязательно признают его *целесообразным* или впечатляющим.

Правило обеспечивает лишь минимальный критерий для порога обнаружения. Для получения различия, которое будет понято и оценено вашим пользователем, вам непременно надо выйти за границу, устанавливаемую порогом. Это не означает, что всякий раз вы должны стремиться к обеспечению возможно более высокой скорости выполнения. Как отмечено в главе 4, бывают ситуации, когда ваша система или ваше решение оказывается слишком быстрым. Если время отклика

уже настолько мало, насколько это допустимо в использовании и возможно технически, и имеются основания предполагать, что клиенты или пользователи все же не удовлетворены, то актуальным становится управление толерантностью, которое рассмотрено в главе 10 «Методы».

## Определение допустимой регрессии

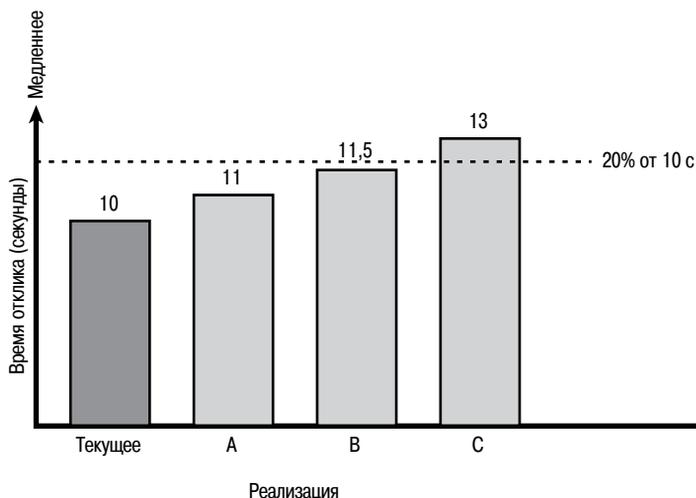
В идеальном компьютерном мире ваш продукт должен с каждой новой версией становиться более компактным, более элегантным, более мощным, более безопасным и более производительным (*performant*)<sup>1</sup>. В действительности же каждая итерация требует компромиссов. Например, ранняя версия вашего приложения требовала для установки 60 Мб пространства жесткого диска, а более поздняя версия из-за своей дополнительной функциональности требует объема памяти, в десять раз большего. Или ранняя версия могла предлагать 200 функциональных средств и требовала немалого объема памяти<sup>2</sup>, а новая версия требует существенно меньшего объема памяти, но и располагает меньшим набором функций. Иногда компромиссы способствуют ухудшению характеристик распределения времени. К примеру, для реализации усиленной защиты ваше приложение может потребовать дополнительного времени при загрузке. Это снижает производительность.

Исходя из того, что правило 20% относится к установленной зоне отсечения, где обнаружение является возможным, вы можете применить его реверсивным способом в своих интересах для минимизации вероятности обнаружения. В такой ситуации вы можете использовать значение 20% как верхнюю границу допустимой регрессии. Например, если различные реализации новых мер безопасности вызывают систематическое снижение производительности, вы можете использовать правило для построения зоны отсечения при выработке сбалансированного компромисса (рис. 5.6). Предположим, что раньше приложению для загрузки требовалось десять секунд. С первой реализацией новых мер время загрузки выросло на секунду. Со второй реализацией добавилось 1,5 с. С третьей – три секунды. В этих условиях наилучшей реализацией является вторая, которая дает ухудшение на 15% без нарушения границ зоны отсечения размером 20%.

---

<sup>1</sup> На момент написания книги термин «производительный» (*performant*) не включен в стандартный словарь, однако он означает «с большими возможностями» (*highly capable*) или «эффективный» (*effective* по-французски). Термин часто используется в промышленности для описания достаточной производительности.

<sup>2</sup> Объем пространства на жестком диске или память, потребляемая компьютерной программой либо файлом.



*Рис. 5.6. В этом примере текущее время отклика составляет десять секунд, однако включение новой функции неизбежно будет стоить некоторых потерь во временных характеристиках. Здесь реализация В новой функции является наилучшей, потому что не выходит за границы зоны 20% вокруг текущего времени отклика и тем самым понижает шансы того, что пользователь обнаружит различие*

## D2: деление надвое посредством среднего геометрического

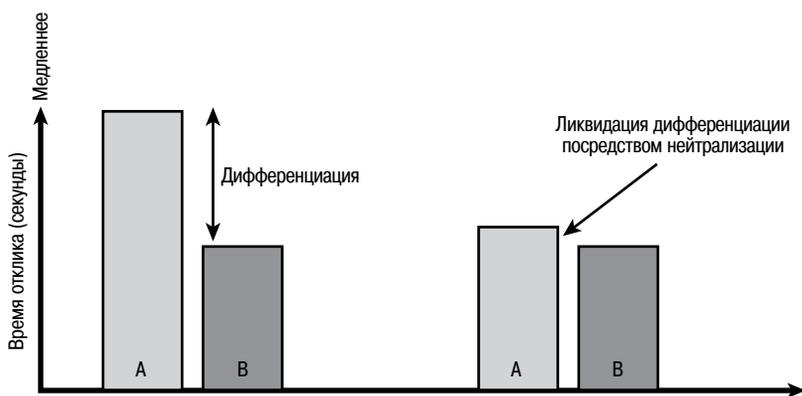
Толчком к улучшению временных характеристик не всегда бывает необходимость усовершенствования, вызванная тем, что ваши клиенты или пользователи не удовлетворены текущими характеристиками. Зачастую вам надо улучшить временные характеристики потому, что конкурирующий эквивалент вашего продукта обеспечивает явно превосходящие показатели. В идеале имеет смысл в ответ вывести рабочие характеристики за пределы зоны соперничества, но в реальной жизни, полной ограничений, это не всегда удается.

Если продукт или служба приобрели существенные отличия от вашего, остается только предпринять должные шаги для сохранения вашего продукта на плаву. Как уже отмечалось, Дж. Мур в своей книге «Dealing with Darwin» описывает преимущества нейтрализации дифференциации:

*Первая из желательных альтернатив исходит из нейтрализации. Ее целью является ликвидация дифференциации путем сосредоточения усилий либо на производительности, которая должна*

*превзойти конкурирующую, либо на понижении рыночных критериев характеристики... Компания Netscape добилась дифференциации своего веб-браузера; тогда компания Майкрософт добилась ее нейтрализации. В 1970-х годах банк Citibank добился дифференциации своих банкоматов по отношению к машинам АТМ; тогда все другие банки добились ее нейтрализации.*

Иными словами, рекомендация Мура состоит в том, что иногда, вместо того чтобы пытаться превзойти конкурента, стоит предпринять минимальные шаги для обеспечения характеристик, находящихся на одном уровне с конкурирующими или хотя бы близко к ним, так чтобы потребители перестали считать, что одни продукты существенно превосходят другие (рис. 5.7).



*Рис. 5.7. Слева реализация B обеспечила дифференциацию посредством заметного улучшения характеристик. Справа реализация A обеспечила характеристики, близкие к характеристикам B, тем самым ликвидировав дифференциацию путем ее нейтрализации*

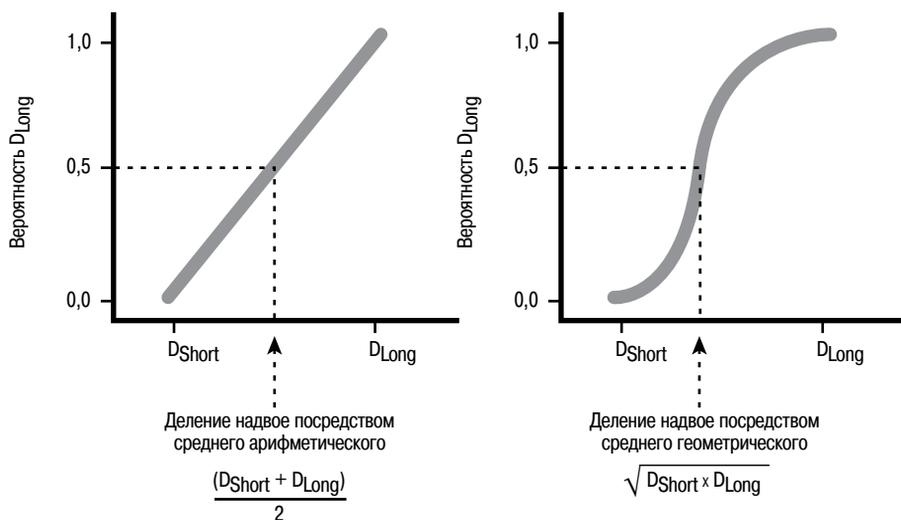
## Критерий типа «ненамного»

«Лакмусовой бумажкой» для предлагаемой Муром ликвидации дифференциации путем ее нейтрализации является *критерий типа «ненамного»*, то есть при внимательном рассмотрении различия хотя и могут быть обнаружены, но не являются существенными. К примеру, критики, оценивающие два или более продуктов, могут сделать вывод, что один продукт превосходит другой, но *ненамного*. Если ваш продукт удовлетворяет критерию типа «ненамного», дифференциация, созданная конкурентом, начинает уменьшаться.

Предположим, ваша компания разрабатывает программное обеспечение для резервного копирования данных и текущая версия вашего продукта требует 60 с для сохранения 1 Гб данных. Конкурент создал эквивалентный продукт, которому для выполнения той же работы требуется 30 с. Технические и другие ограничения не позволяют вам добиться быстродействия выше 30-секундного, но некоторые улучшения

технически возможны. Как вы должны задать новую эксплуатационную характеристику? Более конкретно, к какому новому значению времени вы должны стремиться, чтобы началась «нейтрализация дифференциации»? Где лежит зона критерия типа «ненамного»? Диапазон в гипотетическом примере, согласно исследованиям оценивания времени человеком, составляет примерно 42 с. Каково происхождение этого числа?

Пусть даны две длительности, одна больше другой. Где середина между ними? Многие исследования чувственного оценивания, в особенности проведенные Р. Черчем из университета Брауна и его коллегами, показывают, что середина между двумя длительностями, или результат *деления надвое*, предсказуема, но не совпадает с их средним значением (рис. 5.8). Воодушевленные этими успешными исследованиями на животных, таких как крысы и голубы, другие исследователи восприятия времени определяли середину с человеческой точки зрения и получили те же результаты: средняя точка между двумя длительностями совпадает не со средним арифметическим, а со средним геометрическим этих длительностей.



**Рис. 5.8.** Функция слева показывает, что среднее значение двух длительностей воспринимается как равное их среднему арифметическому. Исследования ощущения времени животными и людьми показали, что это не так. За середину двух длительностей принимается среднее геометрическое, что иллюстрирует функция справа

## Размывание дифференциации нейтрализацией

Среднее геометрическое двух величин – это просто квадратный корень из произведения величин. В предыдущем примере с резервным копированием информации при заданной паре длительностей 30 и 60 с можно определить геометрическое среднее путем извлечения квадратного корня из произведения 30 и 60, в результате чего получится 42.

Исследование деления надвое говорит нам, что выше порога среднего геометрического вероятность ассоциирования значения с большей величиной из пары величин увеличивается. Это означает, что значения выше 42 будут, скорее всего, восприниматься как 60, а не как 30. И наоборот, значения ниже 42 будут, скорее всего, оцениваться как более близкие к 30, а не к 60. Поэтому среднее геометрическое размечает зону отсечения и задание в примере целевой характеристики ниже 42 с (разумеется, чем ниже, тем лучше) теоретически обеспечит соответствие критерию типа «ненамного».

### Управление дифференциацией

Точно так же как правило 20% можно применять к определению целесообразного улучшения характеристик и допустимой регрессии, критерий типа «ненамного» можно применять в обратном направлении. Предположим, что время выполнения резервного копирования в вашем продукте составляет 30 с для 1 Гб данных, но новая реализация внесет некоторое ухудшение. Вы можете применить правило 20% и, с целью снизить вероятность восприятия ухудшения пользователем, сделать так, чтобы временная характеристика новой реализации не превышала 36 с. Однако, в силу того что та же характеристика у конкурента составляет примерно 60 с, можно нарушить границу 20% и обеспечить соответствие критерию типа «ненамного», установив значение 42 с.

### Проблематичная дифференциация

Математически возможно, что значение, удовлетворяющее критерию типа «ненамного», будет ниже, чем граница зоны отсечения 20%. Например, если одна длительность составляет 30 с, а другая – 35 с, то значение, удовлетворяющее критерию типа «ненамного», равняется 32 с, но граница зоны отсечения 20% на четыре секунды выше и равна 36 с. Это теоретически означает, что различие, не удовлетворяющее критерию «ненамного», невозможно обнаружить. В этом случае если существование дифференциации очевидно, то есть вероятность, что она объясняется не различием времен, а еще и другими факторами, например восприятием качества.

## Выводы

Понимание того, как пользователи воспринимают и обнаруживают изменения распределения времени, поможет вам определить требуемые эксплуатационные характеристики. Эта глава описывает в терминах D-уровней (количество вовлеченных в анализ длительностей) два сценария. Для сценариев типа D1, имеющих дело с единственной длительностью, для установления целевой производительности и определения допустимой регрессии применимо правило 20%. Для сценариев типа

D2, включающих две длительности, для ликвидации дифференциации посредством ее нейтрализации или для управления дифференциацией применим критерий типа «ненамного».

## Кроличья нора

### Различия в распределении времени

Hirsh, I. J. Auditory perception of temporal order. *Journal of the Acoustical Society of America*, 31, 1959. 759–767.

Luce, R. D. and E. Galanter. Discrimination. In R. D. Luce, R. R. Bush and E. Galanter (Eds.), *Handbook of Mathematical Psychology* (Vol. I, 191–243). New York: Wiley, 1963.

Michon, J. A. Timing in Temporal Tracking. Assen, NL: van Gorcum, 1967.

Repp, B. H. Compensation for subliminal timing perturbations in perceptualmotor synchronization. *Psychological Research*, 63, 2000. 106–128.

### Закон Вебера

Ekman, G. Weber's law and related functions. *The Journal of Psychology*, 47, 1959. 343–352.

Fechner, G. T. *Elements of Psychophysics* (H. E. Adler, Trans.). New York: Holt, Rinehart & Winston, 1966. (Впервые эта работа была опубликована в 1860 году.)

Killeen, P. R. and N. A. Weiss. Optimal timing and the Weber Function. *Psychological Review*, 94, 1987. 455–468.

Weber, E. H. De subtilitate tactus. (The sense of touch.) In H. E. Ross, D. J. Murray and J. D. Mollon (Eds.). *The Sense of Touch* (19–138). London: Academic Press, 1978. (Впервые эта работа была опубликована в 1834 году.)

### Дифференциация

Moore, G. *Dealing with Darwin: How great companies innovate at every phase of their evolution*. New York: Penguin Group, 2005.

### Деление надвое посредством среднего геометрического

Alan, L. G. Human bisection at the geometric mean. *Learning and Motivation*, 22, 1991. 39–58.

Church, R. M. and M. Z. Deluty. Bisection of temporal intervals. *Journal of Experimental Psychology: Animal Behavior Processes*, 3, 1977. 216–228.



# Глава 6

## Индикация выполнения

Индикация выполнения принадлежит к числу наиболее важных предметов обсуждения, потому что касается одного из мучительных состояний человека – ожидания. Из-за неправильного выбора типа индикации выполнения сравнительно малая длительность может показаться невыносимой, тогда как использование верного типа индикации может способствовать терпимому отношению пользователей к большим длительностям. Эта глава знакомит с классификацией видов индикации выполнения и объясняет, как им можно воспользоваться для определения наиболее подходящего типа индикации.

### Ошибки кремния

Когда речь идет об индикации выполнения, чаще всего в голову приходит образ вращающихся песочных часов или летящих документов, однако индикация выполнения – это нечто большее, чем произвольно выбранная анимация, заполняющая паузу. В главах 3 «Время отклика пользователя и системы» и 4 «Реактивность» я проводил аналогию между человеко-машинным взаимодействием и диалогом между людьми. Наряду с невербальным языком тела (к примеру, поднятие пальца вверх, для того чтобы показать просьбу немного подождать ответа) и словесным выражением (например, использование междометий типа «ух» или «хм», для того чтобы показать, что вы задумались), индикация выполнения является одним из наиболее важных и тонких каналов связи система–человек. Более того, аналогично промахам, слушающимся в человеческой беседе (например, прерывание собеседника на полуслове или отсутствие внимания в процессе беседы), возможны нарушения индикации выполнения, иногда просто грубые.

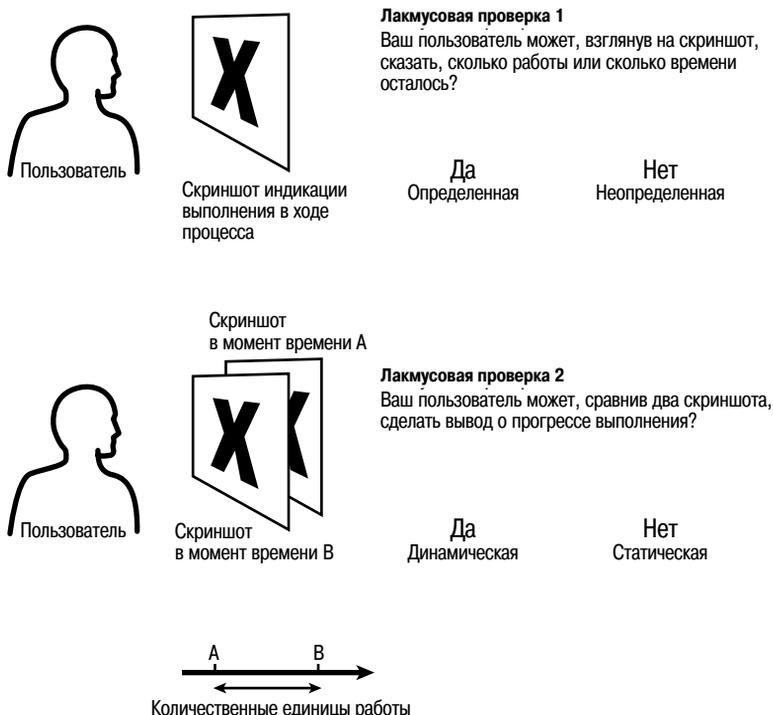
Независимо от того, действительно ли вам интересно читать эту главу или вы делаете это нехотя, возможно, (1) у вас есть индикация выполнения какого-то типа и в самый раз будет задаться вопросом, является ли она наилучшей для использования, или (2) некоторый процесс в вашем решении заставляет пользователя ждать и вы хотите знать, какой вид индикации выполнения следует применить, если это вообще необходимо. Давайте начнем с ситуации, когда индикация выполнения уже присутствует в вашем решении. В качестве первого шага надо определить тип вашей индикации выполнения с учетом особенностей процесса (загрузка из Интернета, обновление, поиск в базе данных, загрузка приложения и т. д.). После того как вы идентифицируете тип или класс своей индикации выполнения, следующий шаг будет состоять в проектировании и тонкой настройке индикации с учетом особенностей ваших пользователей.

## Классификация индикации выполнения

Не все виды индикации выполнения создаются одинаковыми. Для определения типа уже имеющейся у вас индикации выполнения вы можете произвести простое тестирование, состоящее из двух частей и называемое мною *двойной лакмусовой проверкой* (рис. 6.1). Представим, что вы сделали скриншот вашей индикации выполнения в процессе ее функционирования. Может ли пользователь по этому скриншоту сделать обоснованное заключение об оставшейся работе или времени? Эта информация может выражаться в единицах работы (например, число найденных файлов, размер копируемых файлов, число записанных песен, объем загруженных файлов, доля выполненной работы и т. д.) или в единицах времени (чаще всего, сколько времени осталось в секундах, минутах и т. д.). Если ваши пользователи могут сказать или сделать вывод о том, сколько работы или времени осталось, значит, индикация выполнения является *определенной*, по ней пользователь может прогнозировать завершение процесса. Если же ваши пользователи не могут сказать, сколько работы или времени осталось, значит, ваша индикация выполнения является *неопределенной*.

Во второй части проверки представим, что вы сделали второй скриншот после того, как была выполнена некоторая выражаемая количественно работа<sup>1</sup>. В качестве единицы работы может рассматриваться некоторая часть выполнения процесса, которую вы можете определить по коду или какими-то другими способами. Теперь задайте себе вопрос, могут ваши пользователи обнаружить различие между первым и вторым скриншотами или оба скриншота выглядят одинаково. Если могут, то индикация выполнения является *динамической*, потому что

<sup>1</sup> Если вы не можете выбрать количественные единицы измерения работы, то произвольно установите две временные точки в пределах выполняемого процесса, например две и пять минут.



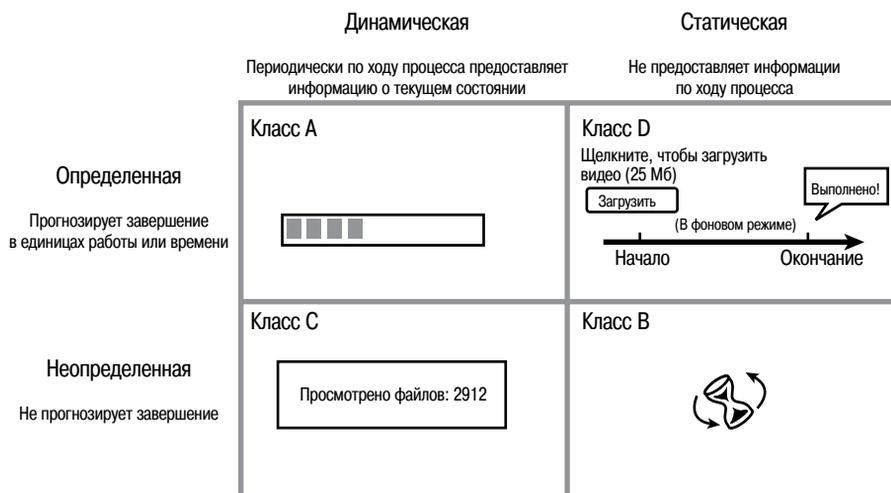
**Рис. 6.1.** Двойная лакмусовая проверка: в первой части теста если пользователи при взгляде на скриншот вашей индикации выполнения могут сказать, какое количество работы или времени осталось, значит, это определенный индикатор выполнения. В противном случае индикация является неопределенной. Во второй части теста если ваши пользователи могут указать разницу между двумя последовательными скриншотами, значит, ваша индикация выполнения является динамической. В противном случае она статическая

состояние процесса непрерывно обновляется и соответствующее отображение предоставляется пользователю. В противном случае она *статическая*, потому что пользователю не предоставляется периодическое отображение обновленного состояния. Интересно было бы определить минимальное количество единиц работы, по которому пользователь может обнаружить различие между любыми двумя последовательно снятыми скриншотами, и проанализировать его продолжительность. В идеале, если вы не хотите терять внимание пользователя, вам нужно выбрать время, затрачиваемое на выполнение объема работы, не превышающее семь секунд (самое большее – десять секунд). Это означает, что надо поставлять некоторую информацию в пределах последовательных временных блоков, которые принадлежат классу вынуждающей реактивности (глава 4). Однако для случаев, когда общая длительность очень велика (несколько минут или более), лучше не привлекать к ней внимание пользователя.

Если ваша индикация выполнения является определенной и динамической, то вы располагаете, по моему определению, *индикацией выполнения класса А*. Индикация этого класса активно снабжает пользователя информацией о том, какое количество работы или времени осталось до завершения процесса. Если ваша индикация выполнения является неопределенной и статической, она попадает в *класс В индикации выполнения*, который чаще всего называется индикацией «занято» или «в работе». Такая индикация должна поставлять обновленную информацию через короткие интервалы времени, потому что иначе пользователи остаются привязанными к процессу, не имея сведений о том, когда он завершится. Если ваша индикация является неопределенной и динамической, то она принадлежит *классу С индикации выполнения*. Она аналогична индикации класса А, за исключением того что, не имея возможности прогнозировать завершение, активно снабжает пользователей информацией о выполненной работе или об изменениях в состоянии процесса. Если ваша индикация является определенной и статической, она принадлежит *классу D индикации выполнения*. Она дает какой-то прогноз завершения, но не предоставляет информацию о длительности процесса. Это характерно для процессов, которые «тонут» в фоновом режиме и затем выдают сообщение о своем завершении. На рис. 6.2 представлены примеры для всех рассмотренных классов.

Можно легко запомнить особенности классов, основываясь на том, что А означает *active* (активный), при такой индикации активно отображается оставшееся количество единиц работы или времени. В означает *busy* (занято), что отражает занятость системы. С означает *changes* (изменения), такая индикация сообщает об изменениях состояния процесса. D означает *disappear* (исчезает), при такой индикации индикатор исчезает и затем появляется при завершении процесса.

Чтобы рассмотреть классы с другой точки зрения, представьте себе посещение большого города в первый раз. Класс А напоминает ситуацию, когда вы знаете, что для того, чтобы попасть в пункт назначения, вам надо проехать еще пять городских кварталов. Класс В отличается лишь тем, что вы не знаете, ни сколько кварталов вы проехали, ни сколько кварталов должны еще проехать. Для класса С можно сказать, что вы знаете, что проехали три квартала, но ничего не знаете о пункте назначения. В случае класса D вы наперед точно знаете, насколько далеко находится пункт назначения, и вас проинформируют, когда он будет достигнут. Как отмечено в начале этого раздела, ни у одного класса нет преимущества перед другим. В зависимости от контекста тот или иной класс является более подходящим, чем другие. Кроме того, можно применять сочетание двух или более классов. Какой же класс индикации выполнения подходит вам? Читайте дальше.



*Рис. 6.2. Если индикация выполнения динамическая и определенная, она принадлежит классу А. Здесь она представлена линейным индикатором выполнения. Если индикация выполнения статическая и неопределенная, она принадлежит классу В. Таковы широко распространенные индикаторы «занято» или «в работе». Если индикатор выполнения является динамическим и неопределенным, он принадлежит классу С. Эта индикация подобна индикации класса А, но без возможности прогнозировать завершение. Если индикатор выполнения является статическим и неопределенным, он принадлежит классу D. Эти индикаторы, такие как всплывающее сообщение, дают понять, что процесс завершен, но не информируют о ходе выполнения, пока завершение не произойдет*

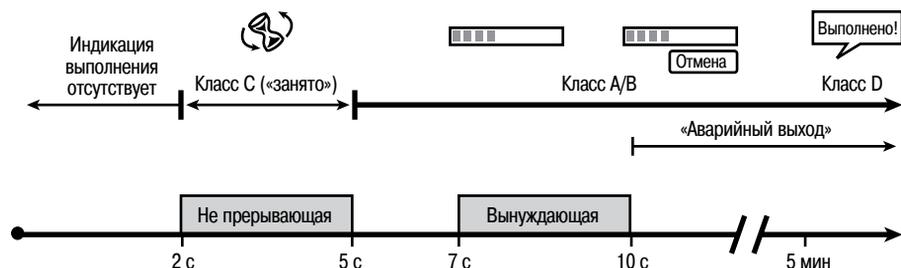
## Выбор правильного класса

В главе 4 я привел первую рекомендацию по индикации выполнения, рассматривая класс не прерывающей реактивности. А именно, для принятия решения о том, нужна ли вообще индикация выполнения, вы можете использовать нижнюю и верхнюю границы интервала, который пользователь считает не прерывающимся. Для процессов, завершающихся менее чем за две секунды, обычно нет нужды в какой-либо индикации выполнения.

Предупреждение: эта рекомендация предполагает, что для интересующего вас процесса взаимодействия правильно определен класс реактивности. К примеру, список, раскрывающийся по щелчку мыши, лучше связать с классом мгновенной реактивности. Вспомните, мы ранее говорили о понимании пользователем того, что компьютеру для выполнения определенных задач требуется «время на обдумывание». Так что общим критерием процессов, обсуждаемых в этой главе, является то, что компьютеру требуется некоторое время для их «обдумывания».

Сохранение документа, посещение веб-сайта, просмотр истории транзакций, запуск приложения и выполнение поиска – яркие примеры таких процессов.

При превышении двух секунд индикация выполнения в том или ином виде необходима. Между двумя и пятью секундами обычно пользователю бывает достаточно простой индикации того, что система функционирует (класс В). Индикация может даваться в форме характерной анимации «занятости» или просто строки текста, сообщающей, что работа продвигается, например «Сохранение документа». При превышении пяти секунд предоставление пользователю какого-то вида активной индикации выполнения (класса А или С) является критически важным. Это не означает, что для индикации недопустимы кратковременные паузы. Она предназначена для периодического информирования пользователя о том, что выполнение продвигается. Другой класс реактивности (вынуждающая) вступает в игру, если время выполнения занимает больше десяти секунд. В этом случае вы должны предусмотреть «аварийный выход» (обычно это кнопка Отмена) для предоставления пользователю возможности прервать или отменить процесс. Для длительностей, которые предположительно составят не менее пяти минут, актуальным является предоставление извещения о завершении (класс D). На рис. 6.3 показаны основные классы индикации выполнения и диапазоны длительности.



**Рис. 6.3.** Схема иллюстрирует зависимость применимости различных классов индикации выполнения от классов реактивности, рассмотренных в главе 4

Большая часть ваших индикаторов выполнения будет относиться к процессам, длительность которых либо значительно превышает пять секунд, либо может непредсказуемо изменяться. Рассмотрев классификацию индикации выполнения для определения пригодности конкретных ее классов для конкретных диапазонов длительностей, в следующем разделе мы сделаем еще один шаг к проектированию и настройке индикации выполнения.

## Проектирование индикации выполнения

Попросту говоря, индикация выполнения представляет пользователю информацию о состоянии процесса или взаимодействия, но не всегда бывает правильно устроена. В зависимости от ряда факторов и контекста одна форма индикации может быть более уместна, чем другая. Мы обратим внимание на три существенных аспекта, которые помогут вам спроектировать оптимальную индикацию выполнения или произвести тонкую настройку той индикации, которая у вас уже есть. Имейте в виду, что вы необязательно должны делать выбор из альтернативных вариантов в рамках каждого аспекта; более вероятно, что вы соедините и согласуете различные аспекты ради удовлетворения требований вашего пользователя. Классификацию из предыдущего раздела можно сравнить с принятием решения, что приготовить к столу – завтрак, ланч или обед, а материал этого раздела – с принятием решения, какую конкретно еду вы будете готовить. Три ингредиента для приготовления вашей еды – это вид отображения, единица продвижения выполнения и тип данных.

### Вид отображения: текстуальный или визуальный

Самый очевидный способ описать индикацию выполнения – это определить, что вы используете для представления выполнения: текст, визуальные объекты или сочетание и тех и других. Самые современные виды индикации выполнения, вероятно, будут пользоваться обоими<sup>1</sup>. Многие предметы бытовой электроники, снабженные маленькими жидкокристаллическими дисплеями, используют индикацию выполнения, основанную на тексте. Другие жидкокристаллические дисплеи выполнены с большей изобретательностью и способны средствами текста имитировать графику и анимацию. Это же характерно для консольных, основанных на командной строке, приложений, лишенных графического пользовательского интерфейса. Визуальные объекты включают широко используемые линейные индикаторы выполнения, летящие документы, вращающиеся песочные часы и т. д.

#### Пользуйтесь текстом, когда

- Для пользователя важны, значимы или содержательны сведения о ходе процесса.
- Пользователи могут и хотят уделять внимание чтению текста.

---

<sup>1</sup> Можно также применять звуковые индикаторы, например воспроизводить аудиофайл при завершении процесса.

Очевидный аргумент в пользу применения текста состоит в том, что это лучший вид описания хода процесса и его основных операций. Следовательно, когда описание хода процесса для пользователя важно, значимо или содержательно, для индикации выполнения предпочтителен текст. Пользователю может потребоваться увидеть описание хода процесса, к примеру, в случае если для него представляют интерес промежуточные сбои или промежуточные процессы. Например, торговый представитель во время получения пользовательской информации не нуждается в сведениях о базах данных, файлах динамически подключаемых библиотек или процессах ввода/вывода. С другой стороны, администратору базы данных может потребоваться более детальное описание критически важных процессов, таких как резервирование информации и т. д. В действительности для критически важных процессов, подобных упомянутому, принято, выходя за рамки отображения описания, давать подробный, типа «строка-за-строкой», отчет о ходе процесса в файле системного журнала.

Вывод текста для индикации выполнения полезен только в том случае, если пользователи могут и хотят читать текст. Следовательно, текст никогда не следует использовать, если пользователи либо не могут читать из-за ограниченного объема внимания, либо не хотят читать слова по каким-то другим причинам. Вообще, использовать текст для индикации выполнения хорошо всегда; вопрос в количестве. Для индикации класса В («занято» или «в работе»), возможно, достаточно одного или двух содержательных слов. Хотя исследования показывают, что люди читают примерно от четырех до пяти слов в секунду, вы должны сводить количество слов к минимуму, стараясь достичь наибольшей их выразительности. Рис. 6.4 иллюстрирует мысль о том, как можно представить информацию о выполнении одного и того же процесса различным количеством слов.

Многие примеры индикации выполнения представляют собой сообщения типа «Пожалуйста, подождите», «Оставайтесь на линии» или «Одну минуту». Такие фразы обычно используются, когда оставшееся количество работы или времени неопределенно. В силу того что они не обеспечивают пользователя информацией во время его ожидания, такие фразы сами по себе являются текстуальными примерами индикации выполнения класса В. Поэтому они лучше подходят к длительностям между двумя и пятью секундами, ведь чем дольше отображается статическая индикация сверх пяти секунд, тем выше вероятность того, что пользователь сделает предположение о зависании системы. Если вы применяете индикацию выполнения класса В в ситуации, не соответствующей классу В, степень удобства использования и удовлетворения в конечном счете станет проблематичной.

Два способа улучшить ситуацию заключаются в наращивании существующей индикации в соответствии с подходящим классом и в преобразовании существующей индикации также в соответствии с подходящим



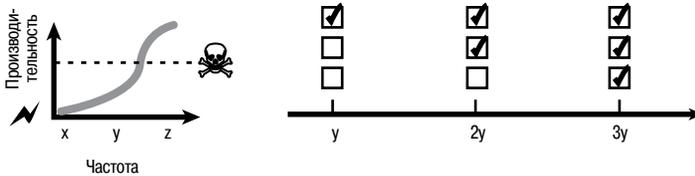
1. Определите, какая информация, численные показатели и т. д. доступны в ходе процесса...



2. Определите, кто ваши пользователи и что для них является содержательным...



3. Выберите компромисс между выполнением процесса и сопутствующими затратами.



**Рис. 6.5.** Многочисленные экземпляры индикации выполнения, содержащие сообщения типа «Пожалуйста, подождите», в действительности принадлежат классу В («занято/в работе»), являясь при этом текстуальными. При некоторых обстоятельствах, однако, например когда длительность сильно превышает пять секунд, этот класс может оказаться непригодным для использования. На этой схеме показаны три шага для исправления ситуации

широко из всех индикаторов выполнения применяется, по-видимому, линейный индикатор выполнения. Графика и символы также пригодны в тех случаях, когда объем внимания пользователя ограничивается необходимостью выбора или текущей задачей. Хирург, вероятно, не пожелает обращать внимание на текстовую полосу на жидкокристаллическом экране компьютерной системы мониторинга жизненно важных показателей своего пациента. Если эти показатели близки к опасному уровню, более уместно включение красных лампочек и громких

сигналов<sup>1</sup>. Иногда пользователи сами решают не обращать внимания на ход процесса, потому что больше интересуются его завершением, чем событиями, происходящими на его протяжении. В таких случаях высока вероятность того, что пользователи станут уделять внимание другим задачам и лишь изредка будут возвращаться к индикации выполнения, чтобы оценить продвижение. Рассмотрим, например, следующий гипотетический сценарий.

Покупатель, находясь дома, открывает веб-сайт, посвященный недвижимости, для поиска вариантов, но сталкивается с требованием установить Java-плагин, для того чтобы эти варианты увидеть. Щелкнув по нескольким кнопкам, он видит линейный индикатор выполнения, который сообщает о том, что загрузка началась. Не желая растратывать время на наблюдение за загрузкой и установкой, он открывает другую вкладку в своем веб-браузере, для того чтобы перейти на другой веб-сайт. Примерно 30 с спустя он возвращается к первому веб-сайту, чтобы посмотреть, завершились ли загрузка и установка. Он быстро замечает, что линейный индикатор выполнения заполнен менее чем наполовину, и немедленно переключается на второй веб-сайт для продолжения путешествия в Интернете.

Очевидно, что в этом обычном сценарии внимание, обращенное к индикации выполнения, незначительно и носит случайный характер, поскольку пользователь переключается на другие задачи. Поэтому предпочтительно использование простых визуальных объектов, которые позволяют пользователям за секунду или две понять, каково продвижение.

## Единица продвижения выполнения: время или работа

Для описания индикации выполнения надо выбрать единицу, используемую для выражения продвижения. В некоторых видах индикации выполнения в качестве такой единицы используется оставшееся или затраченное время, в то время как другие с той же целью применяют количество единиц выполненной, выполняющейся или оставшейся для выполнения работы. Чаще в индикации выполнения вы найдете сочетание единиц времени и работы. Например, многие приложения

---

<sup>1</sup> Использование звуков является прекрасным, разумным способом демонстрации хода процесса. Пользователи с некоторыми типами нетрудоспособности, а также операторы сенсорных систем или систем с повышенной опасностью (к примеру, пилот самолета-истребителя) тоже выигрывают от использования звука. Однако большинство основного контингента пользователей находятся в среде, где применение звука для демонстрации хода процесса отнюдь не является идеальным, это относится, к примеру, к офисным помещениям или библиотекам.

отображают скорость передачи битов, к примеру 128 Кбит/с, что в действительности является выражением количества работы (килобитов), приходящейся на фиксированную единицу времени (секунду).

### **Пользуйтесь единицами времени, когда**

- Вы можете уверенно прогнозировать время завершения.
- Пользователи с нетерпением ждут то, что появится после завершения.
- Пользователи, вероятно, будут работать с другими задачами в период ожидания завершения.
- Предусмотрено взаимодействие с другой активной стороной.

Единицы времени представлены двумя величинами: оставшимся и затраченным временем. Представление затраченного времени актуально лишь в небольшом количестве приложений, поэтому данный раздел фокусирует внимание на представлении оставшегося времени. Те приложения, где затраченное время оказывается полезным, почти всегда охватывают одно или больше осмысленных действий, которые могут быть предприняты пользователем по прошествии определенного интервала времени. Это справедливо для некоторых приложений обмена сообщениями и связи, в которых пользователю полезно знать, что другая сторона не отвечает, с тем чтобы он мог отправить повторное сообщение или предпринять другие необходимые действия. Другая ситуация, где отображение затраченного времени может считаться ценным, имеет место, когда критически важными являются измерения производительности или диагностика. В некоторых случаях затраченное время используется из-за того, что продвижение является неопределенным. Этот ход нельзя считать удачным, потому что пользователям буквально предоставляется средство количественного оценивания ожидания. Представим себе, что хозяйка ресторана появляется перед вами каждые пять минут, чтобы сообщить, как долго вы ждете еду! Повторим, что, если пользователь не должен производить какие-то действия по прошествии определенного промежутка времени, лучше сообщать ему оставшееся время, чем затраченное.

Единицы времени должны применяться только тогда, когда вы можете прогнозировать время завершения достаточно точно. Кроме того, единицы времени являются оптимальными, когда пользователи думают о том, что будет происходить после завершения процесса, поскольку никому не нравится ждать; знание того, что время ожидания сокращается, скрашивает тоску ожидания. Кроме успокоения нетерпеливых пользователей, сообщение оставшегося времени выполняет еще одну функцию – помогает пользователям решить, сколько времени они могут посвятить другим задачам, например могут ли они переключиться на другое приложение, выпить кофе или пойти пообедать.

Два предостерегающих замечания, касающихся количества единиц сообщаемого времени:

- При проектировании представления времени в виде некоторого количества единиц времени следует проявлять аккуратность. Это обсуждается в главе 7 «Представление времени», а сейчас лишь отметим, что ежесекундное сообщение обратно отсчитываемого оставшегося времени может привести к столь же квазиоптимальному опыту, как и прямой отсчет затраченного времени. Якоря времени (1, 2, 3, 5, 10, 15, 20, 30) и диапазоны (например, от пяти до десяти минут) во многих случаях производят лучшее впечатление, чем точное представление оставшегося времени (например, 8:30 мин).
- Количество единиц времени на протяжении малых интервалов времени должно либо оставаться неизменным, либо убывать. Оно никогда не должно расти. Увеличение оставшегося времени является бессмысленным и с полным основанием считается в среде пользователей идиотским. Однако такое явление встречается нередко и обычно имеет место, когда прогнозируемое время завершения вычисляется динамически на основе оставшейся рабочей нагрузки (например, количества копируемых файлов, просматриваемых папок и т. д.) в соответствии с доступными ресурсами, такими как производительность обработки, память, пропускная способность и т. д. В силу того что количество единиц времени в математических уравнениях находится «в одной упряжке» с ресурсами, любая заминка или остановка ресурса приводит к изменению уравнения и влечет за собой объявление необходимости дополнительных затрат времени, увеличивая тем самым значение оставшегося времени (рис. 6.6). Дж. К. Дворак (J. C. Dvorak) в одной из своих заметок в журнале *PC Magazine* описывает это как явление флуктуации времени: «Ладно, это большая папка, и вы принимаете решение спуститься на другой этаж, чтобы немного посмотреть телевизор. Копирование начинается. Все выглядит благополучно. Фактически процесс выполняется довольно быстро. Затем время возрастает до двух часов! Потом оно перескакивает к значению три часа! Затем возвращается к двум часам десяти минутам! Почему так трудно оценить время?»

Избавиться от явления флуктуации времени можно несколькими способами. В идеале, если вычислительные ресурсы могут быть зарезервированы для выполнения задачи, то прогнозируемое оставшееся время будет стабильным, оно не будет расти или колебаться. Это не всегда возможно, потому что вычислительные ресурсы, например центральный процессор и память, могут использоваться текущим процессом совместно со многими другими процессами. Более приемлемый выход заключается в разъединении количества единиц времени и вычислительного ресурса (или любого ресурса, подверженного высокой изменчивости) при вычислении, а также в том, чтобы представлять оставшееся время



*Рис. 6.6. Эта схема иллюстрирует явление флуктуации времени. При нормальной доступности ресурса прогнозируемое оставшееся время будет продолжать уменьшаться. Однако, если ресурсы вдруг становятся ограниченными (например, когда начинает функционировать другой процесс), время завершения пересчитывается и прогнозируемое оставшееся время внезапно возрастает. То, что пользователь видит в результате, — это флуктуации оценки оставшегося времени*

в виде диапазона, базирующегося на высшем и низшем уровнях вычислительного ресурса. Другой путь заключается в том, чтобы для формирования толерантности пользователей дать им, наряду с величиной оставшегося времени, некоторое представление о доступном ресурсе<sup>1</sup>. Такой подход часто используется при объемных загрузках из Интернета (если определена скорость передачи битов) и при установлении соединений по беспроводной сети (если определены интенсивность сигнала и скорость). Если так решить проблему невозможно, вам стоит подумать о применении других единиц индикации выполнения — единиц работы.

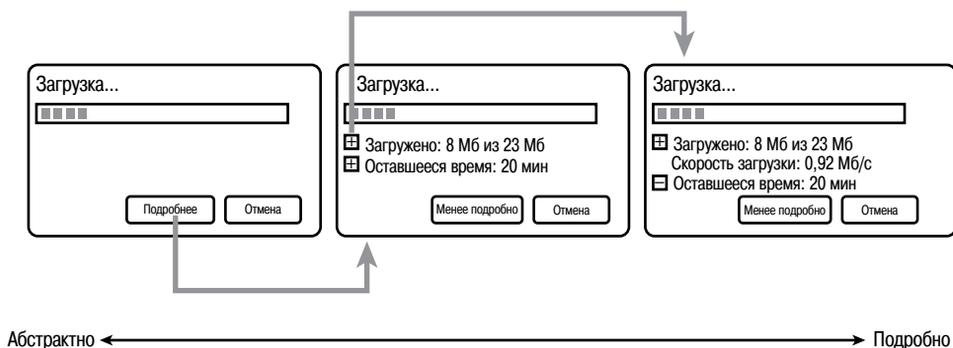
<sup>1</sup> Это также эффективный метод, который, по существу, перемещает «ответственность» на аппаратное обеспечение компьютера пользователя, с тем чтобы пользователи осознавали, что оно является ограничивающим фактором.

## Пользуйтесь единицами работы, когда

- Оставшееся время слишком изменчиво или непредсказуемо.
- Пользователи участвуют в планировании шагов выполнения.
- Промежуточные сбои представляют интерес и имеют практическое значение для пользователя.

Как сказано в предыдущем разделе, когда оставшееся время подвержено флуктуациям, единицы работы могут обеспечить лучшую альтернативу. На рис. 6.5 вверху показаны некоторые примеры единиц работы. Другое практическое правило состоит в том, чтобы пользоваться единицами работы, когда пользователь считает содержательными или значимыми сведения о работе, которая завершилась, о текущей выполняемой работе или о работе, оставшейся для выполнения. Условия соблюдаются, к примеру, если пользователи как-то вовлечены в планирование или настройку шагов выполнения. Отслеживание завершенных шагов дает пользователю уверенность в том, что они выполнены правильно. Замечательным примером служит запись песен на компакт-диск; когда вы узнаете об успешном завершении записи каждой песни, вы приобретаете определенную долю уверенности в том, что работа продвигается в соответствии с вашими ожиданиями. Другой случай, когда полезно отображать единицы работы, имеет место, если отклики на промежуточные сбои пользователю необходимы или представляются содержательными, в особенности когда сбои не задерживают процесс в целом.

Надо помнить о нескольких важных вещах, связанных с применением единиц работы. Во-первых, единицы работы необязательно всегда означают точное число записываемых байтов, копируемых файлов или папок, в которых производится поиск. Они также могут представлять этап установки, завершенное действие и т. д. Кроме того, единицы работы могут отражать процентное отношение, например «35% видео загружено», или некоторое текущее совокупное значение, например «3 из 5 файлов загружены». Во-вторых, всегда можно построить информационные слои, которые становятся все более детализированными и специальными, как только пользователи выбирают для просмотра следующий слой. Это называется *последовательным раскрытием*. Если вы включаете информационные слои, начиная с абстрактного, то пользователи, которые не стремятся понимать важные, но скучные подробности, не будут завалены и перегружены отображаемой информацией. Более продвинутые пользователи могут выбрать для просмотра более подробную информацию. На рис. 6.7 показан этот простой, но мощный прием.



*Рис. 6.7. Эта схема поясняет концепцию последовательного раскрытия. Вначале никакие специальные подробности не отображаются. Опытные пользователи затем могут пожелать раскрыть более подробную информацию (в данном случае – щелкнув по кнопке Подробнее). Если скорость загрузки представляется пользователю содержательной или значимой, он может решить просмотреть ее посредством перехода к соответствующему узлу дерева просмотра. Здесь идея заключается в таком управлении информацией, которое позволяет удовлетворить потребности и основного контингента, и продвинутых пользователей*

## Тип данных: количественный или качественный

Третий способ описать индикацию выполнения – это определить, количественной или качественной должна быть представляемая информация. Количественному типу данных сопутствует численное значение в единицах времени или работы, например «Осталось примерно 30 минут» или «Загружено 3 Мб из 12 Мб». Качественный тип данных обычно выражает некоторое состояние, этап или характеристику выполнения, например «Удаление временных файлов».

### Пользуйтесь количественными данными, когда

- Пользователям нужна объективная информация для анализа сбоев, производительности или другая диагностика.
- Эта информация дает уверенность в том, что работа, соответствующая определенному количеству единиц, успешно завершена.

Отображайте количественные данные, когда пользователю нужна объективная информация. Она является особенно полезной для тех пользователей, кому надо анализировать сбои, измерять производительность или решать другие диагностические задачи. Уместный пример – контроль с помощью циклического избыточного кода (Cyclic Redundancy Check, CRC) в управлении файлами и данными. CRC – это метод обнаружения ошибок, используемый для обеспечения уверенности в том, что файл передан правильно. Если вы не знакомы с CRC, представьте себе, что это извещение о том, сколько элементов должно оказаться в почтовой

посылке. При получении посылки вы хотите убедиться в том, что реальное количество элементов и количество, записанное в извещении, совпадают. Если числа не совпадают, значит, что-то не так. Точно так же при передаче файла CRC обеспечивает совпадение данных, прочитанных из источника, с полученной информацией. Ошибки чтения и записи информации на CD- и DVD-диски, к примеру, часто выявляются посредством CRC. Представьте себе, что пользователь, принадлежащий к основному контингенту, при записи на компакт-диск видит длинную и непонятную строку цифр, выдаваемую алгоритмом CRC во время чтения и записи каждой песни. Разумеется, для пользователя эта информация абсолютно лишена смысла. Все, что должно демонстрироваться пользователю в этом случае, – это лишь простые визуальные объекты, показывающие количество записанных на компакт-диск песен и состояние записи.

Неопределенность усиливает тоску ожидания. Отображение количественных данных может иногда дать некоторую уверенность в том, что все происходит правильно, даже если пользователи не вполне понимают эти действия. Уместный пример – антивирусное сканирование. Несмотря на то что основной контингент пользователей, скорее всего, не нуждается в знании того, какой именно файл сканируется (пока не надо предпринимать никаких действий, например, при обнаружении инфицированного файла), благодаря тому что пользователь в состоянии смотреть на бегущее число или на линейный индикатор выполнения, которые нарастают с каждым отсканированным файлом или группой файлов, он получает некоторую степень уверенности в том, что антивирусное приложение выполняет свою работу. Это яркий пример индикации выполнения класса C в действии. Вы не знаете, когда дело подойдет к концу (неопределенный), но вы знаете, что оно продвигается (динамический). Напомним, что здесь вы также можете пользоваться последовательным раскрытием.

### **Пользуйтесь качественными данными, когда**

- Количественные данные для пользователя бессмысленны.
- Пользователю удобнее прочесть простое описание выполненной работы.
- На промежуточных стадиях выполнения необходимо к чему-то привлечь внимание пользователя.

Один из видов использования качественных данных базируется на упрощенном виде единиц работы – этапах. Это приносит пользу, когда реальные единицы работы для пользователя лишены смысла. Например, вместо сообщения о том, что 251 установочный файл и 36 вспомогательных файлов скопированы в четыре временные папки, будет лучше просто сказать, что все установочные файлы успешно скопированы. Следовательно, качественные данные очень хороши для подведения

итога общей выполненной работы. Многие операции кажутся пользователям слишком специальными, например упомянутый ранее CRC, однако сведения об этапах могут быть полезными и успокаивающими. Представьте, к примеру, что у вас есть подрядчик, периодически докладывающий вам о количестве винтов и шурупов, используемых при перепланировке вашей кухни. Качественная информация в этом случае состояла бы в том, что подрядчик сообщил бы вам, что все ваши шкафчики уже сделаны и что он переходит к кухонному столу.

Качественные данные также уместны, когда требуется привлечь к чему-то внимание пользователя. Некоторые производители материнских плат, например Intel, встраивают повсюду в материнскую плату датчики для слежения за ее рабочей температурой, потому что перегрев может нанести непоправимый вред. Производители могли бы предоставить программное обеспечение<sup>1</sup> (у Intel оно есть и носит название Active Monitor), которое отображало бы текущую температуру (количественно) наряду с использованием качественной информации, например в виде мигающей красной надписи «Материнская плата перегрета!», что определенно привлекло бы внимание пользователя гораздо лучше.

## Выводы

Эта глава знакомит вас с четырьмя классами индикации выполнения. Класс А представляет определенную и динамическую индикацию, класс В – неопределенную и статическую, класс С – неопределенную и динамическую, а класс D – определенную и статическую индикацию. Вы можете осуществить тонкую настройку индикаторов выполнения, выбрав подходящий вид отображения (текстуальный/визуальный), единицы продвижения выполнения (время/работа) и тип информации (количественный/качественный).

## Кроличья нора

### Виды индикации выполнения

Apple Developer Connection. Indicators, 2006. (Примечание автора: вы можете найти онлайн-версию по адресу [http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIIGuidelines/XHIGControls/chapter\\_18\\_section\\_5.html](http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIIGuidelines/XHIGControls/chapter_18_section_5.html).)

---

<sup>1</sup> Многие производители материнских плат включают такую функциональность, но считывание температуры скрыто в настройках BIOS, которые обычно доступны только (путем нажатия назначенной функциональной клавиши и иногда ввода пароля) в течение нескольких первых секунд загрузки компьютера. Как вам это нравится с точки зрения открытости и доступности?

Johnson, J. GUI Bloopers: Don'ts and Do's for Software Developers and Web Designers. San Diego, CA: Academic Press, 2000. (Примечание автора: См. раздел «Chapter 7».)

McInerney, P. and J. Li. Progress indication: Concepts, design, and implementation. IBM developerWorks, 2002. (Примечание автора: вы можете найти онлайн-версию по адресу [www.ibm.com/developerworks/library/us-progind/](http://www.ibm.com/developerworks/library/us-progind/).)

Myers, B. A. Importance of percent-done progress indicators for computer-human interfaces. CHI '85 Proceedings. ACM, 1985.

## Явление флуктуации времени

Dvorak, D. C. Windows' words of doom. *PC Magazine*, May 2, 2007. (Примечание автора: вы можете найти онлайн-версию по адресу [www.pcmag.com/article2/0,1759,2123848,00.asp](http://www.pcmag.com/article2/0,1759,2123848,00.asp).)

## Последовательное раскрытие

Fowler, S. and V. Stanwick. Web application design handbook: Best practices for Webbased software. San Francisco: Morgan Kaufmann Publishers, 2004. (Примечание автора: См. стр. 157 о последовательном раскрытии.)

Hmelo-Silver, C. and H. S. Barrows. Goals and strategies of a problem-based learning facilitator. *Interdisciplinary Journal of Problem-based Learning*, 1, 2006. 21–39. (Примечание автора: последовательное раскрытие, применяемое вне человеко-машинного взаимодействия.)

Nielsen, J. Progressive Disclosure. *Jakob Nielsen's Alertbox*, December 4, 2006. (Примечание автора: вы можете найти онлайн-версию по адресу [www.useit.com/alertbox/progressive-disclosure.html](http://www.useit.com/alertbox/progressive-disclosure.html).)

# Глава 7

## Представление времени

Во множестве случаев, в частности при индикации выполнения, вы должны представить пользователю какую-то информацию о времени, например сообщить значение оставшегося времени или оценку времени завершения. Это может показаться тривиальным и не заслуживающим внимания, но от способа представления времени в UI может зависеть успешность работы и восприятие вашего решения пользователем. Одним словом, – к лучшему или к худшему – способ представления времени может изменить восприятие и повлиять на толерантность. В этой главе обсуждается вопрос о том, когда и как вам следует представлять время в рамках UI вашего решения; вы также ознакомитесь с применением якорей времени.

### Распределение во времени: прошлое, настоящее, будущее

Помимо решения вопросов о том, *что* вы представляете (фразы, единицы времени и т. д.) и *как* вы представляете (в текстовом, графическом виде и т. д.), вы должны подумать о том, *когда* вы даете пользователю информацию о распределении времени. Например, если сказать тому, кто собирается встать в очередь, какое время понадобится, чтобы добраться до ее начала, и те же сведения сообщить человеку, уже отстоявшему пол-очереди, то результаты окажутся различными; такое информирование преследует разные цели. Проще говоря, представление информации может как улучшить, так и испортить впечатление. Каждый из нас знает несколько ужасных историй о том, как рассинхронизация некоторой информации, иногда всего на несколько секунд, изменяла ход событий.

Люди используют информацию, доставляемую посредством UI, для того чтобы – вольно или невольно – составить мнение о взаимодействии или процессе. Это вас не должно удивлять, потому что все мы ежедневно занимаемся чем-то подобным. Мы прогнозируем, опираясь на знакомые примеры, которые видим (длинная очередь означает длительное ожидание), оцениваем качество вещей, учитывая признаки и симптомы (дефекты обусловлены недобросовестностью исполнения), и строим теории, объясняющие происходящее (сломалось из-за плохого качества деталей). Аналогичным образом, пользователи будут привлекать *любую* информацию о распределении времени, для того чтобы понять и сформировать свое отношение к длительности взаимодействия.

Рассмотрим следующие вопросы:

- Сколько всего времени это займет?
- Сколько еще времени это займет?
- Сколько времени это заняло?

Каждый из этих трех вопросов привязан к особому временному ракурсу, то есть к тому, каково ваше положение во времени относительно времени рассматриваемого события. Мы в состоянии воспринимать длительности из трех основных временных ракурсов. Во-первых, мы можем предвидеть, формировать ожидания или прогнозировать длительность события до его начала. В силу того что событие еще не происходило, мы назовем такое оценивание длительности *предварительным*. Например, в UI информирование пользователей о том, какова будет продолжительность загрузки, дает пользователям предварительную оценку длительности загрузки. Во-вторых, мы можем оценивать длительность события в *режиме реального времени*, пока оно происходит. Характерным примером является представление оставшегося времени в ходе загрузки. Наконец, мы можем оценивать длительность события после его завершения. Это последнее оценивание мы назовем *ретроспективным*. Примером является представление пользователю информации о том, сколько времени заняла загрузка. Как отмечалось, то, *когда* вы представляете информацию о длительности какого-то события, может влиять на восприятие пользователя и формировать характер пользовательского отношения. Рассмотрим каждый ракурс подробнее.

## Предварительно: Tickle-Me-Elmo

Не всегда очевидно, что на восприятие можно не только повлиять, но и вообще сформировать его еще до начала реального события. Ключевым фактором в предварительном представлении оценки является то обстоятельство, что определенное мнение о длительности может быть сообщено пользователям заблаговременно. Если мнение отрицательное, люди, возможно, будут в процессе деятельности испытывать сомнение, будучи, в сущности, предупрежденными о том, что результат может не соответствовать их ожиданиям. Формирование восприятия

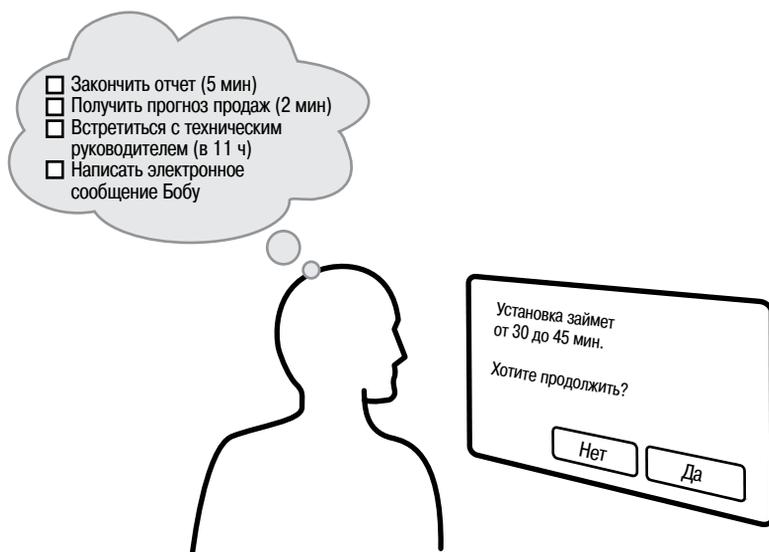
или воздействие на него может зависеть от непредсказуемого набора факторов. В сфере розничной торговли и обслуживания общеизвестно, что, когда люди решают не покупать определенное решение, не брать его в пользование и даже на пробу, они опираются на связанные с решением ассоциации, репутацию, торговую марку и даже цену (необязательно слишком дорого, но и слишком дешево!). Обратная сторона медали состоит в том, что люди согласны пройти через трудности, чтобы добыть желаемый продукт. Например, в 1996 году назойливая торговая реклама и спрос на новые в то время детские игрушки Tickle-Me-Elmo так сильно взвинтили покупателей рождественских подарков, что некоторые из них, по слухам, платили 1500 долларов за 30-долларовую игрушку.

### **Сообщайте время заблаговременно, если**

- Пользователям необходимо решить, «готовы» ли они запустить процесс.
- Пользователи, вероятно, захотят уделить внимание другим задачам.
- Процесс очень длительный и связывает пользователя.

Имейте в виду, что большинство пользователей не посвящают целиком свое время и безраздельное внимание работе с вашим решением. Скорее, у них будет запущено несколько приложений, не говоря уже о других возникающих делах, не связанных с компьютером (телефонный разговор с заказчиком, просмотр телепередач и т. д.). Пользователи, как и компьютеры, во многом многозадачны. Часто для одновременного выполнения нескольких задач им требуется гимнастика ума, повышающая продуктивность и профессиональные навыки: «Пока загружается файл, дайте мне возможность просматривать электронную почту, присланную торговой компанией, так чтобы мы оба могли видеть ее во время телефонного разговора».

Любой процесс, который ущемляет или подвергает риску многозадачные возможности пользователя, отрицательно влияет на потоковое состояние пользователя и его отношение к продукту или службе. Представим себе, что загрузка файла требует коммутированного соединения с помощью линии подключения телефона пользователя или что программа электронной почты так снижает пропускную способность сети, что загрузка файла приостанавливается или существенно замедляется. Очевидно, что следует дать пользователям оценку длительности, с тем чтобы они могли решить, готовы ли к запуску процесса или установлению контакта (рис. 7.1). Примеры включают те случаи, когда процесс потребляет очень много времени или связывает пользователя, – это означает, что процесс принудительно захватывает некоторый уровень приложения, операционной системы или вычислительную систему в целом, так что пользователи вынуждены ожидать возможности продолжения работы.



*Рис. 7.1. Если оценки времени сообщаются пользователям заблаговременно, последние могут решить, готовы ли они работать с процессом без ущерба для своего потокового состояния и понижения уровня других приоритетов*

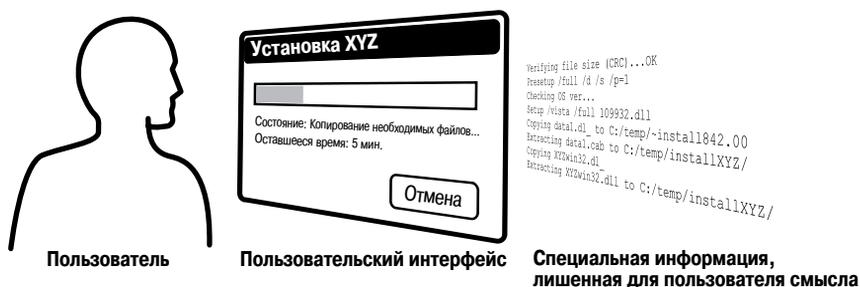
## В режиме реального времени: пробный стикер

При наблюдении заметных изменений в непрерывном режиме объем доступной информации соответствует проделанной работе. В мире индустрии розничной торговли и обслуживания уже много лет назад стало известно, что дать людям опытным путем ознакомиться с продуктом или услугой (посмотреть, потрогать, попробовать, понюхать или послушать) – это один из самых сильных маркетинговых ходов, потому что восприятие, сформированное на основании непосредственного опыта, является более устойчивым, чем составленное опосредованно, через коммерческую рекламу и объявления. Благодаря этому появился мир бесплатных образцов, роликов анонсов кинофильмов, сообществ испытателей, компьютерных программ с 30-дневным сроком пробного использования, пробных поездок на автомобилях, макетов домов и пробных стикеров. Несмотря на то что это работает еще до обретения реального опыта, потребителям прививается вкус, точнее говоря, дается обещание полноценного опыта. Имеется и обратная сторона обладания информацией в режиме реального времени. Если реальный опыт не является тем, что было обещано и ожидалось, информированные потребители будут стремиться избегать его. Благодаря этому появился мир возмещения расходов, гарантий возврата денег и кредитных историй!

## Сообщайте время в режиме реального времени, если

- Происходящие в настоящее время процессы и операции кажутся пользователю слишком специальными или информация о них лишена для него смысла.
- Пользователям необходимо знать значение затраченного времени и работать с ним.

Представление пользователям информации о происходящих в текущий момент процессах в режиме реального времени было отличительной особенностью индикации выполнения классов А и С, описанных в главе 6 «Индикация выполнения». К примеру, представление оставшегося времени обычно обеспечивает уверенность в том, что работа продвинулась, и постепенно информирует пользователей о том, что время ожидания сокращается. Если подробности о том, что выполняется (в единицах работы), кажутся пользователю слишком специальными или лишены для него смысла, лучше отображать оставшееся время, чем оставшуюся работу. Например, подробности о нескольких файлах, копируемых из одной папки в другую, могут иметь смысл для основного контингента пользователей, а детальные сведения о нескольких сотнях файлов, копируемых с установочного DVD-диска во временную папку, – нет. Следовательно, в первом случае демонстрация сведений о копируемых файлах может быть приемлемой, а во втором случае предпочтительно отображение значения оставшегося времени. На рис. 7.2 показан другой пример того, как в UI скрыта информация, не несущая смысла, и отображаются только сведения, поддающиеся интерпретации.



**Рис. 7.2.** Если подробная информация о процессе кажется пользователю слишком специальной или лишена для него смысла, сообщайте ему оценку оставшегося времени в режиме реального времени

Другой способ представления информации в режиме реального времени использует значение затраченного времени, однако повторим, что здесь следует соблюдать осторожность. Практическое правило состоит в том, что затраченное время пригодно только тогда, когда для пользователя оно действительно содержательно, например при диагностике

и оценивании производительности. Отметим, что отсутствие отображения затраченного времени не означает отсутствия слежения за ним. Побуждение пользователей к действиям, когда ожидания становятся длительными, – пример полезности отслеживания затраченного времени.

## Ретроспективно: всегда худший случай

Для людей является естественным оценивать продукт или сервис по результатам опыта его использования (как это делает персонаж комикса «Симпсоны», когда произносит характерную фразу «Вечно все плохо!»). К счастью, исследование показало, что можно свести на нет вредный эффект длительных ожиданий с помощью весомых аргументов или весьма желательных итогов, в частности отличного обслуживания. Очень важно понимание того, как люди рассматривают прошлый опыт, потому что высока вероятность, что они будут повторять его, если сочтут удачным. Повторение или возврат к тому, что считается стоящим, – это основной принцип, называемый законом влияния. Помимо возврата или повторения опыта, имеет место то, что люди начинают рекомендовать продукт, если опыт был удачным (то есть становятся так называемыми «глашатаями рынка» (word-of-mouth marketing, WOMM)), или резко критиковать его в случае негативного опыта. В наш век блогов особенно важно дать людям такой опыт, чтобы они, оглядываясь назад, оценивали его позитивно.

### Сообщайте время ретроспективно, если

- Знание того, какое время занял процесс, является содержательным или значимым для пользователя.
- Необходимо произвести диагностические измерения или оценить производительность.

Ретроспективное сообщение затраченного времени (после завершения процесса) должно осуществляться только в тех случаях, когда информация о том, какое время занял процесс, является содержательной или значимой для пользователя, например во время проведения диагностических измерений или оценивания производительности (рис. 7.3). Для большинства приложений в представлении пользователям затраченного времени большого смысла нет. Если есть какое-то сомнение, заполните анкету: «Мои пользователи будут использовать отчет о затраченном времени для \_\_\_\_\_». Все, кроме фразы «для того, чтобы оценить скорость выполнения процесса», будет характеризовать ваших пользователей, их потребности и то, как они пользуются вашим решением. Если вы не можете заполнить анкету, сообщение затраченного времени может оказаться вредным для вашего решения.



*Рис. 7.3. Сообщайте значение затраченного времени, только если пользователям важно видеть, какое время потребовалось, чтобы полностью выполнить процесс. На схеме показан пример, где администратор базы данных исследует вывод программы текущего обслуживания, который информирует о том, была ли допустимой ее производительность*

## Разговор о времени

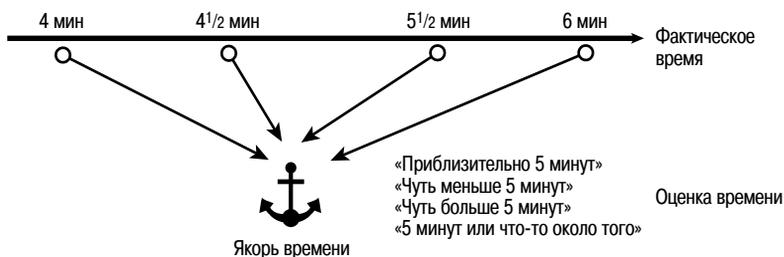
Некоторым людям строго рекомендовано ежедневно выпивать восемь стаканов воды. Смысл здесь в том, что это полезно для здоровья, в частности предотвращает обезвоживание организма. Эта рекомендация проста до тех пор, пока кто-то не начнет выяснять подробности о такой мере объема, как стакан. Разумеется, использование стакана (а не унции или миллилитра) в качестве меры объема объясняется стремлением к простоте и практичности рекомендации. В большинстве случаев, исключая ситуации, где точность жизненно важна, например в нейрохирургии или в аэрокосмической технике, мы в своей жизни не пользуемся точными измерениями и не нуждаемся в них, а довольствуемся приблизительными оценками. Восприятие часто неточно, и для большинства видов деятельности в жизни и в работе иное не требуется. В повседневной деятельности при оценивании и выражении времени мы часто опираемся на хорошо знакомые объекты: мяч для гольфа имеет такой же размер, как котлета. В следующем месяце поступит в продажу новая цифровая камера размером с кредитную карту. Так обычно говорят при оценивании объема, массы, расстояния и т. д.

Оценивая время, мы предпочитаем использовать действительные единицы времени (секунды, минуты, часы), а не ссылки на объекты или события. Например, немногие (по крайней мере, в западной культуре) скажут, что ожидание обслуживания в ресторане составило от четырех- до пятикратной длительности, требуемой для варки яйца вкрутую, или что сервировка стола заняла столько, сколько длится 50-кратное произнесение «Аве Мария». Примером случая, когда мы не пользуемся действительными единицами времени, является сравнение одной длительности с другой. Например, «За то время, пока мы

ожидали еду, мы могли уже закончить обед в другом ресторане». Тем не менее мы склонны упрощать оценки времени так же, как делаем это с оценками других величин.

## Якоря времени

Вспомните вашу последнюю серьезную беседу с кем-нибудь. Какова была ее длительность? Если вам придется использовать единицы времени, например секунды или минуты, высока вероятность того, что для описания длительности вы назовете целое число, такое как 1, 2, 5 или 10. Когда нас просят охарактеризовать длительность обыкновенного события, мы редко даем точные оценки, к примеру 10,7 с или 5,17 мин, если только специально не проводили хронометраж с помощью секундомера или наручных часов. При оценивании времени мы тяготеем к особым числам. Я называю их *якорями времени*, потому что люди стремятся прикрепить свои оценки к одному или более из этих чисел (рис. 7.4). Термин «якорь» используется, чтобы подчеркнуть следующий факт: несмотря на наше знание того, что событие длилось менее или более пяти минут, при выражении оценки мы все-таки тяготеем к пяти.



**Рис. 7.4.** Несмотря на то что обычный человек может обнаружить различие между двумя длительностями, существует тенденция использовать якоря времени при выработке оценок

Для того чтобы проиллюстрировать роль якорей времени, рассмотрим следующие предложения:

- Беседа с моим менеджером заняла не более пяти минут
- Он вышел на сцену и застыл примерно на 30 секунд
- Она опоздала на встречу больше чем на десять минут

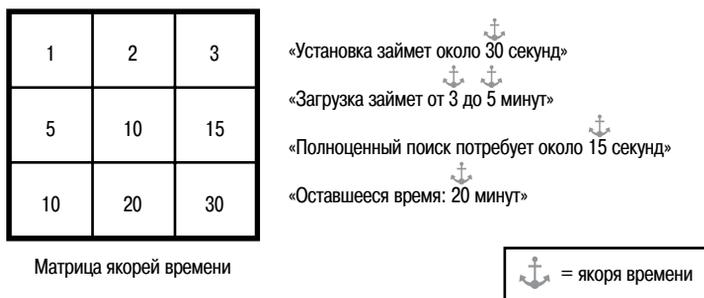
Теперь посмотрим, насколько странно выглядели бы эти предложения без использования якорей времени:

- Беседа с моим менеджером заняла не более 4,3 минуты
- Он вышел на сцену и застыл примерно на 34 секунды
- Она опоздала на встречу более чем на 9,44 минуты

Использование точных значений (в обстоятельствах обычных, случайных, свободных от негативных оценок или в обиходной речи) оставляет впечатление, что приведенные оценки времени являются намеренно точными, и люди либо просто предполагают, что вы хронометрировали событие, либо проявляют интерес к причине точности приведенных оценок. Если вы знаете, что ваша беседа с менеджером длилась 4,3 мин, то, возможно, вы были невнимательны, не так ли?

## Матрица якорей времени

В оценках длительностей, не превышающих нескольких часов, люди тяготеют к числам 1, 2, 3, 5, 10, 15, 20 и 30. Это значит, что, когда требуется оценить малую длительность, люди склонны использовать в своих оценках одно или более из указанных чисел, например «примерно десять секунд» или «от двух до трех минут». Это наблюдается и в отношении длительностей, составляющих несколько часов, но в меньшей степени. Для быстрого запоминания этих чисел служит объект, названный мною *матрицей якорей времени* (рис. 7.5).



*Рис. 7.5. Числа, к которым люди тяготеют при выражении времени, легко запомнить при помощи матрицы якорей времени*

Обычный человек может определить разницу между четырьмя и восемью минутами, так что матрица не подразумевает, что мы можем оценивать время, используя лишь эти целые числа, или что мы воспринимаем все в мире в порциях времени, привязанных к этим числам. Наоборот, здесь речь идет о легко воспринимаемом и практичном наборе чисел, который удобен для нас и которым мы уверенно пользуемся для описания длины интервалов времени, в особенности без подготовки и в обыденных обстоятельствах.

Почему люди тяготеют к этим числам, неясно, но вполне просматривается отчетливое влияние нашей шестидесятеричной (с основанием 60) часовой системы. Другими очевидными факторами влияния являются язык и культура. В некоторых частях света количественное выражение времени в секундах и минутах не является распространенным. В некоторых странах Ислама время обычно выражается по отношению к пятидневному мусульманским богослужениям. В Израиле время часто указывается относительно ежечасных передач новостей.

## Разговор о времени

Подобно тому как люди могут общаться с чужестранцами языком музыкальных нот в кинофильме Стивена Спилберга «Близкие контакты третьего рода» или на математических языках в позднем романе Карла Сагана «Контакт», мы можем применять матрицу якорей времени для того, чтобы сообщать пользователю оценки времени. Три основные разновидности оценок времени – это диапазоны, границы и обратный отсчет.

### Диапазоны: между X и Y

Якоря времени удобны, когда необходимо указать временной диапазон для описания возможной длительности события. Это часто бывает, если изменчивость длительности обусловлена влиянием одного или более различных факторов. Например, если мы полагаем, что конкретный процесс займет приблизительно четыре минуты, то можем заявить (отобразить в UI) диапазон, охватывающий четыре минуты. Обратившись к матрице якорей времени, мы замечаем, что 3 и 5 – это ближайшие целые числа, окружающие наше число, и поэтому мы заявляем, что процесс займет от трех до пяти минут (рис. 7.6).

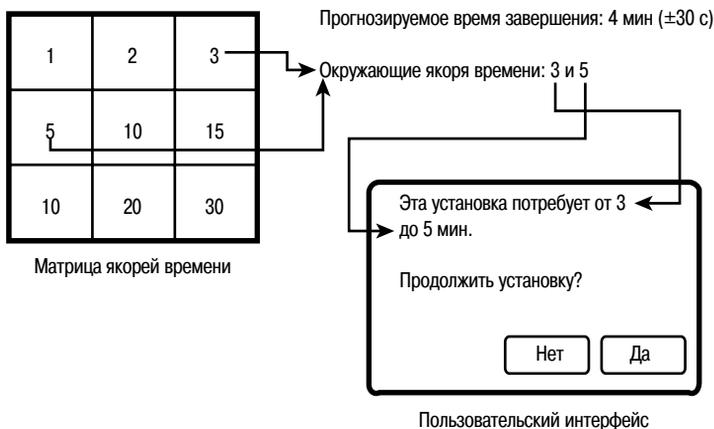


Рис. 7.6. Пример использования матрицы якорей времени для выражения диапазона длительностей в UI

Практическое правило указания диапазонов заключается в том, что не надо пропускать ни один из последовательно встречающихся якорей времени. Например, мы не должны заявлять диапазон от 5 до 15 минут, пропуская 10. Широкие диапазоны не следует использовать, и причина станет ясной, если мы немного пофантазируем. Сравните вашу реакцию на предложения двух гипотетических компаний кабельного телевидения, из которых первая обещает, что ее специалист по обслуживанию будет у вас дома между 10 часами утра и 12 часами ночи, а вторая говорит о приходе мастера между 10 часами утра и 3 часами

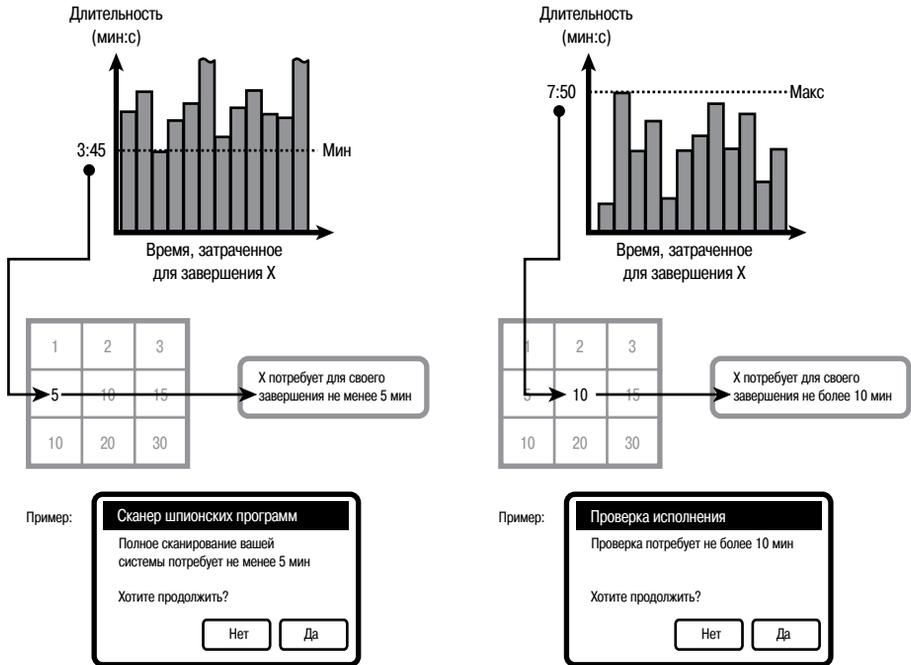
дня. Если разность между двумя якорями времени слишком велика, нам начинает казаться, что диапазон мог бы быть и поменьше. Причина того, что большой диапазон может вызвать раздражение, возможна, связана с классическим принципом психологии под названием «закон Вебера–Фехнера», упомянутым в главе 5 «Обнаружение различий в распределении времени». Отметая подробности, можно сказать, что, вероятно, в этом случае трудно «вставить» один или более якорей времени в промежуток между наибольшим и наименьшим крайними значениями диапазона.

### Границы: меньше или больше, чем $X$

Есть два способа выражения временных границ, и они сильно различаются по своему назначению. Нижние границы несут в себе информацию о том, что конкретное событие будет длиться по меньшей мере  $X$  единиц времени. Верхние границы говорят пользователям о том, что конкретное событие будет длиться не более  $X$  единиц времени. Нижние границы следует применять осмотрительно, потому что, по существу, это предупреждение о неотвратимом ожидании и такие объявления обычно делаются для того, чтобы человек набрался терпения: *Дорога в Ванкувер займет не менее трех часов. Багаж окажется в Сингапуре не ранее чем через неделю. Осмотр дома займет больше двух часов.* Верхние границы, напротив, гарантируют завершение: *Вы будете в Ванкувере не позже полуночи. Вы получите багаж в течение месяца. Осмотр дома будет выполнен в течение одного часа.* Нижние границы предупреждают, а верхние обнадеживают.

Если установить диапазон невозможно или сложно, используйте верхнюю границу для объявления наибольшей из возможных длительностей, например, так: «меньше пяти минут». При установлении верхней границы производите округление сверху до следующего элемента матрицы якорей времени. Например, если мы уверены, что некий процесс закончится не более чем через 7 мин 50 с, мы скажем, что процесс займет около 10 минут. Используйте нижнюю границу, когда необходимо заранее предупредить пользователей, что процесс будет длительным, особенно если процесс связывает пользователей так, что они не могут взаимодействовать с другими приложениями, операционной системой или вообще с машиной до завершения процесса. Это те случаи, когда у вас есть некоторая уверенность в том, что длительность процесса не будет меньше некоторой, вам известной. Устанавливайте при этом нижнюю границу, также производя округление сверху до следующего большего якоря времени. Например, если процесс будет продолжаться по меньшей мере 3 мин 45 с, сообщите пользователям, что процесс займет не менее 5 мин (рис. 7.7). Обещайте меньше и давайте больше.

Аргументом против использования конкретных и точных чисел, таких как 7 мин 50 с или даже 8 мин, является то, что, когда мы используем числа, которые пользователи употребляют не привыкли (то есть не



**Рис. 7.7.** Две иллюстрации того, как используются нижние (слева) и верхние границы (справа). Нижние границы предупреждают пользователя о неизбежном ожидании, что помогает ему принять решение о том, стоит ли запускать процесс (к примеру, сканирование на предмет шпионских программ). Верхние границы гарантируют, что процесс завершится за определенное время, и это действует как хороший стимул для пользователей к запуску процесса

представленные в матрице якорей времени), мы рискуем тем, что даем сомнительное обещание, так как прогнозируемая длительность на самом деле носит вероятностный характер. Пользователь же приобретает твердую надежду, поскольку ваше заявление выглядит как подкрепленное скрупулезным хронометражем и контролем производительности и создается впечатление, что вы каким-то образом точно зафиксировали восемь минут. Напротив, если вы используете следующее большее число из матрицы, то есть десять минут, это, скорее всего, воспримется как грубая оценка благодаря широко распространенному разговорному словосочетанию «десять минут».

### Оставшееся время: Z, Y, X...

До тех пор пока затраченное время сообщать необязательно (0:01... 0:02... 0:03...), всегда лучше использовать оставшееся время или таймер с обратным отсчетом времени (0:54... 0:53... 0:52...), если, вообще, есть нужда в отображении таймера. Оставьте подобное использование

таймеров для относительно малых длительностей, составляющих не более десяти минут. Представьте себе значения таймера с обратным отсчетом времени, начинающимся от значения одного часа: 1:00:00... 0:59:59... 0:59:58... 0:59:57... Очевидно, следить за этим будет не очень приятно!

Из-за того что значения таймера затраченного времени убывают, как на хронометре или секундомере, сопровождение таким таймером длительного процесса равносильно приглашению пользователей к слежению за временем, что нервирует. Несмотря на то что использование таймеров оставшегося времени обычно более предпочтительнее, все же, если обратный отсчет «тикает» каждую секунду, это может оказать столь же вредное воздействие, как использование таймера затраченного времени. Здесь на помощь приходят якоря времени. Например, вместо индикации каждой секунды между 10 мин и 0 с (10:00... 9:59... 9:59... 9:58...) показывайте обратный отсчет в единицах якорей времени: (10 мин... 5 мин... 3 мин... 2 мин... 1 мин... 30 с...).

Отметим, что при представлении оставшегося времени никогда нельзя допускать его роста. Самое большее, на что можно пойти, – это сохранение неизменного значения на каком-то временном отрезке. Оно никогда не должно увеличиваться или, что еще хуже, флуктуировать. Если оставшееся время склонно к флуктуациям, что затрудняет его прогнозирование, значит, в данном случае его индикация – не лучший способ демонстрации хода процесса.

## Ситуации неопределенности

Иногда установка диапазонов или верхних границ невозможна из-за того, что оцениваемое время зависит от факторов, слишком изменчивых или трудно прогнозируемых. Одним из печально известных примеров является доступ к ресурсу, такому как база данных, по сети. Скорость передачи по сети, возможно, и определена, но разнообразные факторы могут препятствовать доступу к базе данных. Дилемма, состоящая в отсутствии возможности прогнозировать длительность и необходимости представить какую-то ее оценку, обычно влечет появление таких фраз, как «несколько секунд», «подождите, пожалуйста» или «через некоторое время». Такая практика свидетельствует о неудачном подходе, потому что эти фразы очень неопределенны и не дают пользователям ощущения комфорта при слежении за ходом процесса.

Индикация времени с помощью фраз «несколько секунд» или «некоторое время» может оказаться проблематичной, потому что подобные слова воспринимаются субъективно и в сильной зависимости от контекста, не говоря уже о культуре, языке, возрасте и т. д. Поэтому маловероятно, что такие понятия будут наделяться количественным значением и восприниматься одинаково, например, всеми потребителями, взаимодействующими с поставщиком услуг, или всеми абонентами,

которые прослушивают звукозапись в телефоне, обещающую, что кто-то «очень скоро» с ними поговорит. Тоска, испытываемая, когда нас ставят в режим ожидания телефонного разговора и продолжительность ожидания неизвестна, в действительности является следствием *неопределенности*, и хорошим средством от нее служит внесение какой-то определенности.

Например, в некоторых компаниях отделы обслуживания потребителей взяли за правило содержать персонал, отвечающий на телефонные звонки по мере их поступления, и теперь позвонившему сразу сообщают, что он будет поставлен в режим ожидания: «Компания ABC, подождите, пожалуйста». Такой подход не лишен целесообразности, потому что пользователь получает какую-то *определенность*, хотя бы в отношении того, что он набрал правильный номер и, по крайней мере, добрался до живого человека на линии. Аналогично, пользователь выигрывает, получая определенность в отношении *любой* выполняющейся работы, даже если оценить время невозможно. Далее приведем несколько рекомендаций и показательных примеров того, как избежать применения неопределенных понятий.

## 1. Предоставляйте информацию, не относящуюся ко времени

Первая рекомендация заключается в применении индикации выполнения. Если время завершения процесса не прогнозируется и пользователю недоступен статус процесса, мы, по существу, сталкиваемся со сценарием класса В (рис. 6.2), или с индикацией «занято/в работе». Следовательно, мы стремимся выполнить упражнение, показанное на рис. 6.5, для перемещения ситуации в левый нижний квадрант, чтобы можно было реализовать индикацию выполнения класса С. Иными словами, мы стремимся отыскать всю возможную информацию и отобразить всю содержательную для пользователя ее часть (например, количество прочитанных строк или найденных программ, информацию об обновлении очереди, этап компиляции и т. д.), чтобы показать, что процесс выполняется.

## 2. Таймеры и тайм-аут

Несмотря на то что затраченное время, как правило, в UI не фигурирует, его можно использовать в скрытом виде для реализации механизма отклика на необычно большие задержки. С этим мы сталкиваемся во многих телекоммуникационных аппаратах, например в мобильных телефонах, когда аппарат перестает подавать звонки после их фиксированного числа или по истечении заданного интервала времени. В мире программного обеспечения мы обычно обнаруживаем такой механизм тайм-аутов в веб-браузерах или других решениях, связанных

с сетями. Ключевым вопросом в такой реализации является определение того, какие действия должны быть предприняты и в какое время. Первым шагом должно быть использование информации или моделей для выяснения распределения времени ожидания и связывания его с показателями успеха и неудачи. Например, вы можете прийти к выводу, что при отсутствии активности свыше 30 с показатель неудачи составляет примерно 90%. В этом случае разумным будет предоставление пользователю средств для продолжения, отмены или перезапуска процесса, если задержка превысила 30 с. При этом существенно, что при превышении десяти секунд (класс вынуждающей реактивности из главы 4 «Реактивность») настоятельно рекомендуется предоставлять «аварийный выход», к примеру, в виде кнопок Отмена или Повторить.

## Грамматика и этикет представления времени

Представляя время в рамках UI, вы должны соблюдать несколько простых правил. В следующих главах вы найдете дополнительные методы и сведения о нарушениях, здесь мы представляем вашему вниманию лишь пять рекомендаций.

### 1. Употребляйте единственное число для единичных значений

Всегда приветствуется написание нескольких дополнительных строк кода для обеспечения уверенности в том, что названия единиц времени употребляются в единственном числе, если это необходимо. Например, необходимо писать не «1 минут», а «1 минута», не «1 секунд», а «1 секунда». Если требуется отобразить количество единиц времени, меньшее единицы, используйте множественное число, например «0 секунд» или «0,5 минут». (Лучше воспользоваться словами «завершено» или «выполнено», чем написать «0 секунд».) Употребление единственного числа для единиц измерения должно использоваться только для цифры 1 или числа, заканчивающегося этой цифрой, не больше и не меньше.

### 2. Ноль означает закончено!

Отображение оставшегося времени «0 секунд», означающее, что процесс завершен, не относится к другим исполняемым процессам. Однако иногда другой процесс «врывается» в вычислительную среду после завершения первого, например, это может быть распаковка загруженного файла. Тогда перед глазами пользователя предстает картина бесконечно выполняющегося, никогда не завершающегося процесса. Делу может помочь информирование пользователей о начале выполнения нового процесса или же включение времени, требуемого дополнительными процессами, в оценку оставшегося времени.

### 3. Пользуйтесь единицами времени единообразно

Для отображения времени принято использовать цифры, например «Эта установка займет от 1 до 2 мин». Можно то же выразить словами, например «от одной до двух минут», но никогда не устраивайте мешанину («от одной до 2 минут»). Двухзначные числа в выражении времени лучше отображать цифрами, например «15 мин», а не «пятнадцать минут».

### 4. Между X и Y

Когда употребляется предлог *между*, убедитесь, что по соседству присутствует союз *и*, например «Время этой установки лежит между тремя *и* пятью минутами», а не «Время этой установки лежит между тремя *до* пяти минут». Использование *до* во фразе «Эта установка займет от одной *до* двух минут» верно.

### 5. Избегайте двусмысленных фраз

Использование словосочетаний «*очень скоро*» или «*через некоторое время*», скорее всего, приведет пользователя в растерянность. Не используйте эти неопределенные понятия, даже если процесс характеризуется непредсказуемым временем завершения. Обратитесь к главе 6 и выберите верный класс индикации выполнения. Словом «*секунда*» в предложении «Мы ответим вам через одну секунду» злоупотребляют, оно не воспринимается буквально. Неформальный опрос показал, что со словосочетанием «одна секунда» обычно ассоциируется ожидание примерно шесть секунд, в то время как «одна минута» и «один час» чаще всего понимаются буквально. Прилагательные и наречия, например «*незамедлительно*» и «*мгновенно*», тоже являются неопределенными.

## Выводы

В представлении времени в рамках UI важно, *когда* представляется информация о времени и *как* она выражается. На восприятие, поведение пользователя и пользовательское отношение может существенно повлиять то, представляется ли информация предварительно, в режиме реального времени или ретроспективно. В этой главе приводятся некоторые правила выражения времени. Для численного выражения времени пользуйтесь якорями времени при обозначении диапазонов, границ и оставшегося времени – тогда пользователь не воспримет оценки как точные. Кроме того, здесь разъясняются некоторые грамматические правила и нормы, относящиеся к выражению времени, например правило единственного числа для единичных значений.

## Кроличья нора

### Как представлять время: предварительно или ретроспективно

Teigen, K. H. and K. I. Karevold. Looking back versus looking ahead: Framing of time and work at different stages of a project. *Journal of Behavioral Decision Making*, 18, 2005. 229–246.

### Заниженная и завышенная оценки при предварительном представлении времени

Roy, M. M., N. J. S. Christenfeld and C. R. M. McKenzie. Underestimating the duration of future events memory: Incorrectly used or memory bias. *Psychological Bulletin*, 131, 2005. 738–756.

Zauberman, G. and J. G. Lynch. Resource slack and propensity to discount delayed investments of time versus money. *Journal of Experimental Psychology: General*, 134, 2005. 23–37.

### Стили написания

The Chicago Manual of Style Online. (Примечание автора: Доступно в Сети по адресу [www.chicagomanualofstyle.org/indexT.html](http://www.chicagomanualofstyle.org/indexT.html). См. раздел «Time».)

### Якоря и оценивание времени

Konig, C. J. Anchors distort estimates of expected duration. *Psychological Reports*, 96, 2005. 253–256.

# Глава 8

## Потоковое состояние пользователя

Понятие «потоковое состояние пользователя» (user flow) часто позволяет удачно описать то, насколько гладким и приятным представляется ваше решение пользователям. Потоковое состояние, отличное от оптимального, может ощущаться пользователями как бесконечный путь, состоящий из мучительных шагов. Напротив, оптимальное потоковое состояние пользователя может содействовать успеху при выполнении задач с большим потреблением времени благодаря тому, что пользователи не будут замечать хода времени. Даже будучи осведомленными о значительной длительности процесса, они могут, как ни странно, радоваться каждому его моменту. В этой главе обсуждается, как обеспечить оптимальное потоковое состояние пользователя, с учетом исследований в области психологии потока.

### Что такое потоковое состояние пользователя

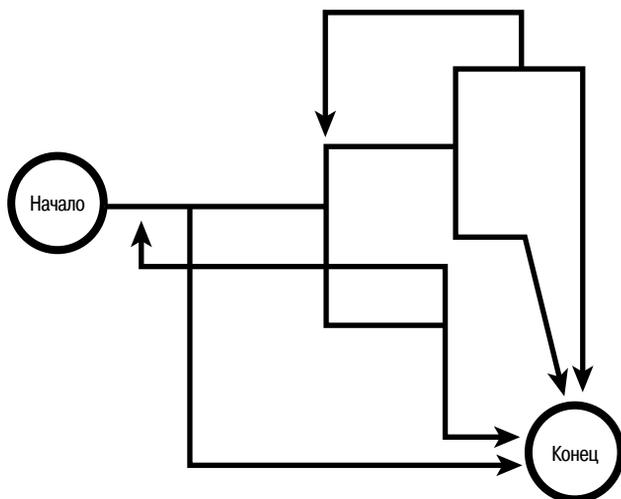
В главе 2 «Восприятие и толерантность» мы говорили о том, что воспринимаемое время может быть оценено с занижением или с завышением. Коротко говоря, недостатки проектирования могут привести к тому, что малые длительности будут казаться длиннее, чем они есть на самом деле, и наоборот, применив тщательно подобранный метод, можно добиться того, что большие длительности будут восприниматься короче. Иными словами, из-за неудачного проектирования малые длительности могут восприниматься как невыносимые, в то время как оптимальное проектирование обеспечивает терпимое отношение к большим длительностям. Степень различия восприятия и толерантности

разных людей частично определяется тем, каково их потокное состояние пользователя в ходе непрерывного, «момент-за-моментом», взаимодействия с вашим решением (будь то программное приложение, продукт или услуга). Простое взаимодействие, например программирование цифрового видеомэгнитофона на запись телепередачи, может быть гладким для одного человека, собирающегося записать телепередачу, которая начинается через час, и не казаться таковым другому, желающему записать телепередачу, которая начинается через минуту. Все это отражается на удовлетворении пользователя. Неудовлетворительное потокное состояние пользователя порождает у него ощущение непродуктивности, неэффективности, беспокойства и т. д. Оптимальное потокное состояние вызывает у пользователя ощущение своей повышенной работоспособности, уверенности и креативности.

Мы можем и без всяких критериев определить, насколько гладким и приятным является то или иное решение в отношении потокного состояния пользователя, однако в контексте человеко-машинного взаимодействия потокное состояние пользователя можно охарактеризовать по некоторым обоснованным критериям. Первое возможное определение, по существу, заключается в рассмотрении вашего решения с точки зрения ощущения того, куда пользователь может «двигаться» при взаимодействии с вашим продуктом. Потокное состояние пользователя может также характеризоваться реальной последовательностью шагов, или траекторией, по которой пользователь проходит при выполнении конкретно поставленной задачи. Третье возможное определение базируется на текущем впечатлении пользователей о прохождении траектории при выполнении задачи. Рассмотрим все три аспекта подробнее.

## Потокное состояние пользователя в виде карты

Структурная схема (или схема другого вида) иногда оказывается полезной при построении картины возможных маршрутов пользователей при работе с вашим решением. Такой вид наглядного пособия может представлять карту потокного состояния пользователя. Эта визуализация дает проектировщикам, инженерам и разработчикам структуры системы представление о возможных траекториях, области действия и сложности проекта системы и часто может помочь в определении недостатков, зависимостей или пробелов в архитектуре вашего решения (рис. 8.1). Ее не следует путать, к примеру, со схемой структурных подразделений или схемой потока данных в системе в последовательности их обработки. Подобные виды схем обычно отображают технологический процесс, технологический маршрут или поток данных. Они ориентированы на процессы, в то время как карта потокного состояния пользователя ориентирована на человека. Это означает, что в центре внимания оказывается не процесс, а пользователь. К числу общепринятых понятий, используемых для описания таких карт потокного

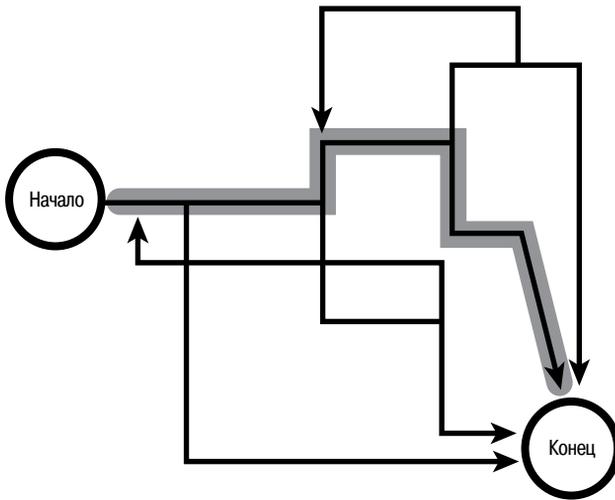


*Рис. 8.1. Карта потокового состояния пользователя дает представление о возможных маршрутах, которыми может пройти пользователь, работая с вашим решением*

состояния пользователя, относятся *поток пользовательского интерфейса, пользовательская навигация, поток заданий, поток взаимодействия и поток деятельности.*

## Потоковое состояние пользователя в виде траектории

Такая карта потокового состояния пользователя, которая показана на рис. 8.1, сравнительно проста и может быть реалистичной только в описаниях простых систем и решений, например автоматизированной сети поддержки пользователей или банкомата, где пользовательское взаимодействие ограничено. Более вероятно, что ваше решение будет содержать большее количество вариантов выбора, процессов, шагов и зависимостей. В любом случае иногда вы должны будете продемонстрировать или протестировать потоковое состояние пользователя в конкретном варианте использования. Предположим, вы проектируете систему помощи студентам колледжа при записи на курсы. Ваша система должна проверить, имеется ли возможность получения стипендии, предъявляет ли курс какие-то предварительные требования и т. д. Далее предположим, что в 85% случаев студенты не смогут записаться на курс. Вы можете захотеть найти этот особый сценарий, чтобы исследовать в нем траекторию потокового состояния пользователя (рис. 8.2). В свете рассматриваемого определения потоковое состояние пользователя описывает шаги, контрольные точки и решения, принимаемые пользователем, вдоль следования по особой траектории. Применение такого подхода обычно относится к тому раннему этапу жизненного цикла продукта, который связан со спецификацией и проектированием, – для

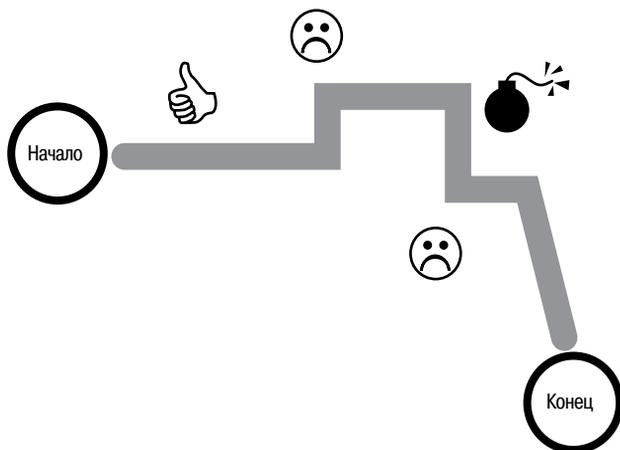


*Рис. 8.2. Потокное состояние пользователя, представленное в виде траектории, описывает шаги, контрольные точки и решения, принимаемые пользователем, вдоль следования по определенной траектории вашего решения*

определения пользовательского отношения – и к тому позднему этапу, который связан с тестированием. Родственные методы обеспечения удобства использования, также основанные на понятии потокного состояния пользователя, включают когнитивный контроль, использование примеров, сценарии, сценарии применения и работу с архивными документами.

## **Потоковое состояние пользователя как описание его впечатления**

Третий способ определения потокного состояния пользователя, основной в содержимом этой главы, фокусирует внимание не на траектории, а на впечатлении пользователя о «путешествии» (рис. 8.3). Траекторию потокного состояния пользователя вполне можно сравнить с путевым дневником путешественника – она говорит вам о его остановках, поворотах, времени путешествия и т. д. Пересказ впечатлений о путешествии, однако, может содержать больше личного – длительные процедуры регистрации, плачущий ребенок в полете, ужасная еда и т. д. Вообще говоря, нет ничего плохого в том, чтобы документировать траекторию потокного состояния пользователя с целью воспользоваться результатами при проектировании или усовершенствовании системы, определении пробелов или модернизации процессов. Что траектории не учитывают – так это качество каждого взаимодействия и то, что прошло через сознание пользователя при его продвижении вдоль траектории. В то время как траектория пользователя концентрирует внимание и характеризует путешествие в целом, от А до Z,



*Рис. 8.3. Потоковое состояние пользователя в виде его впечатления передает качество конкретной траектории*

потоковое состояние пользователя концентрирует внимание на пользователе и непрерывно, «момент-за-моментом», отслеживает качество в продолжение путешествия.

## Оптимизация потокового состояния пользователя

Для того чтобы вы могли понять и оценить идею потокового состояния пользователя, вначале следует ознакомиться с трудом Михая Чиксентмихайи (Mihaly Csikszentmihalyi)<sup>1</sup> о психологии потока. Профессор психологии определяет потоковое состояние как такое, в котором человек

*полностью вовлечен в деятельность ради нее самой. Эго исчезает. Время летит. Каждое действие, движение, мысль следуют из предыдущих, словно играешь джаз. Вовлечено все твое существо, и ты применяешь все умения и навыки, которыми обладаешь.*

Выражаясь предельно упрощенным образом, потоковое состояние подобно особому состоянию из практики дзен, где вы испытываете полное единение с деятельностью. Существует два важных различия между потоковым состоянием и исполнением работы. Во-первых, потоковое состояние не формирует впечатление о скорости выполнения задачи, потому что восприятие времени часто в этом состоянии нарушено.

<sup>1</sup> Михая Чиксентмихайи – профессор психологии, бывший декан факультета Чикагского университета, получивший широкую известность за свою идею потокового состояния, изучаемого им в течение нескольких десятилетий. – *Примеч. науч. ред.*

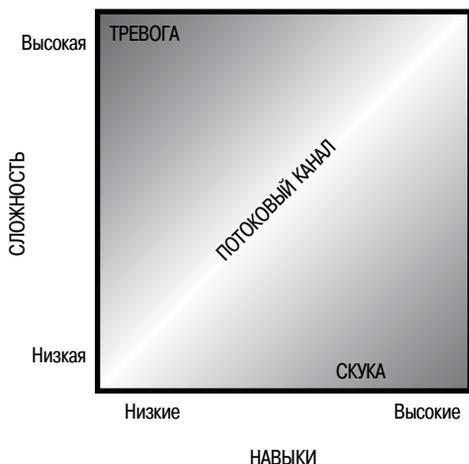
Во-вторых, потокное состояние не формирует представление о степени успеха; люди, достигшие потокного состояния, часто говорят о том, что Чиксентмихайи именуется опытом самодвижения к цели или «самодостаточной деятельностью, совершаемой без надежды на какую-то будущую выгоду, а просто потому, что деятельность сама по себе воспринимается как награда». При этом, даже если задача не выполнена или отсутствует успех, человек все же доволен и вознагражден.

Всеобъемлющий анализ концепции потокного состояния выходит за рамки этой книги, где ей посвящена лишь одна глава, но читателям, которых увлекла эта интереснейшая тема, обязательно следует ознакомиться с книгой М. Чиксентмихайи «Flow: The Psychology of Optimal Experience». Сейчас же достаточно будет сказать, что Чиксентмихайи определил условия существования потокного состояния. Некоторые из этих условий представляют собой внутренние психические состояния, например опыт самодвижения или нарушенное восприятие времени. Другие условия включают внешние факторы; предполагается, что они могут относиться к влиянию среды или других людей. Рассмотрим три условия существования потокного состояния, чтобы вы могли применить эти сведения непосредственно к своему решению.

## 1. Соответствие между умениями и навыками и сложностью задачи

Согласно модели Чиксентмихайи потокное состояние характеризуется оптимальным опытом, при котором сложность задачи соответствует умениям и навыкам человека. Чиксентмихайи поясняет, что «радость всегда приходит в особый момент: когда человек осознает, что предстоящая деятельность в полной мере соответствует его возможностям». Это означает, что, когда воспринимаемая сложность задачи соответствует умениям и навыкам человека, можно сказать, что человек пребывает в так называемом *потокном канале*. Если сложность задачи выше, чем умения и навыки человека, то он окажется в состоянии тревоги из-за того, что задача слишком трудна. И наоборот, когда умения и навыки человека превышают сложность задачи, он будет скучать по той причине, что задача слишком проста (рис. 8.4).

Важно учитывать фактор восприятия пользователем сложности применения вашего решения. Ключевую роль играет то, что мы говорим о воспринимаемой, а не о реальной сложности работы с решением. Например, банкомат должен восприниматься человеком, часто его использующим, как простое в работе устройство; в то же время ребенку, который никогда раньше им не пользовался, он кажется очень сложным аппаратом. Аналогично, программное обеспечение системы автоматизированного проектирования для человека, ранее ею не пользовавшегося, может выглядеть устрашающе и, следовательно, представляться сложным в использовании; однако этого нельзя сказать об архитекторе, работающем с данной системой ежедневно. В связи с этим

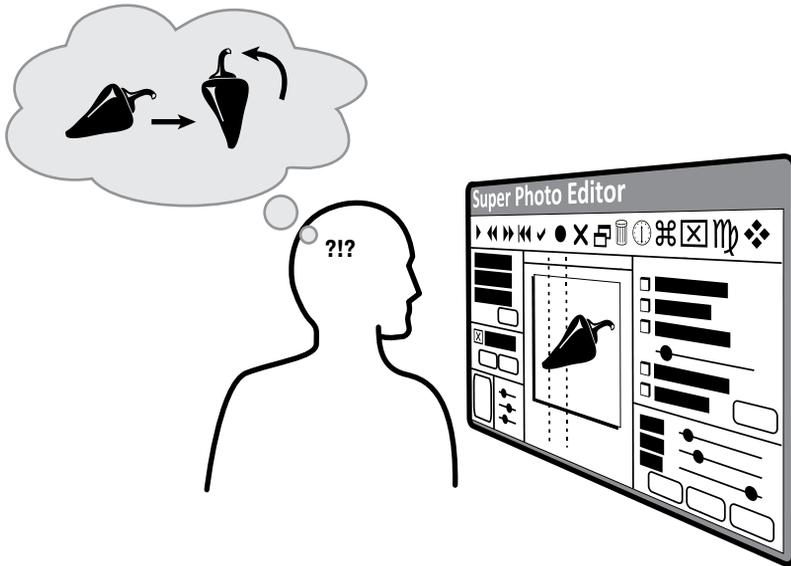


*Рис. 8.4. Согласно модели Чиксентмихайи, если воспринимаемая сложность задачи соответствует умениям и навыкам человека, он пребывает в потоковом канале. Если сложность задачи выше, чем умения и навыки человека, он оказывается в состоянии тревоги. Если умения и навыки человека превышают сложность задачи, он будет скучать по причине ее простоты*

не будем забывать, что восприятие сложности использования меняется со временем. Так, при тренировке воспринимаемая сложность снижается. Справедливо и то, что без адекватной тренированности/опыта или при постоянных неудачах в выполнении задачи ее воспринимаемая сложность может расти.

## Что вы можете сделать

**Подстройте сложность к умениям и навыкам вашего пользователя.** Некоторые пользователи могут быть новичками в использовании вашего решения, в то время как другие могли работать с более ранней версией или с другим подобным решением и оказаться способными применить прежние знания. Например, на рис. 8.5 показан новичок, который хочет повернуть изображение, но, к сожалению, UI приложения слишком сложен для него. В результате простая задача становится сложной. Согласно модели Чиксентмихайи здесь возникает тревога. Возможной рекомендацией в этом случае является встраивание в приложение механизма, который может либо адаптироваться, либо допускать подстройку к среде или режиму, так чтобы результатом было соответствие умениям и навыкам пользователя. Многие приложения при установке или запуске запрашивают пользователей об их уровне подготовки. Другие позволяют пользователям переключаться на другие режимы, например осуществлять переходы между вариантами интерфейса Основной и Расширенный.



*Рис. 8.5. Пользователь-новичок хочет всего-навсего повернуть изображение, но, к несчастью, UI приложения слишком сложен для него. Простая задача на деле обернулась сложной*

Чиксентмихайи пишет, что в высшей точке потокowego состояния люди, среди прочих внутренних ощущений, упоминают самозабвенность и измененное ощущение времени. Однако важно помнить, что не каждый пользователь достигнет подобного состояния из практики дзен, используя ваше решение. Клиенту банка не нужна самозабвенность при получении денег в банкомате. Курьер, вероятно, не захочет испытывать необычное ощущение времени, упаковывая свою ношу. Ключевую роль здесь играет понимание того, как воспринимаемая сложность соотносится с уровнем необходимых навыков пользователя, и соответствующая настройка сложности.

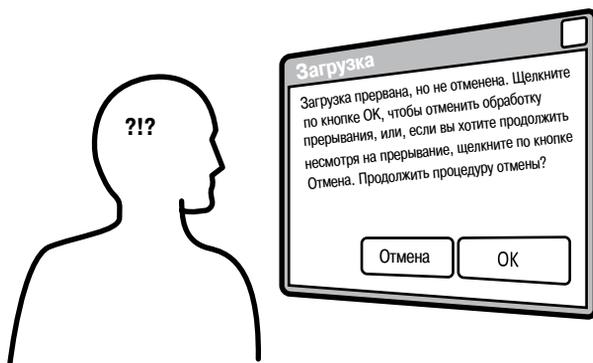
В некоторых случаях вы должны стремиться настроить уровень сложности так, чтобы он был ниже уровня умений и навыков пользователя. Например, если целесообразно сразу поместить пользователей в зону комфорта при использовании нового решения, вам следует убедиться в том, что их текущий уровень умений и знаний позволит им быстро освоить новое решение: «Если вы знаете X, то Y не составит для вас никаких проблем!». Вторым случаем, где уровни умений и навыков пользователя должны быть выше уровня сложности, является слишком высокая расплата за ошибки. В некоторых критических ситуациях ошибки, допущенные при выполнении мелкой задачи, могут отсрочить общий успех проекта или воспрепятствовать ему. Представьте медицинское оборудование или сложные системы вооружений. В этих условиях пользователи и операторы должны быть так хорошо натренированы,

чтобы взаимодействие с решением было их второй натурой. Существуют также случаи, где полезно обратное – придание сложности более высокого уровня по сравнению с умениями и навыками. Это, например, могут быть ситуации обучения и тренировок, где в идеале задача должна быть слегка трудной для пользователя. Другим подобным случаем являются игры. Слишком простая игра скучна, и это отражает модель Чиксентмихайи (см. рис. 8.4).

## 2. Цели и обратная связь

Обычно, прежде чем пользователи начинают взаимодействовать с вашим решением, у них в голове есть общая цель (например, «Хочу обрезать изображение») и мысленные шаги или подчиненные цели (например, «Скопирую изображение с USB-носителя, сохраню его на своем рабочем столе и открою в программе Super Photo Editor. Потом я проведу мышью по изображению для задания области, которую хочу обрезать»). Однако пользователи обычно не формулируют четко эти шаги и не следуют им как инструкциям, если только они не комментируют свои объяснения или не руководствуются информацией из справочной системы приложения. Скорее, это хранимые в памяти шаги, что иногда называют механической памятью. Такие ясные мысленные цели, согласно модели Чиксентмихайи, важны для достижения потокового состояния.

Следующее, что вы можете сделать, – это обеспечить наличие в вашем UI незамедлительной и четкой обратной связи. Запаздывающая обратная связь побуждает людей задаваться вопросом, выполняется ли вообще задача, а неясная обратная связь буквально останавливает взаимодействие и концентрирует внимание пользователей на попытках понять, о чем идет речь, чтобы решить, как быть дальше (рис. 8.6).



*Рис. 8.6. Неясная обратная связь буквально останавливает взаимодействие и концентрирует внимание пользователей на попытках понять предоставляемую информацию, чтобы решить, как быть дальше*

## Что вы можете сделать

**Убедитесь, что обратная связь UI с пользователем и все его интерактивные средства не являются неясными.** Если вы слышите, как ваш пользователь бормочет что-то вроде «Хм, что происходит?», значит, ваш UI неясен. Для начала полезно представить ваше решение в виде дома с множеством комнат. Вам надо убедиться, что все двери снабжены символами, текстом или подсказками, объясняющими, куда они ведут. В литературе по удобству использования это часто называют *доступностью*. Эти «двери» содержат вкладки, кнопки и любые другие элементы управления, которые приводят пользователя к следующему шагу. Небольшая избыточность здесь не повредит. Даже если пользователь неверно истолкует, куда ведет дверь, и «залетит» не в ту комнату, он сможет «выйти», сохраняя хорошее расположение духа, и испытать альтернативные пути. Цель здесь состоит в том, чтобы пользователи не переставали двигаться.

Обучение на ошибках – обычный способ взаимодействия пользователей с новой средой. О том, как люди учатся на ошибках, написаны книги, и все мы слышали о методе проб и ошибок. Другой рациональный подход к обучению я называю «детективным». Определите, что вам нужно сделать, войдите в дом и систематически осмотрите каждую комнату в поисках соответствующего объекта. Нередко как в обычной практике, так и в лабораториях по исследованию удобства использования можно видеть пользователей, щелкающих мышью по всем подряд меню приложения (обычно слева направо) и быстро просматривающих их пункты в поисках нужного действия или функции. Такое поведение основано не на воспоминаниях и размышлениях («Команда поворота должна быть расположена под командами Редактировать, Изображение»), а на узнавании («Нет... не здесь... нет... нет... вот она!»).

Второе, что вы можете сделать, чтобы повысить ясность UI, – это организовать и по возможности сократить информацию, представляемую пользователю. Движения глаз подскажут вам, что опытные пользователи обычно не тратят времени на чтение текста в диалоговых окнах, например таком, которое показано на рис. 8.6, особенно если абзацы выглядят объемными, где больше трех-четырёх предложений. Такое поведение, вероятно, объясняется тем фактом, что пользователь заранее представляет себе, что должно произойти, и для поддержания непрерывности процесса делает то, что кажется ему наиболее логичным.

Поэтому один из подходов состоит в том, чтобы предположить, что пользователи не будут читать много текста, и в качестве надписей на кнопках использовать глаголы, приглашающие к действию, а не обычные слова ОК или Отмена (рис. 8.7). Если в диалоговом окне необходимо разместить несколько абзацев текста, то целесообразно дать четкий вопрос в конце, причем отдельной строкой. Зная о нижней границе класса



*Рис. 8.7. Исследования движений глаз показывают, что некоторые пользователи избегают чтения текста в диалоговых окнах, если абзац выглядит слишком объемным (слева). Для этого есть две превентивные меры: использовать в качестве надписей на кнопках глаголы, приглашающие к действию, и разместить вопрос в конце, в отдельной строке*

не прерывающей реактивности, описанной в главе 4 «Реактивность», которая составляет две секунды, и исходя из известной средней скорости чтения взрослых английских читателей (около пяти слов в секунду), ради сохранения непрерывности пользуйтесь менее чем десятью словами.

### 3. Ощущение контроля

Вашим пользователям нужно не только готовое представление о последовательности шагов до начала работы с вашим решением, им также требуется набор альтернативных последовательностей, которыми можно попробовать воспользоваться, если первая окажется безуспешной. Даже если не возникнет необходимости обращения к альтернативным последовательностям, присутствующая в глубине сознания мысль, что число вариантов больше единицы, является важным фактором, способствующим сохранению потокового состояния пользователей. Чиксентмихайи так поясняет эту идею: «То, что радует людей, – это не ощущение *нахождения* под контролем, а ощущение *контролирования* трудной ситуации». Это подобно тому, как опытные шахматисты держат в уме несколько вариантов и последовательностей ходов. Средний шахматист знает от четырех до пяти возможных ходов. (Это хорошо изучено и называется расчетом вариантов.) Как можно себе представить, ощущение провала единственного варианта по-настоящему нервирует. С другой стороны, мысль о том, что не сработал один вариант из множества, вполне терпима: «Если связь по сети оборвется, я спокойно могу скопировать электронную таблицу на zip-диск и передать его Дженнифер. Если у нее нет zip-привода, я могу записать файл на свой USB-накопитель или на чистый компакт-диск».

## Что вы можете сделать

**Предоставьте возможность отмены последних операций, «аварийные выходы», кнопки Назад и т. д.** Следует отметить, что потокковое состояние является субъективным и психологическим. Подобно тому как вы не можете принудить кого-либо испытывать радость в путешествии, вы не в состоянии точно сконструировать потокковое состояние для своего решения. Ведь это состояние, испытываемое вашими пользователями. Однако есть то, что вы можете сделать, – это произвести настройки и выполнить модификацию своего решения, чтобы минимизировать или ликвидировать преграды на пути к потокковому состоянию. Взаимодействие без очевидной возможности отказаться от дальнейшей работы или вернуться назад подобно длинному участку шоссе без поворотов. Поэтому включение возможности отмены последних действий, «аварийных выходов» (например, кнопок отмены) и кнопок возврата убеждает пользователей, что по ходу продвижения вперед они могут вернуться назад, для того чтобы скорректировать какой-то шаг; если они обнаружат, что находятся на неправильном пути, то могут принять решение вообще прервать процесс. Цель при этом состоит не столько в помощи пользователям принять верное решение на протяжении всей работы, сколько в обеспечении для них возможности исправить ошибочное решение. Чиксентмихайи пишет, что «опыт потоккового состояния обычно описывается как всепоглощающее ощущение контроля – или, точнее, как исчезновение волнения из-за возможной потери контроля». Запомните, что главное здесь – сохранить движение пользователей и, если среда им незнакома, пробудить их к ее исследованию и изучению.

## Выводы

Основываясь на теории Чиксентмихайи о потокковом состоянии, эта глава описывает три способа обеспечения того, чтобы потокковое состояние пользователя в вашем решении было оптимальным. Первый способ заключается в подстройке сложности использования вашего решения к уровню умений и навыков пользователей. Второй обеспечивает ясность целей и обратной связи за счет удаления неясностей в UI. Последний состоит в предоставлении пользователям среды, в которой они сохраняют ощущение потенциального контроля.

## Кроличья нора

### Психология потока

Csikszentmihalyi, M. *Beyond Boredom and Anxiety*. San Francisco: Jossey-Bass, 1975.

Csikszentmihalyi, M. *Flow: The Psychology of Optimal Experience*. New York: Harper & Row, 1990.

## **Потоковое состояние пользователя в человеко-машинном взаимодействии**

Ghani, J. A. Flow in human-computer interactions: Test of a model. In J. Carey (Ed.), *Human Factors in Management Information Systems: An organizational perspective* (Vol. 3). Norwood, NJ: Ablex, 1991.

Ghani, J. A. and S. P. Desphande. Task characteristics and the experience of optimal flow in human-computer interaction. *The Journal of Psychology*, 128, 1993. 381–391.

## **Потоковое состояние пользователя и пользователи Интернета**

Pace, S. A grounded theory of the flow experiences of Web users. *International Journal of Human-Computer Studies*, 60, 2004. 327–363.

Pace, S. Designing for flow. Proceedings of AusWeb04, the Tenth Australian World Wide Web Conference, Southern Cross University, Lismore, 2004. 8–12.

Pace, S. Understanding the flow experiences of Web users. In *OZCHI 2000 Conference Companion: Interfacing Reality in the New Millennium*. (Eds.) C. Paris, N. Ozkan, S. Howard and S. Lu, CSIRO Mathematical and Information Sciences, North Ryde, Sydney, 2000. 2–5.

Pace, S. The roles of challenge and skill in the flow experiences of Web users. *Journal of Issues in Informing Science and Information Technology*, 1, 2004. 341–358.

## **Скорость чтения**

Fry, E. *Teaching Faster Reading: A manual*. Cambridge: Cambridge University Press, 1963.

Ziefle, M. Effects of display resolution on visual performance. *Human Factors*, 40, 1998. 555–568. (Примечание автора: скорость чтения с монитора на ~10% медленнее, чем скорость чтения печатного текста.)



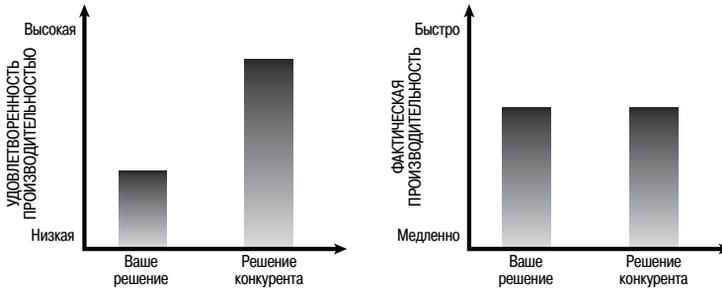
# Глава 9

## Тестирование времени

В некоторых ситуациях необходимо проводить сбор информации о распределении времени. Иногда это вызвано необходимостью сопоставления с эталонами или контроля соответствия заданному уровню эксплуатационных характеристик. Эти измерения касаются системы. В других случаях приходится проводить измерения, касающиеся людей, которые используют вашу систему. Это может быть необходимо из-за жалоб пользователей на низкую производительность системы. И в том и в другом случае сбор надежной и достоверной информации является решающим условием. Эта глава посвящена различным подходам к получению данных о реальной длительности, воспринимаемой длительности и толерантности пользователей.

### Особенности тестирования времени

Если пользователи жалуются, что ваш продукт или какая-то его функция работает медленно, вам следует заняться исследованием и собрать информацию для проверки обоснованности жалобы, и если пользователи окажутся правы, то подумать, можете ли вы что-то предпринять и что именно. В других случаях пользователи могут утверждать, что ваш продукт работает медленнее, чем продукт конкурента. Тогда разумно будет собрать численные данные для определения того, действительно ли имеется существенное различие между продуктами. Ситуация становится интересной, когда различия нет, но пользователи все же говорят, что ваш продукт хуже. В этом случае причиной может быть восприятие или толерантность (рис. 9.1). В следующем разделе



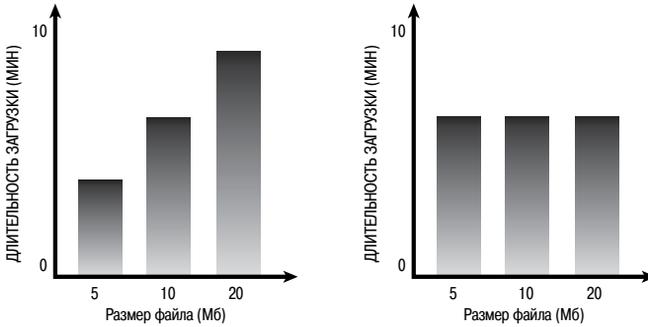
*Рис. 9.1. Сравнение может подтвердить, что пользователи на самом деле оценивают решение вашего конкурента выше по уровню производительности, реактивности и т. д. (слева). Однако сравнение уровней производительности обоих решений (справа) может показать, что объективно они вовсе не различаются. В этом случае вы, возможно, имеете дело с проблемами восприятия или толерантности*

мы обсудим, когда и как вы можете выполнять замеры реальных длительностей, измерять воспринимаемую длительность и оценивать толерантность пользователей.

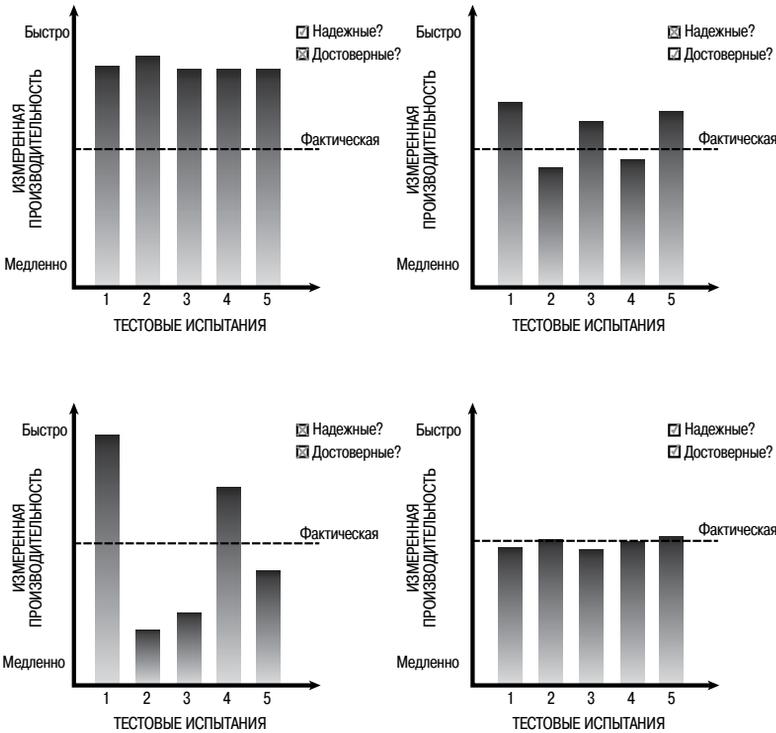
## Получение надежной и достоверной информации

*Студенты, изучающие планирование эксперимента для сбора данных, настойчиво повторяют два слова – надежность и достоверность. Надежность, разумеется, характеризует то, насколько ваши данные надежны. Это показатель того, получите ли вы в повторном эксперименте те же числа, что и в первый раз, если будете применять один и тот же способ получения данных. Если нет, ваши данные не надежны. Достоверность показывает, действительно ли полученные вами данные отражают или представляют то, что вы стремились получить. Иными словами, действительно ли вы измеряете то, что хотите измерить. Иногда в проверке достоверности ваших данных могут помочь эксперименты. Предположим, к примеру, что вы измеряете длительность загрузки информации объемом 10 Мб. Вы намеренно можете урезать наполовину или удвоить объем загружаемых данных и посмотреть, изменяется ли длительность загрузки согласно ожиданиям. Это не позволяет убедиться в 100-процентной достоверности ваших данных, однако может показать вам отсутствие достоверности. Например, если длительность загрузки остается неизменной, ваши данные, могут быть не достоверными. В этом случае, возможно, вы измеряете что-то другое (рис. 9.2).*

В силу того что достоверность и надежность взаимно независимы, результат сводится к одному из четырех следующих возможных наборов данных (рис. 9.3): (1) надежные, но не достоверные, (2) не надежные, но достоверные, (3) не надежные и не достоверные и наконец желаемые



*Рис. 9.2. Данные на диаграмме слева представляют возможные и ожидаемые изменения при увеличении размера файла. Однако, если длительность загрузки остается неизменной (справа) вне зависимости от увеличения или уменьшения размера файла, то, возможно, метод измерения длительности загрузки не обеспечивает достоверности. Вы при этом, вероятно, измеряете что-то другое!*

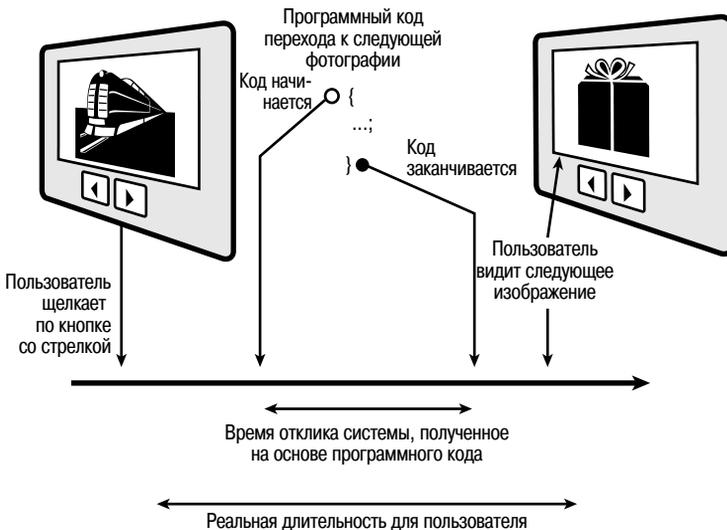


*Рис. 9.3. В силу того что достоверность и надежность взаимно независимы, результат сводится к одному из четырех наборов данных: надежные, но не достоверные (вверху слева), не надежные, но достоверные (вверху справа), не надежные и не достоверные (внизу слева) и наконец желаемые надежные и достоверные (внизу справа)*

(4) надежные и достоверные. Стоит потратить время на проверку того, являются ли ваши данные надежными и достоверными, поскольку решения, как правило, принимаются на основе их анализа.

## Выполнение замеров реальных длительностей

Когда пользователи говорят, что ваш продукт или какая-то его функция работает медленно, вы безотлагательно должны приступить к сбору всех фактов, существенных для решения этого вопроса. Одним из решающих факторов является реальная длительность рассматриваемого процесса. Для пользователя, который жалуется на слишком долгую процедуру входа в систему, это может быть длительность процесса от щелчка по кнопке Войти до момента, когда пользователь узнает, что вход прошел успешно. Для пользователя, который просматривает цифровые фотографии, это может быть время, затрачиваемое на переход к следующей фотографии после щелчка по кнопке с изображением стрелки (рис. 9.4). Обладание объективными данными о длительностях соответствует, разумеется, пребыванию в отправной точке диагностирования многих проблем производительности. Однако это ключ к решению вопроса о том, связаны ли в действительности заявления пользователей с проблемой восприятия или проблемой толерантности.



**Рис. 9.4.** В этом примере время отклика системы обычно определяется как время, необходимое для завершения выполнения некоторого кода, предназначенного для перехода к следующей фотографии. Для пользователя, однако, начальным моментом при восприятии длительности является щелчок мышью, а конечным – момент, когда он видит фотографию на экране

Несмотря на то что обсуждение методов сбора информации о реальных длительностях выходит за рамки этой книги, необходимо коснуться некоторых правил выполнения замеров реальных длительностей.

## Тщательно определяйте реальные длительности

Важно то, как вы определяете начальный и конечный моменты рассматриваемого процесса. При разработке программного обеспечения время отклика системы часто определяется как время, необходимое для завершения выполнения некоторого кода или требуемое системе для выполнения команды. Однако, если взаимодействие инициировано пользователем – например, посредством щелчка мышью, нажатия клавиши или голосовой команды, – начальный момент должен определяться вводом команды пользователем, а конечный момент фиксируется тогда, когда пользователь видит, что команда успешно выполнена.

Например, исследования показали, что проходит примерно от 10 до более 70 мс до того, как операционная система распознает нажатие клавиши, кроме того, на отображение символа требуется дополнительное время, которое зависит от частоты обновления экрана. Поэтому на практике рекомендуется к значениям времени отклика, полученным с помощью кода, добавлять 50 мс в том случае, если время отклика системы определяется как время от момента, когда пользователь выполнил ввод (например, щелчком мышью), до момента, когда он ощущает, что ввод воспринят системой (отжатие кнопки). Если время отклика системы оценивается исключительно как характеристика производительности системы, добавлять 50 мс не требуется.

## Выбирайте надлежащую точность

Разные виды взаимодействия требуют различной точности. Простой способ определения подходящей вам точности состоит в том, чтобы задать себе вопрос, является ли интервал длительностью 0,001 с для вас и для пользователя значимым. Если да, вам требуется миллисекундная точность. Если нет, поднимитесь по шкале времени до значения 0,01 с и задайтесь тем же вопросом относительно нового интервала. Продолжайте увеличивать величину интервала, пока не придете к выводу, что различие обретает смысл. Например, если некоторая функция вашего продукта выполняется за время между 5 и 10 с, длительности величиной 0,001, 0,01 и 0,1 с не будут иметь значения. Однако 1-секундное различие на фоне длительности между 5 и 10 с может стать заметным. Следовательно, здесь вы подведете черту и сделаете вывод, что ваши данные должны характеризоваться точностью 1 с.

Обычно различия в несколько миллисекунд не ощущаются пользователем (к примеру, между 0,321 и 0,322 с), если только (и до тех пор, пока) множественные различия не складываются в картину, которую



*Рис. 9.5. Требуемая точность и методы получения данных для некоторых характерных видов пользовательского взаимодействия. Прямой пользовательский ввод, например нажатия клавиши, перемещения мыши, щелчки мышью и использование сенсорного экрана, обычно требует точности до децисекунд. Более длительные взаимодействия, например установка приложения, конечно, не потребуют того же уровня точности*

пользователи начинают воспринимать. В одной из предыдущих глав мы говорили о правиле 20% и о том, что если длительности различаются более чем на 20%, то пользователям обнаружить различие легче. Однако людям свойственны физиологические ограничения, блокирующие нашу способность обнаруживать очень малые различия длительностей. Например, мы не можем обнаружить различие в одну миллисекунду. Исследования показывают, что минимальный порог обнаружения примерно равен 15 мс. А различие человеком музыкального звука, к примеру, происходит только при превышении 20 мс. Ниже этого порога любой звук кажется щелчком.

Поскольку пользователи связаны с UI, взаимодействия, которые, по их представлениям, должны происходить мгновенно, обычно требуют точности от десятых до сотых долей секунды. Прямой пользовательский ввод, например нажатия клавиш, перемещения мыши, щелчки мышью и использование сенсорного экрана, обычно требует такой точности. Более длительные взаимодействия, например установка приложения, конечно, не потребуют того же уровня точности. На рис. 9.5 показана требуемая точность и методы получения данных для некоторых типичных видов пользовательского взаимодействия.

## Выбирайте подходящий метод

В соответствии с требуемой точностью существует несколько способов сбора информации о временных параметрах. Для измерений, требующих очень высокой точности, может понадобиться рассмотреть возможность применения специального оборудования, например обработчика измерений (Measurement Computing) PCI-CTR05. Такие специализированные таймеры и счетчики обычно располагают собственными, независимыми от компьютерной системы, генераторами тактовых импульсов и нередко устанавливаются отдельно. Для основной категории приложений этого обычно не требуется. Обычно применяется более простой и легкий способ получения данных о распределении времени – с помощью приложения с использованием кода для взаимодействий и процессов, требующих достаточно точных измерений. Табл. 9.1 содержит примеры кодов «секундомера». Применение реальных секундомеров и формата времени ММ:СС приемлемо лишь в случаях, когда достаточно секундной точности, например при загрузках длительностью несколько секунд и выше.

*Таблица 9.1. Простые примеры определения реальной длительности на разных языках программирования*

Язык программирования	Код «секундомера»
Visual Basic	<pre>Dim sw As New Stopwatch sw.Start() 'Perform something here sw.Stop() Console.WriteLine(sw.ElapsedMilliseconds) Console.ReadLine()</pre>
C#	<pre>Stopwatch sw = new Stopwatch(); sw.Start(); //Perform something here sw.Stop(); Console.WriteLine(sw.ElapsedMilliseconds); Console.ReadLine();</pre>
C++	<pre>Stopwatch^ sw = gcnew Stopwatch; Sw-&gt;Start(); //Perform something here Sw-&gt;Stop(); Console::WriteLine(sw-&gt;ElapsedMilliseconds); Console::ReadLine();</pre>
ActionScript (Adobe)	<pre>var stopwatch:Stopwatch = new Stopwatch(); stopwatch.start(); //Perform something here trace(Stopwatch.stop());</pre>

Таблица 9.1 (окончание)

Язык программирования	Код «секундомера»
Java	<pre>long startTime = System.currentTimeMillis(); //Perform something here long stopTime = System.currentTimeMillis(); System.out.println(stopTime - startTime);</pre>
PHP	<pre>\$stimer = explode(' ', microtime()); \$stimer = \$stimer[1] + \$stimer[0]; //Perform something here \$etimer = explode(' ', microtime()); \$etimer = \$etimer[1] + \$etimer[0]; printf(\$etimer-\$stimer);</pre>

Отметим, что непроизводительные издержки пользовательского ввода и время, потребляемое результирующим отображением, должны быть вынесены за рамки отчета о распределении времени.

## Не используйте пользователей в качестве хронометров!

Когда люди хотят отследить точную длительность, они обычно ведут счет в уме или вполголоса («одна тысяча – раз», «одна тысяча – два» ...). Но чаще они просто вспоминают, сколько длилось то или иное событие. В этом случае, сообщая о какой-то длительности, многие тяготеют к якорям времени («примерно пять минут», «менее трех секунд» и т. д.). Несмотря на то что первый подход точнее второго, оба они представляют неадекватные измерения объективного времени. Поэтому длительности, полученные от пользователей, должны рассматриваться в лучшем случае как приблизительные оценки. Объективное время должно измеряться объективными средствами!

## Измерение воспринимаемой длительности

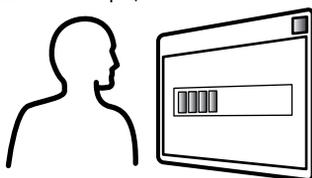
Измерение воспринимаемой длительности не столь просто, как реальной, но иногда вам понадобятся и эти измерения. Следующие разделы ознакомят вас с некоторыми возможными методами измерения воспринимаемой длительности. Напомним, что целью здесь является получение численного значения, которое можно будет сравнить с реальной длительностью, а не оценка удовлетворения пользователя или его восприятия скорости.

### Словесное оценивание

Простейший способ получения сведений о воспринимаемой длительности состоит в том, чтобы спросить пользователей, сколько длилось

то или иное событие. Ограничением такого словесного оценивания является то, что люди склонны сообщать свои оценки, ориентируясь на якоря времени. Возможный способ противодействия влиянию якорей состоит в том, чтобы использовать шкалу для оценки пользователей. Например, после того как пользователь осуществил конкретную операцию, вы можете попросить его установить нижнюю и верхнюю границы длительности ее выполнения. К примеру, «X длилось больше 5 секунд, но не больше 15». Нанесите эту нижнюю и верхнюю границы на противоположные концы шкалы, не делая никаких отметок между ними, и попросите пользователей указать точку внутри диапазона, которая соответствует длительности X, заключенной между двумя предельными значениями. Для получения численного значения надо будет вспомнить немного математики из курса средней школы. На рис. 9.6 показаны соответствующие шаги.

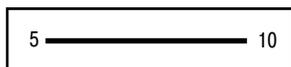
Шаг 1. Попросите пользователя выполнить операцию.



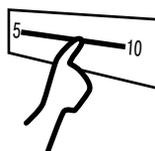
Шаг 2. Попросите пользователя установить минимальную и максимальную границы длительности.



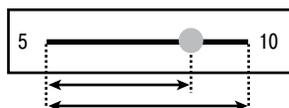
Шаг 3. Нарисуйте шкалу с указанными границами, без отметок внутри.



Шаг 4. Попросите пользователя указать точку между двумя границами, показывающую, какова была длительность выполнения операции.



Шаг 5. Вычислите значение относительно двух границ.



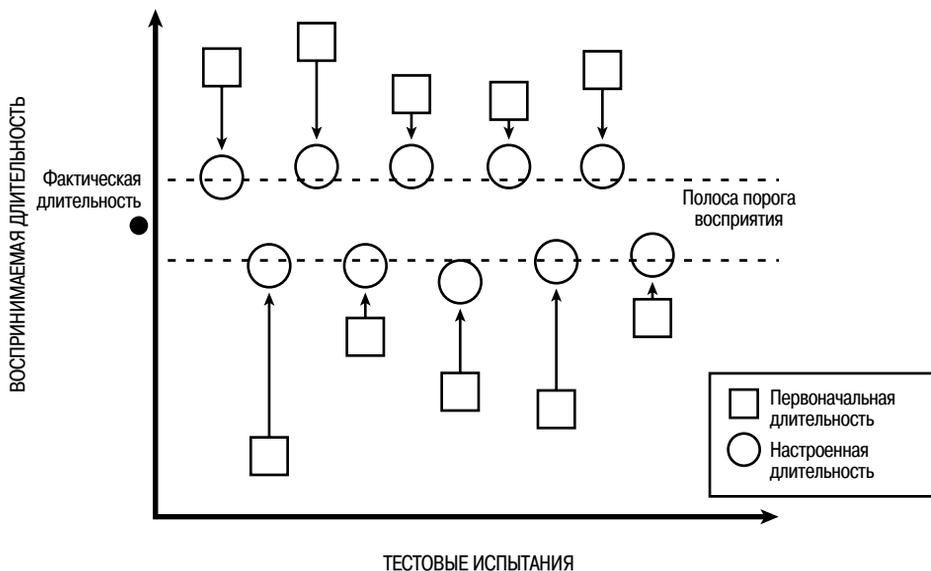
**Рис. 9.6.** Словесное оценивание ограничено в том плане, что люди склонны сообщать свои оценки в привязке к якорям времени. Схема иллюстрирует способ противодействия влиянию якорей

## Репродуцирование

Рассмотрим другой способ получения от пользователей информации о воспринимаемой длительности. Он состоит в том, что пользователи должны, мысленно представляя себе процесс выполнения задачи, комментировать его вслух. При этом можно пользоваться секундомером или просто сказать, когда событие началось и окончилось. Вы же замечайте, как долго продолжалось репродуцирование, и будете считать эту величину воспринимаемой длительностью. Пусть, например, задача пользователя – щелкнуть по кнопке для запуска процесса поиска. Завершением взаимодействия будет появление результатов поиска. После того как пользователи выполнят эту задачу, дайте им секундомеры и попросите произвести их запуск и останов, чтобы определить, какова, по их мнению, длительность поиска. В качестве альтернативы предложите им средства для воспроизведения своих взаимодействий в цифрах. Так или иначе, несколько испытаний позволят вам получить вполне удовлетворительную оценку воспринимаемой длительности. Ограничением такого подхода является то, что он хорошо работает лишь при сравнительно малых длительностях.

## Настройка

Сходный с репродуцированием метод состоит в том, чтобы позволить пользователям настраивать скорость (например, воспроизведения видео) до тех пор, пока она не окажется той, что обеспечит длительность, которую они полагают реальной. Например, предположим, что пользователи переходят на веб-сайт и наблюдают некоторую последовательность событий до тех пор, пока веб-страница не будет полностью загружена. Покажите пользователям видеозапись загрузки веб-страницы и обеспечьте им возможность настраивать скорость воспроизведения до тех пор, пока она, по их мнению, не окажется равной реальной скорости; при этом они не должны видеть никаких численных показателей настройки. Вдобавок по возможности во всех испытаниях устанавливайте начальную скорость воспроизведения случайным образом. Так, если воспроизведение вначале будет медленным, пользователи увеличат его скорость, и наоборот. Как и для других методов, здесь действует то правило, что несколько испытаний лучше, чем одно. Применяя этот метод, вы довольно часто будете сталкиваться с тем, что получаемые данные образуют группы, расположенные ниже и выше реальной длительности. Одна группа, вероятно, будет представлять испытания, в которых пользователи увеличивали скорость, а другая – в которых снижали (рис. 9.7). Не вдаваясь в детали, скажем, что, скорее всего, это происходит из-за наличия у пользователей порога обнаружения различия. Помните правило 20%?



*Рис. 9.7. Применяя метод настройки, нередко можно наблюдать объединение данных в группы, одна из которых относится к испытаниям, где пользователи увеличивали скорость, а другая – к испытаниям с понижением скорости*

## Оценивание толерантности

В одной из предыдущих глав мы говорили о том, что рассматривать воспринимаемую длительность в отрыве от толерантности бессмысленно. Само по себе утверждение пользователя, что извлечение записей о клиентах из базы данных заняло десять секунд, для вас бесполезно и не интерпретируемо. Однако, если пользователь добавляет к сказанному, что обычно это занимает лишь две секунды, наше понимание ситуации начинает улучшаться. Имеется множество возможных способов оценивания толерантности, и некоторые из них описаны в данном разделе.

### Ожидаемая реактивность

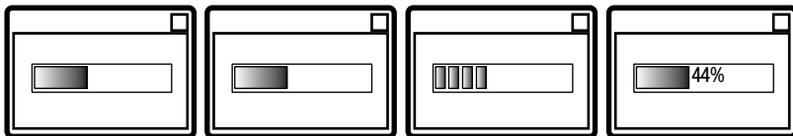
Самый быстрый и «дешевый» способ оценить толерантность пользователей состоит просто в их опросе. Вопрос «Насколько быстрым было X?» мы заменяем таким: «Насколько быстрым должно быть X?» или таким: «Насколько быстрым обычно бывает X?». Это наиболее простой и прямой метод оценивания толерантности, однако, не комбинируя его с другими методами, вы получите информацию, которая может оказаться не слишком полезной. Возможен подход, при котором вы просите пользователей сопоставить реактивность взаимодействия,

подлежащего оцениванию, с другими взаимодействиями, знакомыми ему, например «X должно загружаться так же быстро, как Y» или «X должно происходить быстрее, чем Y, но медленнее, чем Z».

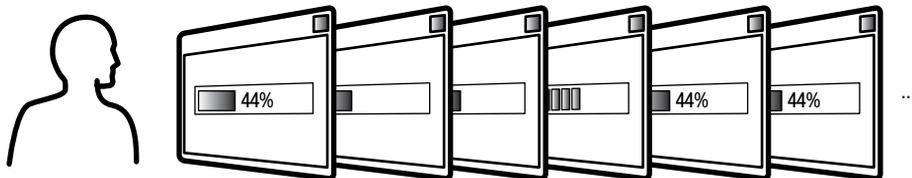
## Экспериментирование

Другой простой способ оценивания толерантности заключается в следующем: вы показываете пользователям различные опытные образцы или версии решения, чтобы они взаимодействовали с ними и давали оценку каждого из них по какой-то шкале реактивности. В идеале вам следует представлять образцы в случайной последовательности и проследить, чтобы каждый из них был показан более одного раза (рис. 9.8). Вы можете поэкспериментировать с этой техникой, если предполагаете, что на толерантность влияют другие факторы. Например, вы можете создать последовательности образцов с одинаковой производительностью и реактивностью, но разных брендов, с различным размером текста, звуками, сопровождающими щелчки мыши, и т. д. Такой подход распространен в изучении рынка и в научных исследованиях. Не

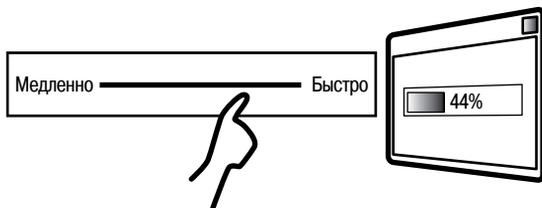
Шаг 1. Создайте различные оболочки или скины для одного опытного образца.



Шаг 2. Показывайте образцы в разной последовательности и дайте пользователям возможность взаимодействовать с каждым из них более одного раза.



Шаг 3. Попросите пользователей дать оценку производительности каждого образца.



*Рис. 9.8. Простой способ оценить толерантность заключается в том, чтобы последовательно показывать пользователям различные опытные образцы или версии решения, попросив их испробовать все варианты и дать оценку каждого из них по какой-то шкале реактивности*

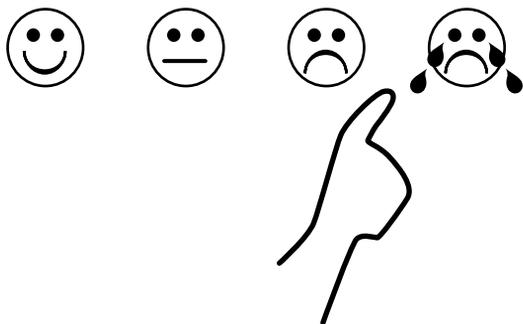
так давно, к примеру, исследователи обнаружили, что дети предпочитают еду в упаковке Макдоналдса такой же еде в «небрендовой» упаковке!

## Продуцирование

Этот метод сходен с методом репродуцирования, рассмотренным ранее. Идея заключается в понуждении пользователей имитировать или описывать взаимодействие для того, чтобы вы поняли, какую реактивность они ожидают от взаимодействия. Это не репродуцирование, где пользователь пытается воспроизвести свой опыт. В продуцировании вы лишь просите пользователя указать, насколько реактивным должно быть взаимодействие.

## Метод кроссmodalного сравнения

Следующая техника называется методом кроссmodalного сравнения. Здесь вы просите пользователей указать свойство в одной системе измерений, совпадающее по интенсивности со свойством в другой системе измерений. Это эффективно, когда пользователям трудно описать качество исследуемой функции или выразить свое отношение к нему либо когда вы хотите избежать субъективных отклонений и искажений. Педиатры пользуются подобным методом, чтобы помочь детям выразить, как сильна их боль, показывая картинки с лицами, иллюстрирующими постепенно возрастающий уровень боли, и прося указать то лицо, которое подходит к их боли (рис. 9.9). Этот метод не предполагает приблизительных словесных оценок, однако то, что вы получаете от пользователя, представляет собой не численное значение, а отношение. Поэтому он хорошо работает там, где пользователи должны сравнить две длительности. Полученная вами информация даст представление о величине различия двух длительностей в восприятии пользователя.



*Рис. 9.9. Педиатры часто помогают детям выразить, как сильна их боль, показывая картинки с лицами, выражающими постепенно возрастающий уровень боли, и прося указать то лицо, которое выражает степень их боли*

## Кое-что о постановке эксперимента

Как уже отмечалось, когда люди пытаются оценивать событие по времени, они склонны делать это субъективно. В контексте человеко-машинного взаимодействия это вряд ли будет способствовать успеху исследования, потому что большинство пользователей не отслеживают намеренно время каждого отдельного взаимодействия. Чаще пользователи полагаются на свою память и оценивают длительность события так: «Установка программы заняла больше получаса» или так: «Поиск занял от пяти до десяти минут». Далее мы приводим правила, которыми вам следует руководствоваться для обеспечения уверенности в том, что вы получаете надежные и достоверные данные.

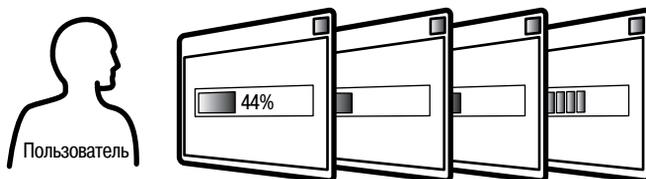
### Влияние последовательности: кто пришел первым

Сравнение длительностей для двух продуктов или решений приходится выполнять не так уж редко. В отношении того, чтобы оба решения оценивал один и тот же человек, есть аргументы «за» и «против». Один из аргументов «за» состоит в том, что вы получаете пары оценок, которые легко сопоставлять, потому что они принадлежат одному человеку. Этот подход к постановке эксперимента называется *интраиндивидуальным* (*within-subject*). При интраиндивидуальном подходе каждому пользователю предлагается испытать все опытные образцы. При *интериндивидуальном* (*between-subject*) подходе каждому пользователю предлагается испытать только один вид образца (рис. 9.10). Один из наибольших рисков в интериндивидуальном экспериментировании заключается в эффекте последовательности, то есть последовательность, в которой пользователь работает с двумя или более объектами, реально влияет на его оценку, заключение или восприятие. Предположительно – и имеется ряд доказательств этого, – результат известного теста «Кока-кола или Пепси-кола», показавшего, что «Пепси» вкуснее, объясняется влиянием последовательности!

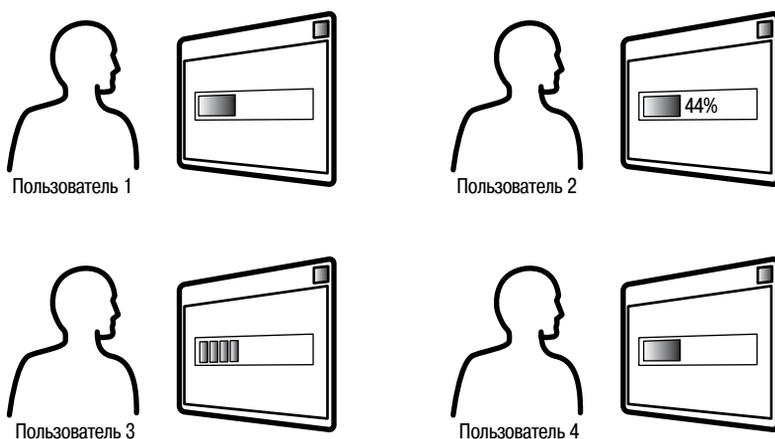
### Влияние протяженности действия и тренированности пользователя

Идеальный план, если таковой возможен, предполагает, что пользователям будет представлена последовательность, более длинная, чем просто А–В. Пусть, к примеру, пользователи проходят через последовательность А–В–А–В–А–В. На восприятие времени пользователя влияет протяженность деятельности и его тренированность, поэтому более длинная последовательность может привести к «выравниванию» восприятия, что теоретически обеспечит вас более ясными и типичными данными. Вспомните эффект Барнабуса: выполнение любого действия в первый раз кажется дольше, чем в последующие.

Интраиндивидуальный подход



Интериндивидуальный подход

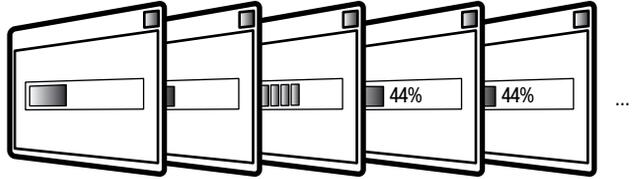


*Рис. 9.10. При интраиндивидуальном подходе каждому пользователю предлагается испытать все образцы. При интериндивидуальном подходе каждому пользователю предлагается испытать только один вид образца*

## Держите кота в мешке!

Как отмечалось ранее, люди при измерении времени пользуются различными стратегиями. В процессе подготовки они проявляют склонность к использованию различных ментальных приемов. Вспоминая пережитый опыт, они полагаются на память. Если пользователи знают, что вам нужны данные для оценки времени, они бессознательно будут отслеживать свое взаимодействие с помощью какой-то субъективной системы отсчета. Во многих случаях это нежелательно. Поэтому вам лучше не сообщать пользователям (по крайней мере, до конца вашего эксперимента), что вы заинтересованы в их оценках длительности. Кроме того, имеет смысл скрыть от пользователей свою цель сбора оценок времени. Один из способов «маскировки» ваших намерений состоит в размещении «сигналов» к оценке времени внутри серий вопросов. Для большей уверенности в том, что пользователи не догадятся об истинных целях ваших исследований, вы даже можете добавить фиктивные испытания, которые не будут учитываться в результатах (рис. 9.11).

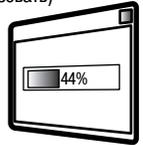
Представьте образцы пользователю.



Попросите пользователей оценить различные качества и свойства каждого образца.  
Разместите вопрос о реактивности среди других.

- |                           |                         |       |                         |
|---------------------------|-------------------------|-------|-------------------------|
| 1. Наглядность            | (Не наглядный) 1        | _____ | 10 (Очень наглядный)    |
| 2. Эстетичность           | (Не приятный) 1         | _____ | 10 (Очень приятный)     |
| 3. Легкость изучения      | (Трудный в изучении) 1  | _____ | 10 (Легкий в изучении)  |
| 4. Реактивность           | (Медленный) 1           | _____ | 10 (Быстрый)            |
| 5. Простота               | (Сложный) 1             | _____ | 10 (Простой)            |
| 6. Легкость использования | (Трудно использовать) 1 | _____ | 10 (Легко использовать) |

To further prevent users from realizing that you are measuring time perception, remove the responsiveness question from a few trials.



- |                           |
|---------------------------|
| 1. Наглядность            |
| 2. Эстетичность           |
| 3. Легкость изучения      |
| 4. Реактивность           |
| 5. Простота               |
| 6. Легкость использования |

Испытание 1

- |                           |
|---------------------------|
| 1. Наглядность            |
| 2. Эстетичность           |
| 3. Легкость изучения      |
| 4. Простота               |
| 5. Легкость использования |

Испытание 2

- |                           |
|---------------------------|
| 1. Эстетичность           |
| 2. Легкость изучения      |
| 3. Реактивность           |
| 4. Простота               |
| 5. Легкость использования |

Испытание 3

- |                      |
|----------------------|
| 1. Наглядность       |
| 2. Эстетичность      |
| 3. Легкость изучения |
| 4. Реактивность      |
| 5. Простота          |

Испытание 4

- |                           |
|---------------------------|
| 1. Наглядность            |
| 2. Эстетичность           |
| 3. Легкость изучения      |
| 4. Простота               |
| 5. Легкость использования |

Испытание 5

**Рис. 9.11.** Один из способов «маскировки» ваших намерений состоит в размещении «сигналов» к оценке времени внутри серий вопросов и исключении истинного предмета исследования (о реактивности) из некоторых испытаний

## Выводы

Получение надежных и достоверных данных является решающим для диагностики проблем, связанных с реальной и воспринимаемой производительностью. При замерах реальных длительностей вам следует проявить тщательность, выяснив, что именно ваш пользователь считает моментами начала и окончания работы, они-то и определяют время отклика. Выбор надлежащей точности и подходящего метода сбора данных также является решающим для обеспечения надежности и достоверности информации. Для измерения воспринимаемого времени вам нужны отзывы пользователей о том, как долго нечто происходило; вам потребуется, чтобы они воспроизвели длительность

или настраивали ее в соответствии со своими воспоминаниями. Для оценивания толерантности вы просто можете спросить пользователей, каков ожидаемый ими уровень реактивности; показать им серию образцов и попросить дать их оценку; позволить пользователям проимитировать действие, чтобы выразить свои ожидания; или попросить оценить реактивность по шкале, не относящейся к реактивности. При сборе информации о воспринимаемой длительности и толерантности важно не упускать из виду потенциальные помехи, например влияние последовательности представления образцов и тренированности пользователя. Наконец, целесообразно скрывать свои намерения, чтобы не дать пользователю догадаться о том, что вы занимаетесь измерением восприятия времени.

## Кроличья нора

### Методологии тестирования

Bindra, D. and H. Waksberg. Methods and terminology in studies of time estimation. *Psychological Bulletin*, 53, 1956. 155–159.

Fraisse, P. Perception and estimation of time. *Annual Reviews of Psychology*, 35, 1984. 1–36.

Gescheider, G. A. *Psychophysics: Method and Theory*. NY: John Wiley and Sons, 1977.

Gescheider, G. A. *Psychophysics: Method, Theory, and Application*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1985.

Zakay, D. Time estimation methods – do they influence prospective duration estimates? *Perception*, 22, 1993. 91–101.

### Оценивание времени человеком

Getty, D. Counting processes in human timing. *Perception & Psychophysics*, 20, 1976. 191–197.

Killeen, P. R. and N. A. Weiss. Optimal timing and the Weber function. *Psychological Review*, 94, 1987. 455–468.

Yarmey, A. D. Retrospective duration estimations for variant and invariant events in field situations. *Applied Cognitive Psychology*, 14, 2000. 45–57.

### Инструментальные ограничения

Beringer, J. Timing accuracy of mouse response registration on the IBM microcomputer family. *Behavior Research Methods, Instruments, & Computers*, 24, 1992. 486–490.

Shimizu, H. Measuring keyboard response delays by comparing keyboard and joystick inputs. *Behavior Research Methods, Instruments, & Computers*, 34, 2002. 250–256.

Ulrich, R. Time resolution of clocks: Effects on reaction time measurement: Good news for bad clocks. *British Journal of Mathematical and Statistical Psychology*, 42, 1989. 1–12.

# Глава 10

## Методы

Все методы этой главы относятся к одному из двух основных подходов, нацеленных на улучшение восприятия времени пользователем при работе с вашим продуктом и рационального распределения этого времени. Первый подход заставляет пользователей воспринимать реальную длительность процесса как более короткую. Это называется *управлением восприятием*. Второй подход заключается в повышении уровня толерантности пользователей в отношении длительности. Это называется *управлением толерантностью*. В данной главе обсуждаются оба подхода, полезных при проектировании UI.

### Управление восприятием

Делайте все возможное для того, чтобы фактическое время, затраченное на исполнение процесса, казалось пользователям более коротким, чем на самом деле. Как отмечено в главе 4 «Реактивность», бывает, что какой-то процесс выполняется слишком быстро, однако везде, где только возможно, добивайтесь того, чтобы пользователи ощущали, что время «летит». Далее мы рассмотрим следующие методы, которые можно применить в вашем решении, чтобы заставить время «лететь» в ощущении пользователей:

- Упреждающий запуск
- Раннее завершение
- Скрытая декомпоновка
- Нисходящие длительности
- Нелинейная индикация выполнения

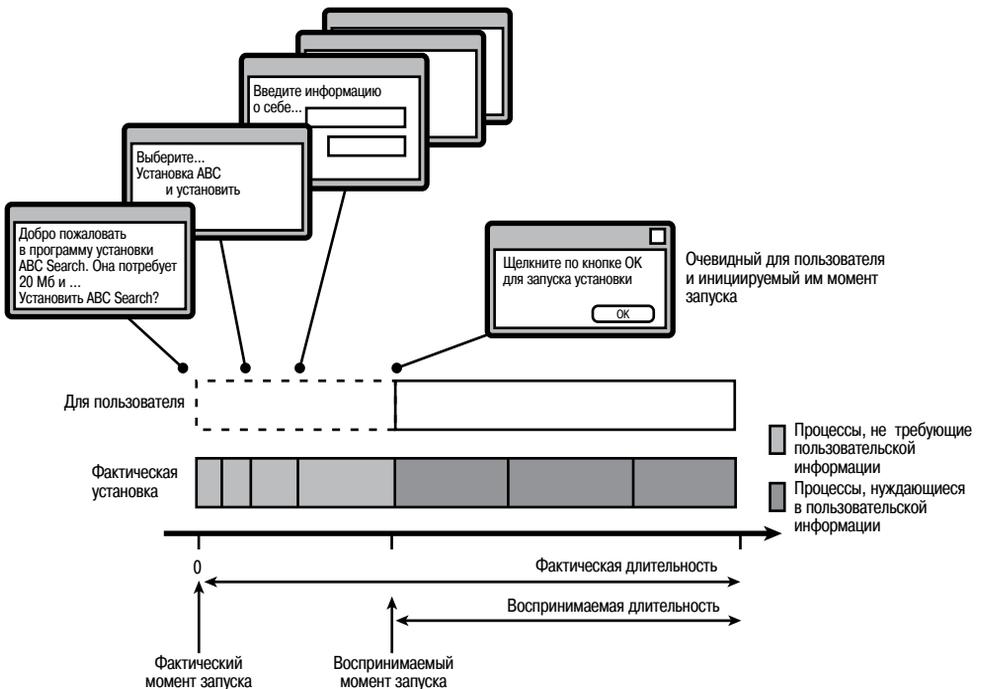
- Неразрывные длительности
- Информирование
- Целенаправленное отвлечение
- Запустить-и-забыть

## Упреждающий запуск

Часто, чтобы запустить процесс, нет необходимости ждать, пока пользователь завершит ввод всей информации. Если есть такая возможность, запускайте процесс, не ожидая завершения ввода пользователем всех данных.

### Как и почему это работает

Люди способны определить длительность только при наличии ясно воспринимаемого времени начала и окончания. Временем старта в их сознании является тот момент, когда в конце концов процесс явно запускается. Ярким примером того, когда следует применять данный способ, является длительная установка программы (рис. 10.1). Многие



**Рис. 10.1.** В верхнем примере в то время, когда пользователь вводит информацию для установки программы ABC Search, процесс установки уже начался. Пользователь же считает началом установки тот момент, когда он явно запускает процесс щелчком по кнопке ОК

процессы установки требуют от нас ввода информации о пути к файлам, выполнения программных и пользовательских настроек. Установка начинается, когда вся информация оказывается введенной. Между тем некоторые виды пользовательской информации не являются необходимыми для запуска каких-то процессов, например копирования установочных файлов во временную папку на компьютере пользователя.

### **Имейте в виду**

Процессы, которые вы запускаете согласно этому методу, прежде чем пользователь инициирует процесс в явном виде, должны быть «безопасными» для пользователей и программной среды. Рассмотрите правовые, этические и другие ограничения, когда принимаете решение о том, какие процессы можно запускать с упреждением. Кроме того, не забудьте встроить механизмы очистки, которые должны действовать в случае отказа пользователя от продолжения работы.

## **Раннее завершение**

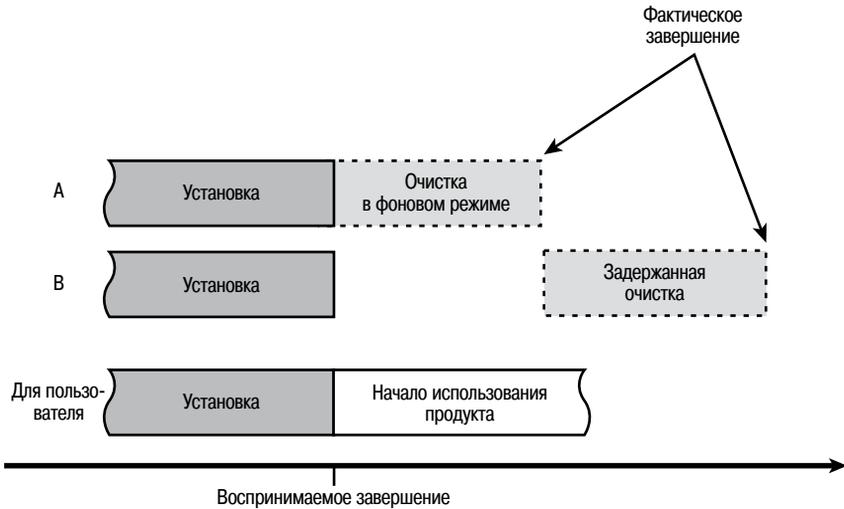
Точно так же как можно начать некоторые действия с упреждением, вы можете заблаговременно «завершить» событие либо за счет выполнения некоторых процессов в фоновом режиме, либо задерживая эти процессы так, чтобы у пользователей создавалось впечатление, что процесс в целом завершен. Способ хорош для процессов, которые требуют довольно продолжительной очистки после своего завершения. У пользователей может возникнуть нетерпеливое желание начать работу с вашим продуктом или переключиться на другую задачу. Поэтому, вместо того чтобы заставлять их ждать, пока ваша программа завершит очистку, позвольте им полноценно работать на той стадии, когда приложение еще выполняет процессы, о которых пользователю нет нужды знать.

### **Как и почему это работает**

Как и для предыдущего метода, преимущество достигается по той причине, что время окончания определяется без учета процессов очистки. Кроме того, этот метод полезен благодаря своей «деликатности» по отношению к беспокойным пользователям. Представьте себе, что постановщики пьес на Бродвее заставляли бы публику после спектакля наблюдать, как рабочие сцены уносят реквизит и разбирают декорации!

### **Имейте в виду**

В примере А на рис. 10.2, когда кажется, что установка завершена, пользователи сразу же могут начать работать с установленным приложением. Если процессы очистки требуют существенных вычислительных ресурсов, следствием может быть заметное снижение производительности



**Рис. 10.2.** Вы можете заблаговременно «завершить» событие либо за счет выполнения некоторых процессов в фоновом режиме (пример А), либо задерживая эти процессы (пример В), так чтобы у пользователей создавалось впечатление, что процесс в целом завершен. Это позволяет пользователям незамедлительно начать работать с вашим продуктом

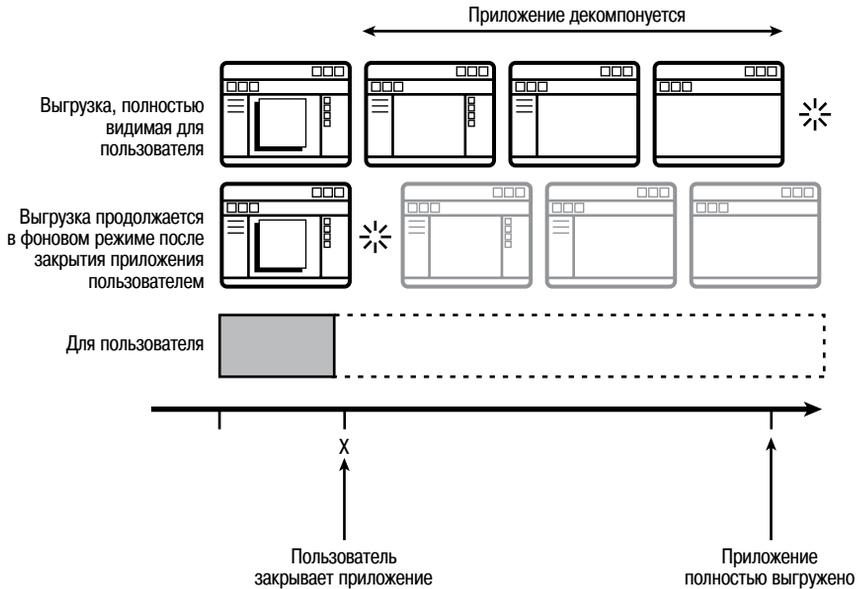
при немедленном начале работы с приложением. Решить эту проблему можно путем задерживания очистки (пример В) или ограничения объема требуемых ресурсов, возможно, посредством их распределения и «прореживания» процессов очистки при их функционировании в фоновом режиме.

## Скрытая декомпоновка

Будучи сходным с предыдущим методом, этот метод требует выгрузки из памяти и декомпоновки составляющих вашего решения, видеть которые пользователю не надо. В примере, показанном на рис. 10.3, при закрытии приложения пользователь не слишком нуждается в том, чтобы наблюдать последовательность исчезновения частей вашего приложения. Напротив, как только пользователи закрывают ваше приложение, немедленно должен исчезнуть весь UI, а необходимая выгрузка из памяти и декомпоновка могут проходить незаметно в фоновом режиме.

### Как и почему это работает

Метод полезен по тем же причинам, что и раннее завершение. Это как в поговорке «С глаз долой – из сердца вон». Вдобавок метод дает иллюзию отличной реактивности, потому что исчезновение происходит



**Рис. 10.3.** Как только пользователи закрывают ваше приложение, немедленно должен исчезнуть весь UI, а необходимая выгрузка из памяти и декомпиловка могут проходить незаметно в фоновом режиме

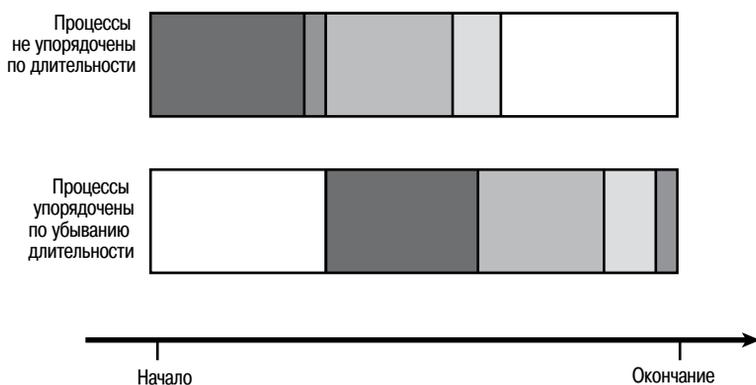
немедленно, в ответ на команду пользователя. Поскольку люди склонны при оценке времени ориентироваться на изменения среды, то с психологической точки зрения метод, делая изменения невидимыми для пользователя, устраняет возможность оценить время, затрачиваемое на выгрузку UI из памяти.

### Имейте в виду

Поскольку этот метод работает аналогично методу раннего завершения, вам следует принять во внимание издержки самого процесса декомпиловки. Если декомпиловка требует солидных вычислительных ресурсов, возможно, следует обеспечить «прореживание» процесса в фоновом режиме, так чтобы он не вызывал слишком большого падения производительности.

## Нисходящие длительности

Многие протяженные процессы составлены из нескольких отдельных subprocessов, что может быть очевидным для пользователя (рис. 10.4). Например, установка приложения может требовать установки механизмов поддержки, технологий, выполнения других условий. По возможности организуйте последовательность subprocessов так, чтобы самые продолжительные из них выполнялись первыми.



*Рис. 10.4. Вверху представлен процесс, составленный из пяти subprocessов с различными длительностями. Организуйте последовательность subprocessов так, чтобы самые непродолжительные из них выполнялись в конце*

## Как и почему это работает

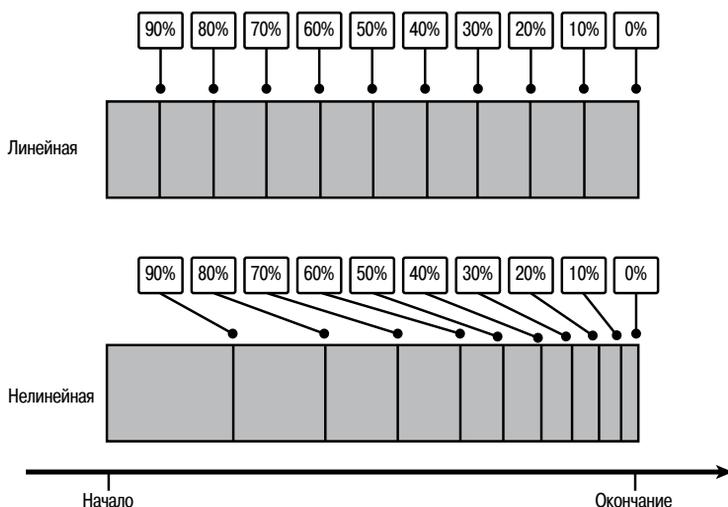
Немногие пользователи любят отслеживать весь процесс, если он занимает много времени. Когда дело доходит до продолжительных задач, большинство пользователей предпочитают узнать, сколько времени займет процесс, а сами тем временем переключаются на другую задачу и возвращаются к процессу перед его завершением. Поскольку пользователи обычно хотят увидеть завершение задачи, выполнение коротких процессов в конце предпочтительнее, ведь демонстрация быстро выполняющихся subprocessов выглядит более обнадеживающей, чем медленное выполнение длительных.

## Имейте в виду

Этот метод хорошо работает с длительными задачами, благодаря тому что пользователи более терпимы к быстро выполняющимся задачам и часто любят наблюдать за процессом. Для продолжительных задач всегда предоставляйте пользователям информацию об оставшемся времени или оставшихся subprocessах, потому что они, скорее всего, обратятся к другим задачам и будут возвращаться к данной время от времени, чтобы посмотреть, как идет процесс.

## Нелинейная индикация выполнения

Для продолжительных процессов вы можете, взамен линейной индикации выполнения (50% означают ровно половину работы), сообщать о продвижении работы нелинейным образом, как показано на рис. 10.5.



*Рис. 10.5. На примере вверху, благодаря тому что продвижение выполнения демонстрируется нелинейным образом, оно кажется более быстрым, если пользователи наблюдают окончание выполнения процесса, что они часто и делают в случае длительных процессов*

При этом если пользователи просматривают индикацию периодически, то продвижение кажется быстрым. Это подобно методу нисходящих длительностей, но, вместо сообщения пользователю числа оставшихся субпроцессов, в этом методе представлено процентное отношение оставшегося времени к общему.

## Как и почему это работает

Этот метод, подобно методу нисходящих длительностей, основан на том факте, что люди склонны наблюдать окончание выполнения длительных задач. Логика, на которой базируется организация нелинейной индикации выполнения в этом методе, состоит в том, что не любые 10% или десятиминутные блоки воспринимаются одинаково. Когда пользователи теряют терпение и толерантность при выполнении длительной задачи, одни и те же десять минут с каждым разом кажутся все более долгими.

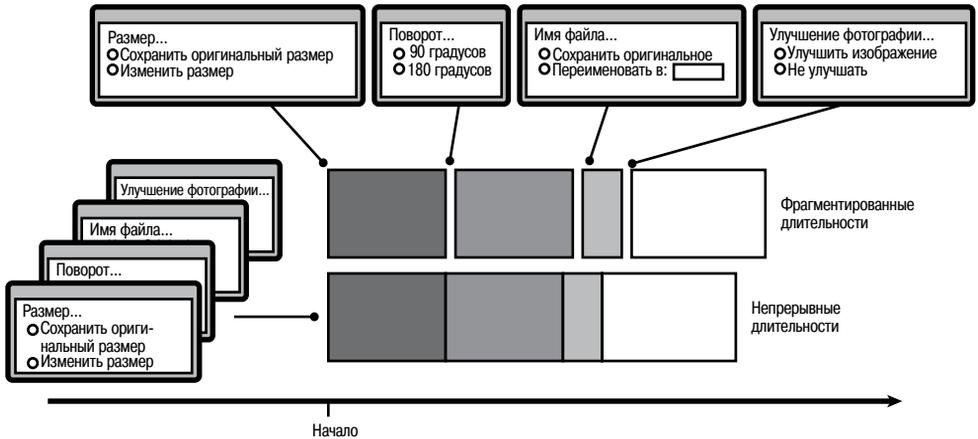
## Имейте в виду

Этот метод предполагает, что точные показатели оставшейся работы или времени не являются критичными для пользователя. Иными словами, ваши пользователи не должны безусловно полагаться на эти значения при принятии важных решений или при выполнении ответственных задач. Помните, что этот метод наиболее предпочтителен для

продолжительных задач и что пользователи должны быть проинформированы о предстоящем длительном процессе. В этом случае очень важно дать пользователям представление о том, как долго он будет длиться.

## Непрерывные длительности

Запросите ввод всей пользовательской информации в начале и обеспечьте, чтобы продолжение выполнения задачи, вплоть до ее завершения, не требовало присутствия пользователя, вместо того чтобы останавливать промежуточные процессы для запросов дополнительных данных от пользователя. Этот подход часто используется при установке ПО, но встречается и в других взаимодействиях. Как показано на рис. 10.6, примером может служить программа редактирования фотографий, которая запрашивает несколько пользовательских вводов для завершения выполнения.



*Рис. 10.6. В этом примере приложение запрашивает ввод всей пользовательской информации в начале (пример внизу), а затем задача выполняется до конца, не требуя присутствия пользователя. Это предотвращает прерывание промежуточных блоков действия, которое дает пользователю ощущение выполнения нескольких процессов и не только вызывает восприятие того же периода времени как более продолжительного, но и может привести к реальному увеличению времени выполнения задачи*

### Как и почему это работает

Нет ничего ужаснее, чем, вернувшись к процессу, который вы запустили час назад, обнаружить, что UI фактически ожидает вашего щелчка по кнопке Да, чтобы вы подтвердили, что действительно желаете продолжить. Подход, при котором задавание вопросов производится шаг за шагом, пусть даже систематически, порождает несколько блоков

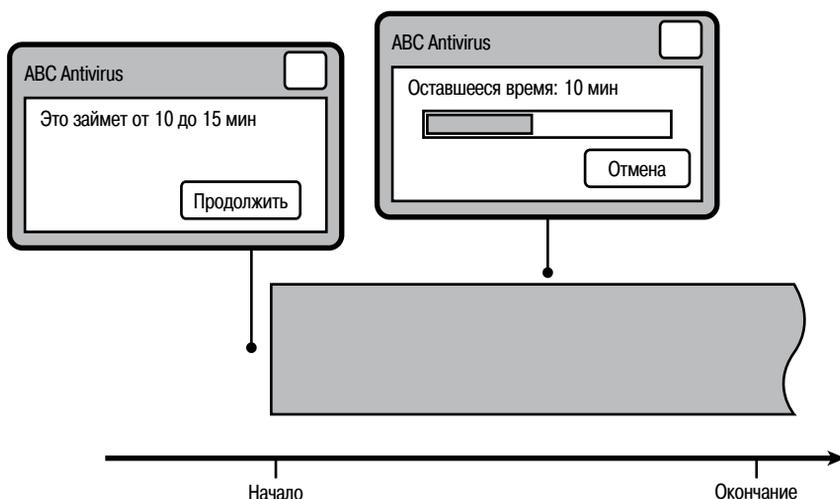
длительности, что не только вызывает восприятие того же периода времени как более продолжительного, но и может привести к реальному увеличению времени выполнения задачи.

### Имейте в виду

Для таких приложений, как редактор фотографий, проведение пользователей через все шаги так, чтобы они понимали, какие манипуляции производятся с изображением на каждом шаге, является вполне оправданным. Альтернативой может быть такое же проведение по шагам, где изменения демонстрируются не на реальном изображении, а на его миниатюре, с тем чтобы дать пользователям представление о манипуляциях, производимых с изображением.

## Информирование

Предоставьте пользователям информацию надлежащего объема, уровня и типа о том, сколько должен продолжаться процесс или задача (рис. 10.7). Исследование показало, что клиенты банка, информированные об ожидании, воспринимают ожидание как более короткое по сравнению с фактическим. Как отмечено в главе 6 «Индикация выполнения», объем, уровень и тип информации зависят от того, что для пользователей значимо и что они в действительности делают с информацией.



**Рис. 10.7.** Два диалоговых окна предоставляют пользователям информацию о том, сколько времени займет выполнение антивирусной программы, но объемы, уровни и типы предоставляемой в окнах информации различаются в соответствии с состоянием процесса

## Как и почему это работает

Неопределенность сведений о продолжительности предстоящего события порождает наибольшее искажение оценки времени. Предоставление пользователям информации об общей продолжительности процесса или о том, сколько времени он еще будет продолжаться, дает им определенность и уверенность, что дело продвигается. Вдобавок это позволяет им чувствовать, что они тратят свое время эффективно и производительно, потому что во время выполнения процесса могут обратиться к другим задачам.

## Имейте в виду

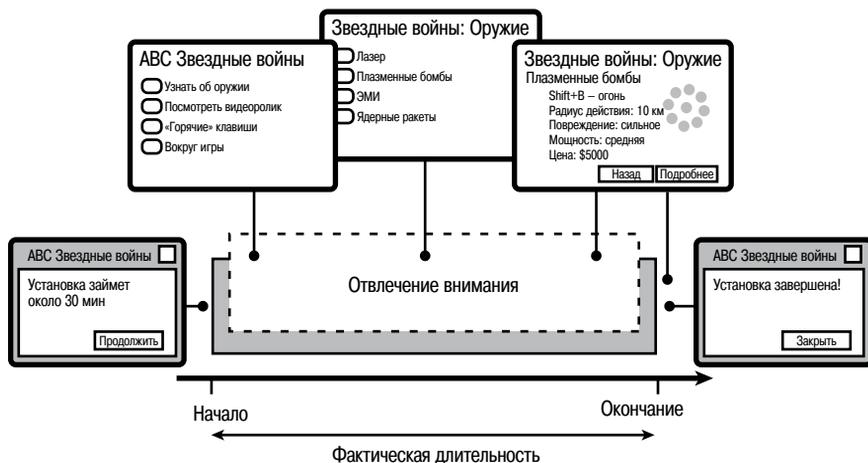
Не забудьте о применении якорей времени, когда формируете оценки времени. Всегда задавайтесь вопросами, содержательна ли информация для пользователя и может ли она быть ему полезна. Администратор базы данных, вероятно, нуждается в отображении детальной информации о прохождении процесса создания резервной копии, в то время как офисному пользователю, скорее всего, не нужны сведения о загадочных системных файлах, загружаемых при запуске его почтовой программы.

## Целенаправленное отвлечение

Этот метод сродни поговорке «Счастливые часов не наблюдают». По ходу длительной задачи или процесса предоставляйте пользователю информацию, интересную или важную для него. Это часто делается при длительных процессах установки.

## Как и почему это работает

Этот простой и широко распространенный метод (вспомните парк аттракционов) полезен не только тем, что отвлекает пользователей от ожидания. Такое отвлечение способствует тому, что пользователи «забывают» о своем ожидании завершения процесса. Кроме того, некоторое внимание к реальному процессу предпочтительнее, чем полный отрыв от него. В примере на рис. 10.8, где пользователь выполняет установку игры, имеет смысл предоставить возможность ознакомления с оружием, которое можно выбирать в игре. Чем больше ваш пользователь будет при этом вовлечен в привлекательное для него взаимодействие, тем лучше. Это означает, что предоставление возможности открытия интерактивных слайдов посредством щелчков лучше, чем просмотр автоматического слайд-шоу.



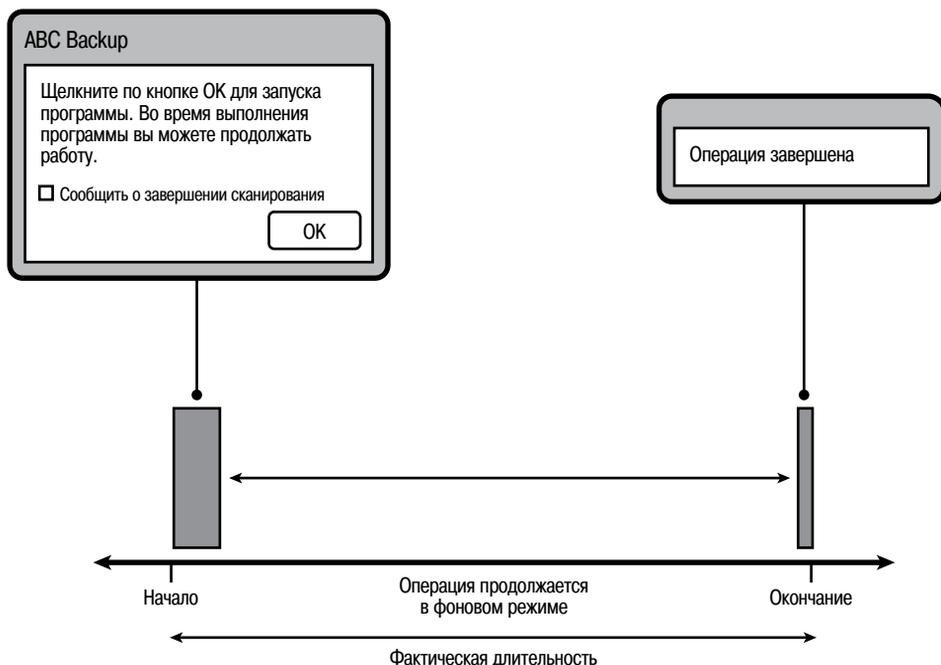
*Рис. 10.8. Во время выполнения длительных задач или процессов предоставляйте пользователю информацию, интересную или важную для него*

## Имейте в виду

Повышая число вариантов отвлекающих занятий (коды доступа к секретному оружию, информация об игровом процессе и т. д.), вы увеличиваете число пользователей, которые могут быть удовлетворены, ведь у каждого свои предпочтения. Как говорится, на вкус и цвет товарищей нет. Кроме того, любое отвлечение должно быть организовано так, чтобы непроизводительные издержки не сильно влияли на производительность реального процесса. Отметим, что циклическое повторение одного и того же материала, например слайд-шоу, – неудачный прием (и в следующей главе мы объясним, почему).

## Запустить-и-забыть

Результат выполнения некоторых процессов и задач, отличных от установки и загрузки, не представляет для пользователя никакой ценности. Важно лишь само по себе успешное завершение. Это, например, может быть системное управление задачами, в частности плановое антивирусное сканирование. Проинформируйте пользователей о выполнении таких процессов в фоновом режиме, как показано на рис. 10.9 на примере резервного копирования ABC. Обратите внимание и на сообщение о завершении.



**Рис. 10.9.** Если результат выполнения процесса представляет незначительную ценность для пользователей, проинформируйте их о выполнении процесса в фоновом режиме и подготовьте сообщение о завершении

## Как и почему это работает

Этот метод сродни поговорке «С глаз долой – из сердца вон». Исследования показали, что во многих случаях если временной промежуток между двумя точками наполнен действиями, то воспринимаемая длительность укорачивается по сравнению с вариантом отсутствия действий. Это называется *иллюзией наполненной длительности*. Поэтому десятисекундный звуковой сигнал кажется длиннее, чем тот же десятисекундный интервал, но с парой сигналов.

## Имейте в виду

Важнейшим фактором успеха этого метода является отвлечение. Удостоверьтесь в том, что выполняемый процесс вообще не будет привлекать внимания пользователей. Если пользователь по какой-то причине вынужден ждать окончания процесса, ожидание не уходит «из сердца вон», а скорее наоборот. Это доказано исследованиями: ничем не заполненное время воспринимается как более длительное по сравнению

с заполненным. Разлука усиливает любовь! Есть несколько простых способов предотвратить праздное бездействие пользователей, например предложить им продолжать заниматься другой работой или обеспечить автоматическое выполнение полезной задачи по окончании процесса (в частности, выключения компьютера). Кроме того, важно дать пользователям возможность проверять степень выполнения и остановить процесс, если в этом возникнет необходимость (например, при выключении компьютера).

## Управление толерантностью

Во многих случаях невозможно убедить пользователей в том, что длительность меньше фактической. Вообще говоря, это относится к довольно длительным процессам, например таким, как установка программ или антивирусное сканирование. В этих обстоятельствах обдумайте стратегию управления толерантностью (возможно, в дополнение к управлению восприятием). Методы управления толерантностью не направлены на маскировку протяженных процессов, вместо этого они имеют целью повысить толерантность и терпеливость пользователей. На последующих страницах мы рассмотрим восемь таких методов:

- Меньше обещать, больше предоставлять
- Модель шкалы цен
- Якоря времени
- Ценность ожидания
- Буферизация и предложение
- Лишь в первый раз
- Контекстные контрольные данные
- Отсчет последних секунд

### Меньше обещать, больше предоставлять

Объявляя характеристику длительности процесса, слегка завысьте фактическую величину. Когда процесс оканчивается раньше, чем было обещано, это воспринимается как «приз».

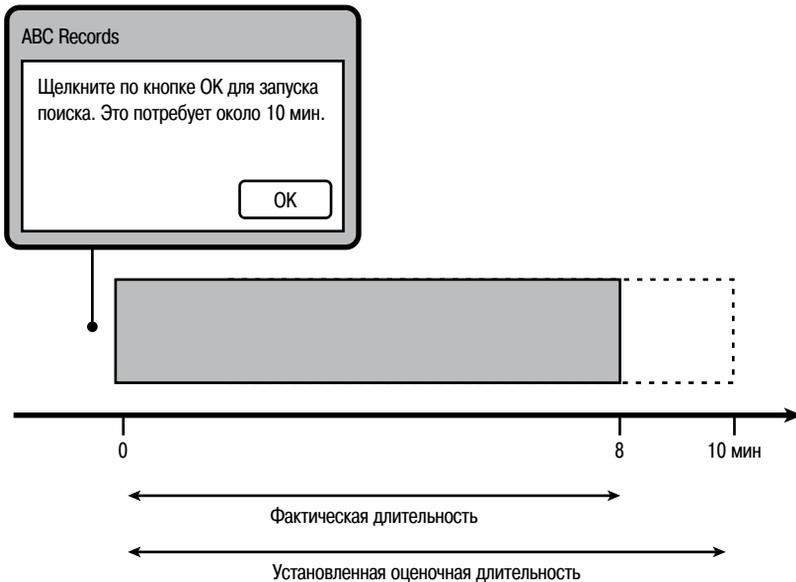
#### Как и почему это работает

Этот метод применяется во многих областях бизнеса и обслуживания. В ресторанах многие официанты называют посетителям время ожидания с завышением предполагаемой величины. Посетители, вставая из-за стола раньше, чем рассчитывали, бывают приятно удивлены. Тот же

метод применяется в парках аттракционов, таких как «Уолт Дисней уорлд» и «Диснейленд», где отображаемые оценки времени ожидания на аттракционах обычно превышают средние фактические.

### Имейте в виду

Не переусердствуйте, преувеличивая оцениваемую длительность. Это означает, что, если процесс реально выполняется за 8 мин (рис. 10.10), вам не следует говорить, что он займет 30 мин. Правило большого пальца – использовать ближайший больший якорь времени (они обсуждались в главе 7 «Представление времени»).



*Рис. 10.10. В этом примере объявляемая длительность равна 10 мин, тогда как реальная длительность составляет 8 мин. Когда процесс заканчивается раньше объявленного времени, это бывает приятно*

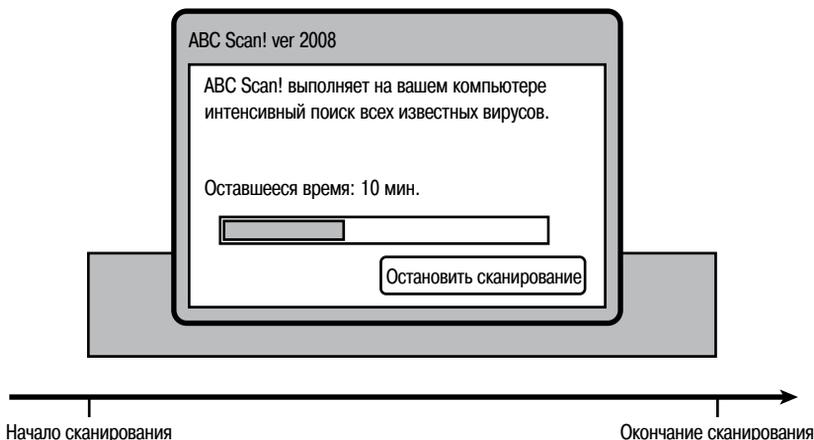
## Модель шкалы цен

Во время выполнения длительной задачи рассказывайте пользователям о ее полезности. Если они будут понимать, что система выполняет задачу, которая даст им преимущества или принесет пользу, недоступную с помощью других средств, они отнесутся к ожиданию с большей толерантностью.

### Как и почему это работает

Этот метод работает благодаря формированию толерантности пользователя путем убеждения его в необходимости исполняемого процесса.

Например, на рис. 10.11 пользователю сообщается, что антивирусная программа выполняет «интенсивное» обнаружение вирусов в его системе. Эта «значимость» указывает на то, что пользователь должен дать задаче выполниться до конца, в отличие от того случая, когда UI просто сообщает о «сканировании». Название этого метода восходит к веб-сайтам по продаже авиабилетов, дающим пользователям возможность поиска наилучшего варианта среди «миллионов предложений». Если вы хотите выбрать наилучшее предложение, надо провести глубокий поиск, не так ли?



*Рис. 10.11. В этом примере антивирусная программа недвусмысленно сообщает пользователю о том, что выполняет насущную и важную задачу, полезную и выгодную пользователю. Это формирует толерантность пользователя, потому что ожидание принесет ему ощутимую пользу*

## Имейте в виду

Значимость задачи, о которой вы сообщаете пользователям, должна быть для них существенной. Пользователь, сознательно запускающий процесс сканирования, вероятно, осведомлен о полезности и важности процесса сканирования и воспримет длительность с большей толерантностью. А вот запланированное сканирование, начинающееся автоматически, без вмешательства пользователя, может показаться невыносимым. В этом случае до пользователя должна быть доведена другая информация о пользе сканирования, например «Со времени последнего сканирования вашего компьютера стало известно о 142 новых вирусах».

## Якоря времени

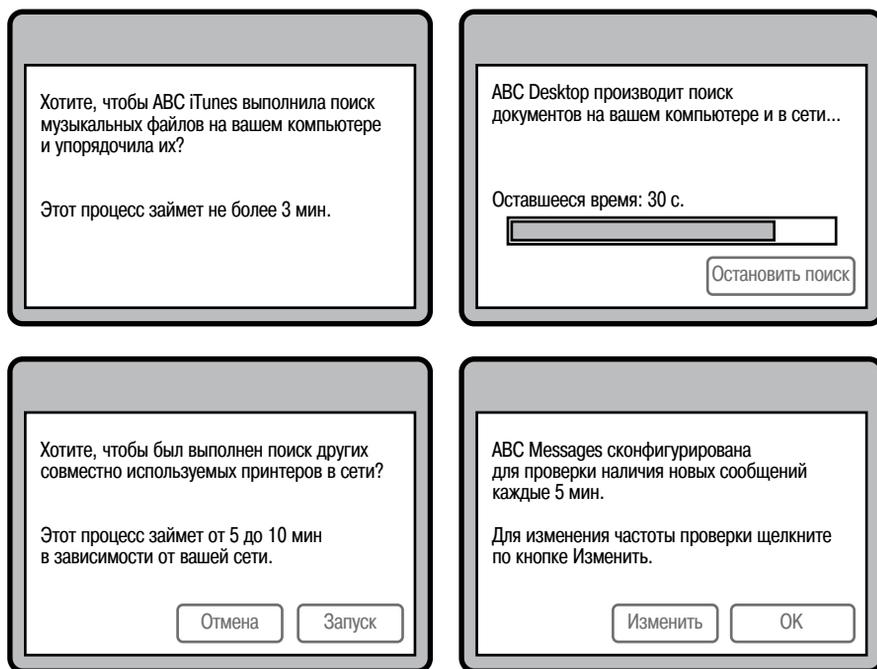
При отображении времени в UI пользуйтесь якорями времени, особенно при объявлении максимального времени («менее чем»), оценивании оставшегося времени, оценивании диапазона или частоты.

## Как и почему это работает

За более подробной информацией о якорях времени обратитесь к главе 7 «Представление времени».

### Имейте в виду

Якоря времени полезны для системы при представлении времени пользователям, однако их использование не является необходимым. На рис. 10.12 на примере внизу справа щелчок по кнопке Изменить дает пользователям возможность указать желаемое время (1, 2, 3, 4, 5, 6 и т. д.), не всегда совпадающее с якорями времени.



*Рис. 10.12. На примерах вверху якоря времени используются для представления объявленного максимального времени (вверху слева), оценки оставшегося времени (вверху справа), оценки диапазона (внизу слева) и частоты (внизу справа)*

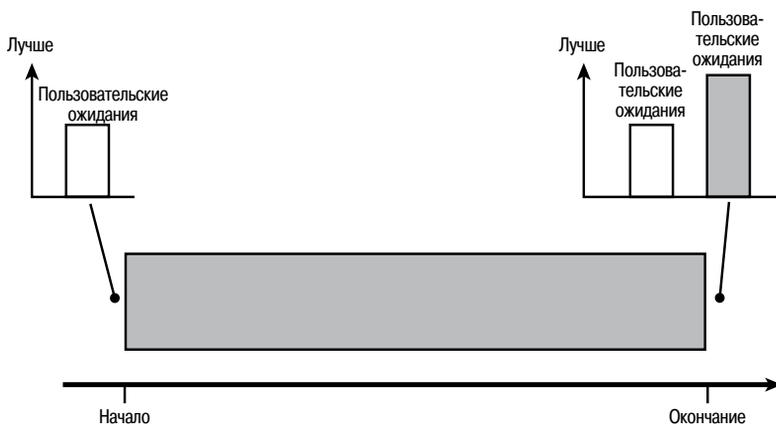
## Ценность ожидания

Убедите пользователей в том, что ожидание будет вознаграждено, и обеспечьте, чтобы получаемый результат соответствовал их ожиданиям или

превосходил их. Заставьте их ощущать «ценность ожидания». Если пользователи когда-нибудь будут повторять тот же процесс, их толерантность окажется выше.

## Как и почему это работает

Эффективность этого метода основана на формировании толерантности во время ожидания и обеспечении удовлетворения после ожидания (рис. 10.13). К примеру, многие рестораны в своих зонах ожидания гордо выставляют восторженные отзывы из журналов и газет. Помимо того что они популяризируют и хвалят ресторан, эти настенные «трофеи» работают на формирование толерантности во время ожидания. Обеспечение обслуживания, действительно вознаграждающего ожидание, приобретает решающее значение. Исследование показало, что сильная положительная оценка потребителей, полученная по окончании длительного ожидания, исправляет отрицательное впечатление, сформированное самим по себе длительным ожиданием. Это значит, что, стоя в очереди, потребитель может вначале лишь страдать от ожидания, но, получив прекрасное обслуживание, тот же потребитель может решить, что период ожидания был не так уж мучителен, как казалось раньше. В таком случае потребительская толерантность в следующий раз будет выше благодаря пониманию того, что ожидание в ресторане вознаграждается.



*Рис. 10.13. Толерантность формируется тем, что пользовательское восприятие качества вашего решения превосходит ожидания*

## Имейте в виду

Существует психологический порог, выше которого большинство людей уже не хотят ждать. Люди, не знающие о ценности ожидания или не

убежденные в ней, будут, вероятно, принадлежать к числу тех, кто ждать не согласится. Для тех, кто решается на ожидание, неопределенность (недостаток информации об ожидании, к примеру) быстро съедает все терпение и толерантность. Известны виды бизнеса, где задержка искусственно добавляется ко времени обслуживания для формирования жажды (иногда буквальной) обслуживания. (Вспомните сцену «фашистского супа» из сериала «Сейнфелд» (Seinfeld)!) Запомните, что это тонкий прием, и его следует применять, только если ваш продукт или сервис действительно имеет достаточный уровень для восторженного восприятия.

## Буферизация и предложение

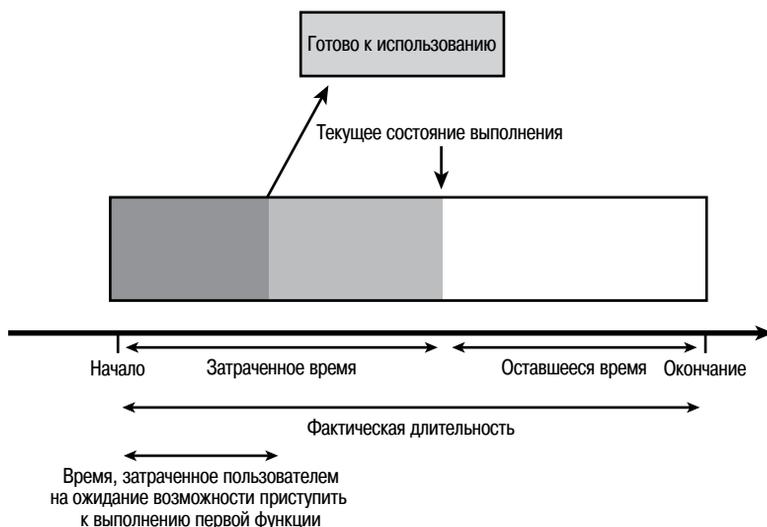
Предоставляйте пользователям что-нибудь, чем они смогут себя занять, прежде чем станет доступным решение или продукт в целом. Ярчайшим примером является потоковое видео в Интернете – вам не нужно ждать загрузки всего видеофайла, для того чтобы начать просмотр.

### Как и почему это работает

Это эффективно просто потому, что пользователям надо меньше ждать. К примеру, многие приложения позволяют пользователю начать работу, не будучи полностью загруженными. Рассмотрим большое приложение, включающее множество функциональных компонентов, такое как программа электронной почты. Когда пользователи запускают программу электронной почты, они, скорее всего, сразу захотят выполнить некоторые простые действия, а не ждать, пока будут доступны все возможности программы. Если почтовая программа сложна и требует много времени на загрузку, то имеет смысл вначале загрузить те функции, которые пользователи могут начать немедленно выполнять (проверка новых электронных сообщений, создание нового сообщения и т. д.), а потом уже то, что срочно пользователям, вероятно, не потребуется (настройки, конфигурирование учетных записей, проверка правописания и т. д.).

### Имейте в виду

Некоторые функции, которыми люди желают воспользоваться, могут оказаться зависимыми от других частей программы, поэтому полезно составить схему того, что из приоритетных пользовательских задач и паттернов использования технически возможно предоставить немедленно. Для решений типа потокового контента необходимо определить, какой объем данных следует буферизировать, чтобы пользователь не «застревал» в ходе процесса (рис. 10.14).



**Рис. 10.14.** В показанном примере, когда появляется одна готовая к использованию часть решения, она становится доступной пользователю. Этот метод обычно используется в интернет-решениях, основанных на потоковом видео или аудио, но он также пригоден, например, для приложений, работающих на стороне клиента

## Лишь в первый раз

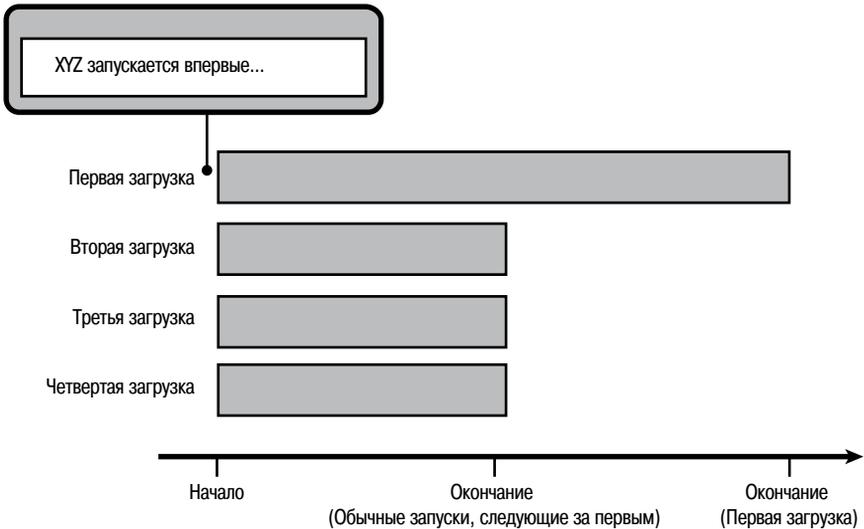
Некоторые решения имеют задержки, возникающие только при использовании в первый раз. В таких случаях информируйте пользователя, что это однократная задержка, которая случается только тогда, когда решение запускается в первый раз.

### Как и почему это работает

Этот метод эффективен, потому что дает пользователям информацию (точнее, разумное объяснение задержки) для формирования толерантности. Это то же самое, что объяснить ваше опоздание незнанием данного района. Этот метод должен использоваться лишь однажды. Впоследствии это будет выглядеть как неудачная отговорка!

### Имейте в виду

Ситуацию, описанную выше и показанную на рис. 10.15, следует отличать от «холодной» загрузки. «Холодная» загрузка – это начальный запуск приложения (например, загрузка компьютера утром). Последующие запуски (например, в течение дня) называются «теплыми» перезагрузками. Некоторые приложения имеют медленную «холодную» загрузку, потому что определенные компоненты приложения требуют



*Рис. 10.15. В показанном примере первая загрузка решения является недопустимо медленной, но последующие загрузки происходят сравнительно быстро. В таких случаях просто информируйте пользователей, что это однократная задержка, имеющая место лишь тогда, когда решение запускается в самый первый раз*

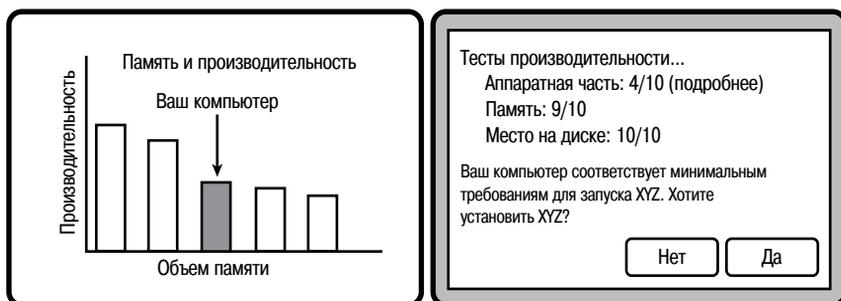
загрузки в память. Когда загрузка в память окончена, последующие процессы запуска выполняются быстрее. При медленной «холодной» загрузке можно использовать, например, предварительную загрузку требуемых компонентов, что, по существу, является подготовительным «разогревом». К этому следует подходить со всей тщательностью и осторожностью, особенно, если «разогрев» отнимает львиную долю производительности, требуемой приложению.

## Контекстные контрольные данные

Предоставляйте пользователям контрольную информацию, чтобы помочь им понять зависимости вашего решения.

### Как и почему это работает

В то время как многие методы повышения толерантности склоняют пользователей, собственно, к тому, чтобы они были толерантными, этот метод, наоборот, посылает пользователю сообщение типа «это не я, это вы». На рис. 10.16 показано два примера. Если говорить более точно, метод переносит бремя ответственности за производительность с самого решения на другие зависимости, например, от пропускной способности сети, скорости работы процессора, памяти и т. д. Один такой



*Рис. 10.16. Решение слева информирует пользователя о производительности его компьютера по сравнению с компьютерами других конфигураций. Аналогичным образом, решение справа представляет пользователю точные контрольные данные и информирует его о том, соответствует ли его система минимальным требованиям для установки решения*

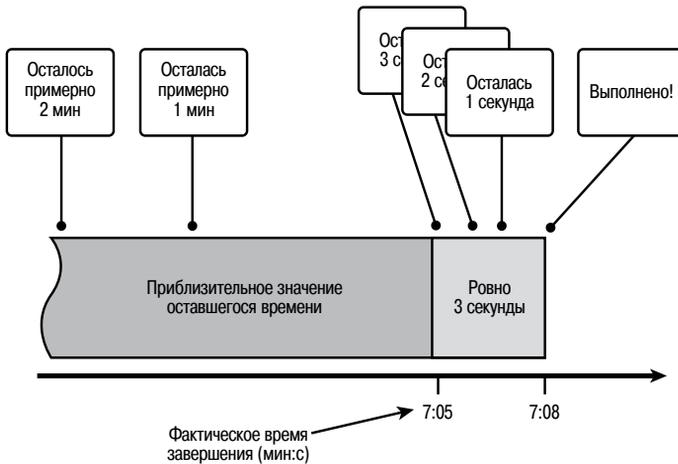
пример – программный пакет Matlab от MathWorks. Программа технических расчетов содержит встроенную функцию (bench), которая выдает множество технических характеристик машины пользователя, от общих до частных. Фактически Matlab проверяет основные данные, по отношению к которым ведется оценивание, и показывает пользователю, что производительность приложения в большой степени определяется аппаратной частью машины пользователя. Аналогично, многие веб-сайты дают пользователям диапазон оценки времени загрузки в зависимости от скорости интернет-соединения пользователя. В других решениях реализуется отображение системных параметров, от которых напрямую зависит производительность компьютера.

### Имейте в виду

Можно поместить слова «в зависимости от вашего аппаратного обеспечения» рядом с оценкой диапазона длительностей. Напомним, что данный метод переносит бремя ответственности на пользователя, поэтому все должно быть выполнено очень корректно. Убедитесь, что в сообщениях UI отсутствуют случайные намеки на то, что система пользователя никуда не годится. Если его система не оптимальна для вашего решения, всегда давайте подробное сообщение о соответствующих аспектах, например недостаточный объем памяти или емкость жесткого диска и т. д.

### Отсчет последних секунд

В ходе процесса используйте якоря времени для представления оставшегося времени, однако, когда процесс практически завершен, добавьте несколько значений в секундах и покажите обратный отсчет времени до нуля (рис. 10.17).



*Рис. 10.17. Когда процесс уже завершен, добавьте несколько значений в секундах и покажите обратный отсчет времени до нуля*

## Как и почему это работает

Метод работает благодаря созданию иллюзии точности. Несмотря на то что отображение таймера обратного отсчета времени на протяжении всего процесса невозможно и не рекомендовано, это оказывается полезным в конце, когда процесс выполнен. В соответствии с данным методом, хотя процесс уже завершен, вы добавляете несколько секунд, чтобы показать, что точность оценки времени завершения идеальна.

## Имейте в виду

Это работает для относительно больших длительностей, по крайней мере в несколько минут. Добавление пяти секунд к десятиминутной длительности будет выглядеть не слишком эффектно. Добавление пяти секунд к трехсекундной длительности – бессмыслица.

## Выводы

Эта глава знакомит вас с девятью методами управления восприятием, которыми надо пользоваться для того, чтобы фактические длительности воспринимались более короткими, чем они есть на самом деле. Если воздействовать на восприятие сложно, следует использовать управление толерантностью. Вторая половина главы знакомит вас с восемью дополнительными методами, служащими для повышения толерантности пользователя.

## Кроличья нора

### Исследования, посвященные веб-сайтам

Bouch, A., A. Kuchinsky and N. Bhatti. Quality is in the eye of the beholder: Meeting users' requirements for Internet quality of service. In proceedings of the SIGCHI conference on human factors in computing systems, 2000. 297–304.

Dellaert, B. G. C. and B. E. Kahn. How tolerable is delay consumers' evaluations of Internet websites after waiting. *Journal of Interactive Marketing*, 13, 1999. 41–54.

Rose, G. M., M. L. Meuter and J. M. Curran. On-line waiting: The role of download time and other important predictors on attitude toward e-retailers. *Psychology and Marketing*, 22, 2005. 127–151.

Ryan, G. and M. Valverde. Waiting in line for online services: A qualitative study of the user's perspective. *Information Systems Journal*, 16, 2006. 181–211.

### Ожидания и удовлетворение потребителей

Antonide, G., P. C. Verhoef and M. van Aalst. Consumer perception and evaluation of waiting time: A field experiment. *Journal of Consumer Psychology*, 12, 2002. 193–202.

David, M. M. and J. Heineke. How disconfirmation, perception and actual waiting times impact customer satisfaction. *International Journal of Service Industry Management*, 9, 1998. 64–73.

Houston, M. B., Bettencourt. L. A. and S. Wenger. The relationship between waiting in a service queue. *Psychology & Marketing*, 15, 1998. 735–753.

Maister, D. H. The psychology of waiting lines. In Czepiel (Ed.), *The Service Encounter*. Lexington, MA: Lexington Books, 1985. 113–123.

Pruyn, A. and A. Smidts. Customers' reactions to waiting: Effects of the presence of 'fellow sufferers' in the waiting room. *Advances in Consumer Research*, 26, 1999. 211–216.

Unzicker, D. K. The psychology of being put on hold: An exploratory study of service quality. *Psychology & Marketing*, 16, 1999. 327–350.

### Исследования и методы индустрии розничной торговли и обслуживания

Evangelist, S., B. Godwin, J. Johnson, V. Conzola, R. Kizer, S. Young-Helou and R. Metters. Linking marketing and operation: An application at Blockbuster, Inc. *Journal of Service Research*, 5, 2002. 91–100.

Hui, M. K. and D. K. Tse. What to tell consumers in waits of different lengths: An integrative model of service evaluation. *Journal of Marketing*, 60, 1996. 81–90.

Katz, K. L., B. M. Larson and R. C. Larson. Prescriptions for the waiting in line blues: Entertain, enlighten and engage. *Sloan Management Review*, Winter, 1991. 44–53.

Kellaris, J. J. and R. J. Kent. The influence of music on consumers' temporal perceptions: Does time fly when you're having fun? *Journal of Consumer Psychology*, 1, 1992. 365–376.

Leclerc, F. How should one be told to hold. *Advances in Consumer Research*, 28, 2002. 78.

Leclerc, F. How should one be told to hold. *Advances in Consumer Research*, 28, 2002. 78.

North, A. C., D. J. Hargreaves and J. McKendrick. Music and on-hold waiting time. *British Journal of Psychology*, 90, 1999. 161–164.

Tansik, D. A. and R. Routhieaux. Customer stress-relaxation: The impact of music in a hospital waiting room. *International Journal of Service Industry Management*, 10, 1999. 68–81.

Tom, G. and S. Lucey. Waiting time delays and customer satisfaction in supermarkets. *Journal of Services Marketing*, 9, 1995. 20–29.

# Глава 11

## Нарушения

Можно создать ощущение того, что время «летит», но можно сделать и так, что время в восприятии пользователей будет «ползти». Все, что приводит к восприятию фактических длительностей как увеличенных по сравнению с теми, какими они в действительности являются, или снижает толерантность пользователей, представляет собой нарушение, потому что препятствует формированию оптимального пользовательского отношения. В этой главе описываются некоторые из таких нарушений и обсуждаются пути их предотвращения и исправления.

### Нарушение восприятия

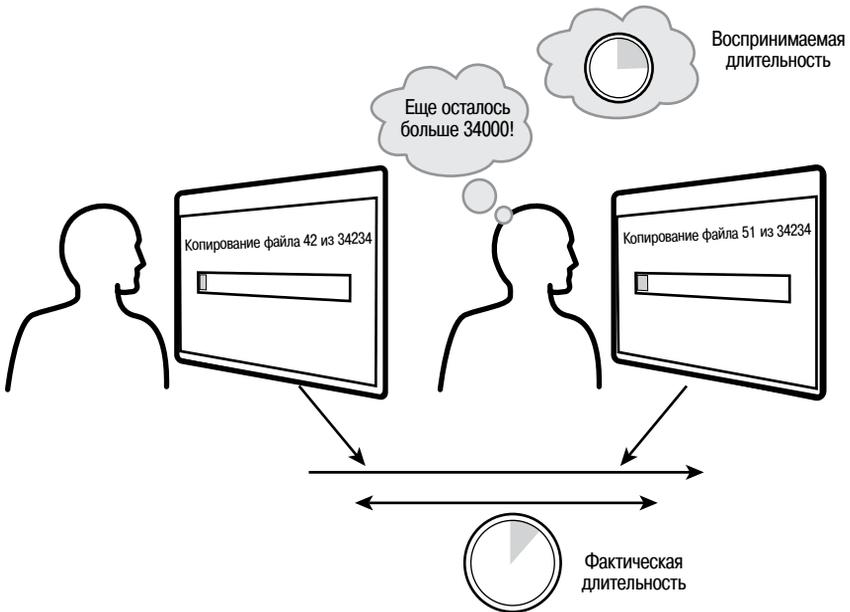
Обычно пользователи не измеряют секундомером каждое свое взаимодействие с компьютером. Пользователи, если только они не проводят специальное тестирование, чаще всего оценивают предполагаемую длительность процесса, проверяя, сколько времени уже прошло, и на основании этого приблизительно вычисляя общую продолжительность. Первая группа нарушений включает те, которые вынуждают пользователей ощущать фактические длительности как удлиненные по сравнению с реальными. На восприятие длительности влияет множество факторов. Ниже приведен список нарушений, обладающих факторами, которые негативно сказываются на пользовательском отношении:

- Наблюдение за чайником
- Не отпускающее ожидание
- Негативная оценка

- Затраченное время
- Эффект Барнабуса
- Перегруженность информацией
- Фрагментированные длительности
- Беспокойство

## Наблюдение за чайником

Необходимость быть внимательным на протяжении всего долгого процесса может привести к оцениванию реальной длительности с завышением (рис. 11.1).



*Рис. 11.1. В этом примере пользователь непрерывно следит за операцией копирования. Это похоже на слежение за чайником с водой, которую надо вскипятить*

### В чем заключается нарушение

Почему, если следить за процессом, то кажется, что вода в чайнике закипает медленнее? Не вдаваясь в научные тонкости, можно сказать, что одно и то же количество воды, закипает примерно за одно и то же время в одинаковых условиях. Главное здесь то, что обычно мы не наблюдаем за выполнением длительных процессов (кипячение воды в чайнике, стирка одежды в стиральной машине, запекание пирога в духовке и т. д.), потому что больше интересуемся тем, когда они закончатся.

Аналогично, пользователи не ведут непрерывное слежение за длительными процессами и более склонны к периодическим проверкам того, не завершился ли он. Если пользователь не может выполнять никакие действия, пока не завершился тот или иной процесс, например поиск нужного файла, обычно его больше всего интересует, когда же этот процесс закончится. Если пользователь неотрывно следит за процессом, не имея информации о том, когда он завершится, то произвольно возникает надежда, что это произойдет в следующую секунду. Подобное нетерпение мы испытываем, ожидая, когда закипит вода в чайнике.

### **Имейте в виду**

Если пользователю нет нужды или не интересно знать подробности процесса, не сообщайте их. Глава 6 «Индикация выполнения» дает некоторые рекомендации на этот счет. Кроме того, помните, что мы привыкли полагаться на знаки и сигналы – свисток чайника или сигнал микроволновой печи, – сообщающие нам о том, что процесс завершился. Это освобождает нас и позволяет обратиться к другим предметам или задачам. Аналогично, вы должны предоставить возможность пользователям и даже поощрить их обратиться к другим задачам, пока выполняется длительный процесс, и обеспечить для них оповещение о завершении процесса. Дополнительно можно обдумать предоставление возможности выполнения следующей задачи (например, перезагрузки компьютера, закрытия приложения и т. д.), когда процесс завершится.

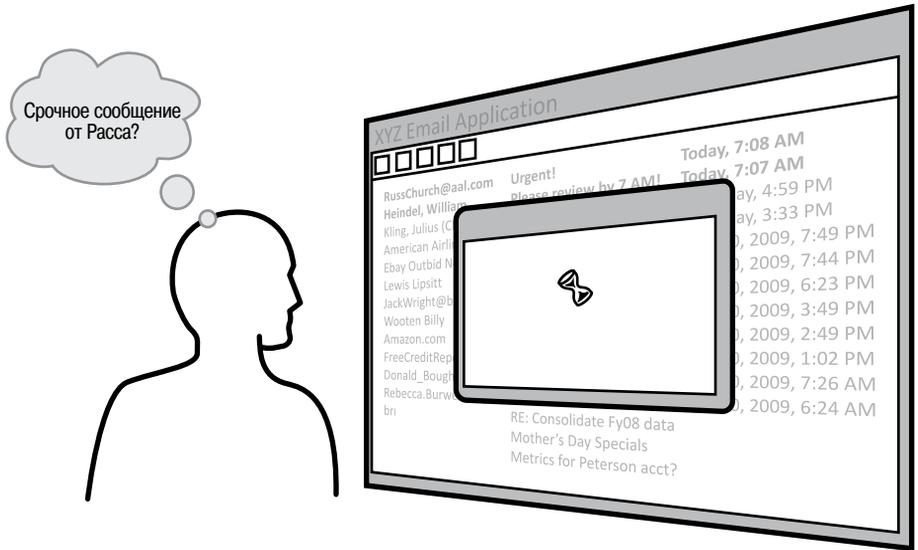
## **Не отпускающее ожидание**

Любые функционирующие процессы, которые держат компьютер пользователя, операционную систему или приложение в качестве «заложника», заставляют пользователя воспринимать «захваченное» время как более длинное по сравнению с реальным.

### **В чем заключается нарушение**

Ради безопасности некоторые современные самоочищающиеся духовки запираются сами во время автоматической мойки. В момент начала самоочистки духовка становится «захваченной», и ее нельзя использовать до завершения процесса. Предположим, что, когда самоочистка уже началась, вы вдруг вспомнили, что вам срочно нужна духовка. Каждая отдельная секунда, проходящая до завершения процесса, мучительна для вас, особенно если нет индикации того, когда все окончится. В большинстве случаев задачи, подобные самоочистке, запускаются по воле пользователя. Это менее болезненно, чем случаи, когда процесс запускается неожиданно, как неизбежное «Это проверка системы экстренного оповещения» Федеральной комиссии связи США, всплывающее, когда вы смотрите телевизор.

Когда пользователи запускают приложение, они обычно держат в голове последовательность шагов, ведущих к какой-то цели. Неожиданные и особенно несвоевременные прерывания, например, со стороны обслуживающей задачи, как показано на рис. 11.2, захватывают приложение и в значительной степени нарушают потоковое состояние пользователя. В представлении пользователя каждая прошедшая секунда, когда он не в состоянии вернуться в потоковое состояние, – это потерянная секунда. Потерянное время воспринимается как текущее более медленно, чем время активности.



*Рис. 11.2. В этом примере программа электронной почты XYZ прерывает свое взаимодействие с пользователем для выполнения обслуживающей задачи*

## Имейте в виду

Помните, что фактическое время, захватываемое процессом, не является причиной воспринимаемого увеличения длительности. Решение может само себя заблокировать для выполнения какой-то задачи (такой, как показана на рис. 11.2) на десять секунд, которые могут оказаться мучительными для пользователя. Однако позже пользователь может потратить более 30 с на выбор прилагательного для использования в электронном сообщении. Первое может восприниматься как потеря, а второе нет. Величина интервала времени, определяющая порог, при превышении которого пользователь начинает ощущать потерю времени, на самом деле является субъективной. Снова вспомним проверку Федеральной комиссии связи. Люди раздражались бы меньше, если бы

проверка всегда проходила во время рекламы, потому что ценность времени во время передачи для зрителя выше, чем во время рекламы.

Правило большого пальца заключается в том, что, если длительность процесса превосходит десять секунд, пользователям должна предоставляться возможность остановки процесса или переключения на другие задачи (это описано в главе 4 «Реактивность»). Для процессов, продолжающихся несколько минут, рассмотрите вариант минимизации приложения, чтобы подтолкнуть пользователей к выполнению других задач, при этом обеспечьте уведомление о завершении процесса.

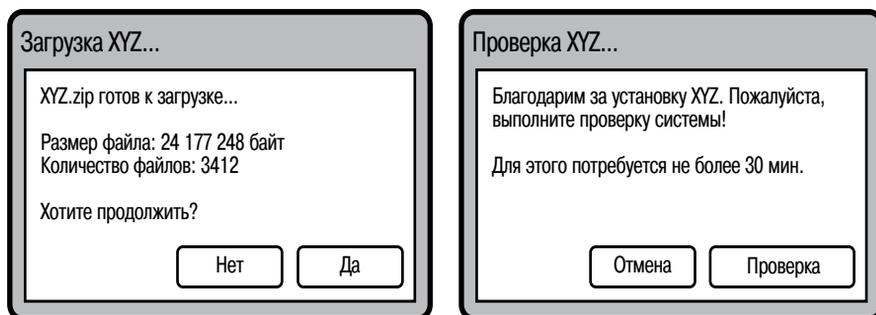
## Негативная оценка

Любые преднамеренные и непреднамеренные виды индикации в UI, которые явно объявляют о том, что процесс будет длительным, оставляют у пользователя впечатление, что процесс отнимает слишком много времени.

### В чем заключается нарушение

Часто мы судим о том, как долго нечто будет длиться, на основании как прямой, так и косвенной индикации в рамках среды. Представьте, например, что вы выезжаете на шоссе и обнаруживаете сильно затрудненное движение. Разумно будет предположить, видя количество машин и их малую скорость, что ваша поездка займет больше времени, чем вы думали.

Аналогично, люди полагаются на прямую индикацию, предоставляемую, к примеру, индикаторами выполнения, и косвенную индикацию в рамках UI для того, чтобы предположить, сколько времени займет нечто, их интересующее. Пользователи хотят получить какую-то оценку



*Рис. 11.3. UI слева использует очень крупные числа для величин, которые пользователь может и не понимать, а UI справа прогнозирует длительность, неоправданно большую для выполнения проверки системы. В обоих случаях пользователя вынуждают предполагать худшее и сомневаться в том, стоит ли вообще продолжать работу*

времени, потому что это позволит им расставлять приоритеты, планировать и уделять внимание другим задачам (проверке наличия новых сообщений, посещению ванной комнаты и т. д.). Наглядная информация в UI является главным и самым прямым видом индикации. Если люди решают, справедливо или нет, что процесс займет слишком много времени (как показано на двух примерах на рис. 11.3), есть риск, что они не захотят даже его запускать.

### **Имейте в виду**

Всегда внимательно определяйте уровень информации, которую пользователь должен видеть. Основной контингент пользователей может быть заинтересован лишь в знании длительности процесса, в то время как опытные пользователи могут желать знать объем дискового пространства или другие потребляемые ресурсы. Если у вас смешанный контингент, рассмотрите применение механизма последовательного раскрытия (это описано в разделе «Единица продвижения выполнения: время или работа» главы 6).

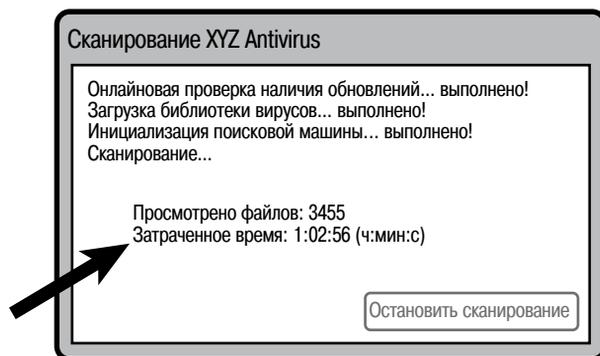
## **Затраченное время**

Любая индикация в UI, тупо сообщающая пользователям, какое время затрачено, действует только как болезненное напоминание о том, сколько времени пропало даром.

### **В чем заключается нарушение**

Подобное нарушению типа «наблюдение за чайником», это нарушение напоминает нам о том, в чем мы обычно не нуждаемся и на что не хотим обращать внимание, – о непрерывном исполнении очень протяженных действий. Значение затраченного времени нам не нужно, если только не приходится использовать его для принятия решения. Иногда отслеживание затраченного времени полезно или даже является решающим (например, при приготовлении пищи, в некоторой аппаратуре, при дорогостоящих телефонных звонках и т. д.). Для многих других действий и процессов, в частности в индустрии розничной торговли и обслуживания, информирование о затраченном времени скорее вредно, чем полезно, потому что оно становится постоянным и неприятным напоминанием о том, как долго люди ждут (рис. 11.4).

Аналогично, в человеко-машинном взаимодействии затраченное время является напоминанием о времени, прошедшем безрезультатно, если только оно не должно служить пользователю основанием для выполнения каких-то действий. Как и в случае предыдущего нарушения, пользователям интереснее знать, когда процесс закончится.



*Рис. 11.4. В этом примере UI программы антивирусного сканирования сообщает, сколько времени затрачено. В большинстве случаев эта информация служит только как напоминание пользователю, сколько времени пропало даром*

## Имейте в виду

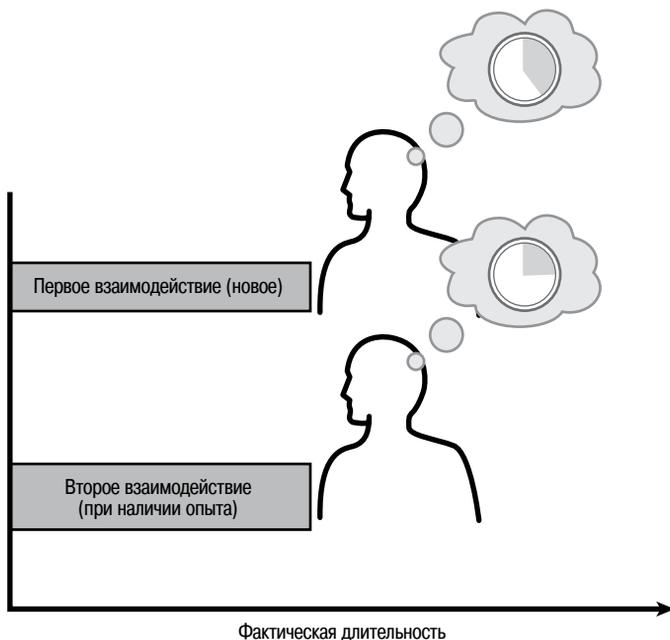
Не используйте индикацию затраченного времени лишь потому, что оставшееся время нельзя определить. Раздел об оставшемся времени в главе 7 «Представление времени» дает рекомендации по этому вопросу. Самый предпочтительный вариант – предоставить пользователям информацию о времени завершения. Вдобавок обдумайте способ представления информации о завершившейся (законченные этапы, просмотренные таблицы и т. д.) или оставшейся работе (этапы, которые предстоит завершить, таблицы, которые надо будет просмотреть, и т. д.). Для получения более подробной информации читайте главы 6 и 7.

## Эффект Барнабуса

Первый опыт выполнения незнакомого процесса, скорее всего, будет казаться более длительным, чем последующие.

### В чем заключается нарушение

Барнабус – это имя лабораторной крысы в Брауновском университете, которая может выполнить последовательность впечатляющих трюков, чтобы получить в награду пищу. Когда короткий видеоролик трюков Барнабуса был показан людям дважды, более 90% зрителей отметили, что второе воспроизведение видео было короче первого. Забудем об уникальности крысиных навыков, важно то, как наш мозг обрабатывает информацию и как мыслительная нагрузка влияет на восприятие времени. Это часто происходит, когда мы вторично едем в какое-то место; мы можем поклясться, что первая поездка была более длительной (рис. 11.5).



*Рис. 11.5. Из-за того что мозг более активен при первом знакомстве с ранее неизвестным процессом, длительность часто представляется большей, чем при вторичном опыте, несмотря на то что объективно длительности равны*

Большинство взаимодействий с компьютером (например, загрузка, поиск, антивирусное сканирование и т. д.) обычно осуществляются часто, тогда как некоторые выполняются всего один или два раза (например, установка, конфигурирование системы и т. д.). Эффект Барнабуса проявляется для первого типа взаимодействий, то есть когда взаимодействие является повторяемым; при этом первый опыт воспринимается как более длительный, чем последующие. Очевидно, наш мозг работает более напряженно, когда мы имеем дело с чем-то новым; когда мы знакомы с предметом, требуется меньше умственных усилий. Это различие в осмыслении двух сеансов работы, вероятно, влияет на наше восприятие времени в этих сеансах.

### **Имейте в виду**

Помните, этот результат возникает только при условии, что пользователи обращают внимание на детали в ходе первого опыта и что имеется два или более опытов для сравнения. Следовательно, возможный способ предотвращения эффекта Барнабуса состоит в исключении возможности сравнения путем отвлечения внимания пользователя от самого процесса. Эффект Барнабуса следует отличать от случаев, когда

первый запуск приложения в самом деле достаточно медленный. Некоторые приложения запускаются медленнее в первый раз, но загружаются значительно быстрее при последующих запусках. В тех случаях, когда происходит вялая «холодная» загрузка, имеет смысл предупредить пользователей, чтобы эта медлительность не оказалась неожиданной, и заверить их в том, что последующие запуски будут происходить быстрее.

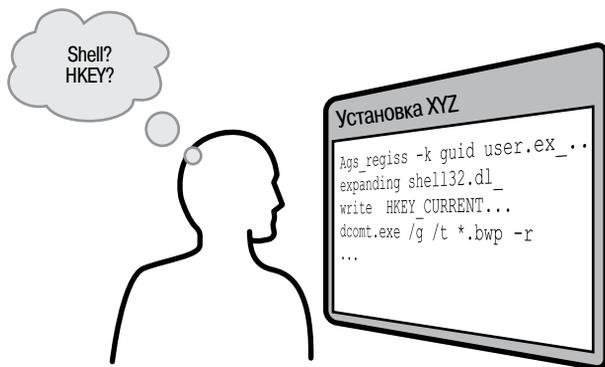
## Перегруженность информацией

Перегруженность пользователей излишним объемом информации в ходе выполнения процесса может вызывать искаженное восприятие длительности.

### В чем заключается нарушение

Перегруженность пользователей излишним объемом информации может привести к тому, что длительность процесса будет казаться большей, чем фактическая. Во-первых, когда люди обращают внимание на информацию о событии, не связанную со временем, восприятие информации, связанной со временем (к примеру, длительности), искажается. И наоборот, когда люди обращают внимание на информацию о событии, связанную со временем, страдает восприятие информации, не связанной со временем (к примеру, подробности, относящиеся к процессу). Этот компромисс описан в теории под названием *гипотеза ослабления*.

Когда UI отображает постоянный поток информации, такой, как показан на рис. 11.6, внимание пользователя естественным образом направлено на него. UI занимает обычного пользователя и уводит внимание от времени, искажая его восприятие. Результатом является оценивание фактической длительности с завышением.



**Рис. 11.6.** В этом примере UI программы установки отображает подробности, бессмысленные для пользователя или ошарашивающие его своим количеством. Если пользователям приходится уделять внимание этим подробностям, восприятие времени обычно искажается

## Имейте в виду

Учтите, что речь идет не о количестве информации самом по себе, а о том, какова значимость информации для пользователя. Одна и та же информация может утомлять того, кто ее не понимает, и не влиять негативно на другого, понимающего ее человека. Следовательно, наполнение длительности информацией может удлинить воспринимаемое время (по сравнению с «пустой» длительностью) или укоротить его в зависимости от природы и значимости информации для пользователя. Относительно каждого блока информации, который вы хотите отобразить в UI, следует задаться вопросом, может ли эта информация быть полезной вашему типичному пользователю.

## Фрагментированные длительности

Прерывания рассекают единый процесс на множество частей, которые, скорее всего, будут восприниматься как более длительные, чем одна, которую запомнили как целостную или состоящую из меньшего числа частей (рис. 11.7).

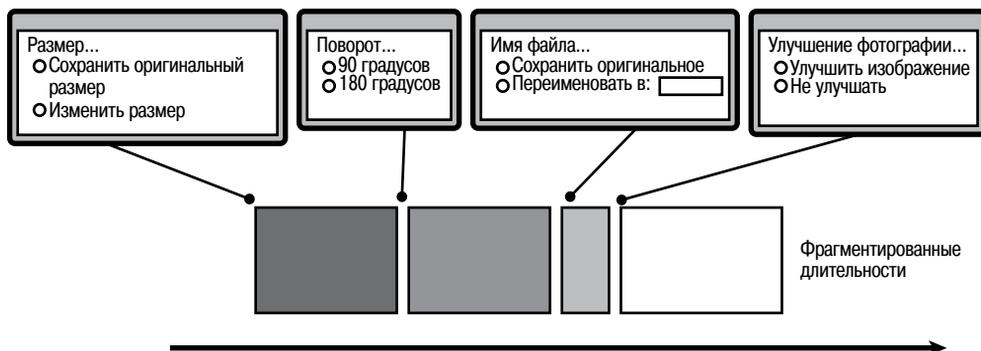


Рис. 11.7. В примере единый процесс разбит на отдельные части перемежающимися запросами пользовательского ввода

## В чем заключается нарушение

Если процесс или действие разделено на части и не хранится в памяти как одно неразрывное событие, люди запоминают множество маленьких частей, составляющих целое. Опасность запоминания отдельных частей – это то, о чем гештальтпсихология говорит следующее: целое больше суммы частей. Продолжая развивать эту идею, гештальтпсихология разъясняет, что наш мозг использует части для формирования восприятия чего-либо в целом и образ этого целого больше, чем реальный объект. Возьмите, к примеру, 30-минутную телепередачу. Хотя мы знаем, что доля рекламы наверняка составляет треть телепередачи

от ее начала до конца, мы воспринимаем телепередачу как получасовую и вряд ли скажем, что нас порадовала любимая 20-минутная программа вчера вечером.

В человеко-машинном взаимодействии, если единый процесс содержит множество контрольных точек, требующих внимания пользователя и обратной связи, этот процесс, по существу, разбивается на множество процессов (в представлении пользователя), у каждого из которых есть своя длительность. Чем больше шагов и контрольных точек запомнил пользователь, тем больше отдельных длительностей складывает человек, чтобы оценить, сколько продолжался процесс.

### **Имейте в виду**

Это нарушение относится к длительным задачам, которые в действительности могут выполняться без перемежающихся прерываний с требованием пользовательского ввода. Это может быть, например, установка программы, длительное сканирование и выполнение поиска. Это нарушение также иногда встречается в работе некоторых затяжных программ-мастеров UI, которые играют для пользователей роль проводников по процессу. В идеале мастера должны провести пользователя через последовательность принятия решений, а затем выполнить необходимые действия как единое целое после сбора всей пользовательской информации.

Дополнительное преимущество организации таких процессов как менее прерывистых заключается в предотвращении неприятного опыта, когда вы запускаете долгий процесс с уверенностью в том, что все будет готово, когда вы вернетесь после перерыва. Увы, вернувшись, вы видите диалоговое окно «Вы действительно хотите продолжить?», ожидающее вашего ответа.

## **Беспокойство**

В состояниях стресса и сильного возбуждения восприятие времени значительно искажается и задержки обычно преувеличиваются.

### **В чем заключается нарушение**

Некоторые американские военнослужащие на Ближнем Востоке имеют портативные устройства голосового перевода, помогающие им преодолеть языковой барьер и общаться с местным населением. Многие приложения и решения считаются критическими или предназначенными для использования в ситуациях, характеризующихся сильным стрессом. Исследования показали, что в состояниях стресса и сильного возбуждения люди склонны неверно оценивать время. Например, при необходимости срочного использования (как в примере на рис. 11.8)



*Рис. 11.8. Представьте себе, что военный находится в ситуации, когда необходимо срочно получить перевод высказывания. После ввода высказывания и щелчка по кнопке Произнести приложение застревает на загрузке языковой базы данных*

или приема некоторой информации каждая отдельная секунда становится необычайно ценной и поэтому каждая секунда задержки кажется текущей очень медленно.

### Имейте в виду

Для приложений, которые могут использоваться в ситуациях, когда пользователь находится в критических условиях или в обстановке, допускающей возможность некоторого стресса, всегда рекомендуется отдать предпочтение простоте в противовес сложности. Другим превосходным примером является приложение, работающее в полицейских машинах. Полицейские должны взаимодействовать с приложением для получения критически важной информации, возможно, в весьма напряженной обстановке, и прерывания по ходу задачи следует минимизировать.

## Нарушение толерантности

Некоторые нарушения не вызывают удлинённого по сравнению с реальным восприятием интервалов времени. Скорее, они вынуждают пользователей терять терпение и толерантность. Как отмечено в главе 2 «Восприятие и толерантность», на толерантность могут влиять факторы, как присущие UI (например, использование неуместной индикации выполнения и т. д.), так и обычно не поддающиеся вашему контролю (например, репутация бренда и т. д.). Этот раздел посвящен тому, что находится под вашим контролем, чтобы вы могли распознать это и корректировать:

- Неопределенность
- Нарушение обещаний

- Обещания компании кабельного телевидения
- Сверхточность
- Повторяющиеся циклы
- Неожиданное дополнение
- Задержка использования

## Неопределенность

Лишение пользователей какой бы то ни было индикации или информации о времени завершения процесса порождает ощущение, что время течет медленнее.

### В чем заключается нарушение

На протяжении всего периода изучения нарушений во множестве направлений исследований неопределенность выступает в качестве самого главного обвиняемого. Не говоря уже о нашей общей нетерпеливости, отсутствие сведений о том, как долго продлится процесс, препятствует нам в реализации многозадачности, в расстановке приоритетов и планировании. В двух примерах, показанных на рис. 11.9, отсутствует индикация того, сколько времени проработает приложение, если его запустить, и, что еще хуже, когда выполнение завершится, если приложение уже запущено. Вынуждать пользователей запускать процесс, не сообщив им, какова будет его продолжительность, так же плохо, как вынуждать кого-то совершать покупку, не сообщив ему цену.



*Рис. 11.9. В примере слева отсутствует индикация того, сколько продлится сканирование для поиска шпионских программ. В примере справа отсутствует индикация того, насколько процесс продвинулся или сколько времени осталось до конца*

### Имейте в виду

Лучшее лекарство от неопределенности – это определенность. Всегда, где только возможно, стремитесь предоставлять информацию, чтобы

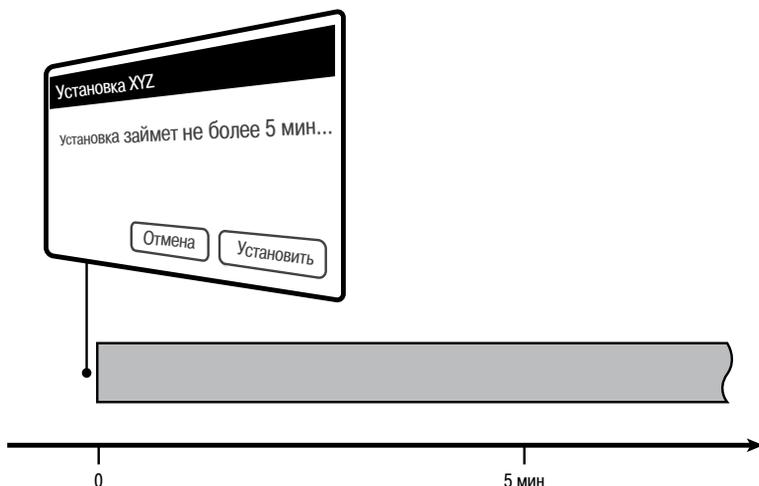
создать у пользователей ощущение определенности. Как отмечено в главе 6, следует рассмотреть возможность использования как единиц времени, так и единиц работы.

## Нарушение обещаний

Если процесс продолжается сверх той длительности, которая была оценена и обещана UI, пользовательская толерантность ухудшается.

### В чем заключается нарушение

Естественно, если процесс продолжается дольше, чем было обещано, толерантность пользователя снижается. Вспомните, что пользователи могут употребить предоставляемую вами в UI информацию для расстановки приоритетов, планирования и реализации многозадачности. Предположим, у пользователя есть ровно пять минут до момента, когда он должен покинуть офис, чтобы успеть на автобус, и установка, например, показанная на рис. 11.10, обещает продлиться не более пяти минут. Пять минут проходят, а она не завершается. Несмотря на то что первые пять минут пользователь вполне мог быть терпеливым, его толерантность резко снижается в последующее время.



*Рис. 11.10. В этом примере UI показывает, что длительность будет менее пяти минут. Однако процесс превысил это время. Толерантность при превышении обещанного времени, разумеется, снизилась*

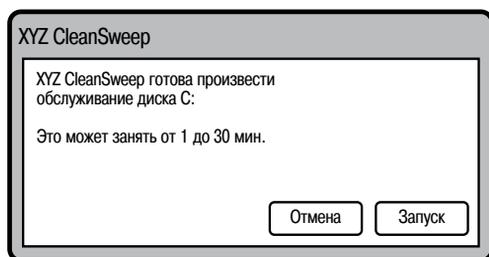
### Имейте в виду

Рассматривайте толерантность как ментальный бюджет, который пользователи составляют, прежде чем приступить к работе или запустить процесс. Когда ваше приложение съедает этот бюджет, вы вступаете на тонкий лед. Поэтому проявляйте осмотрительность, когда даете

обещания в UI, ведь пользователи считают оценки времени точными и используют эту информацию для расстановки приоритетов и реализации многозадачности.

## Обещания компании кабельного телевидения

Если диапазон оценки длительности слишком велик, толерантность снижается уже на этапе, предшествующем началу процесса (рис. 11.11).



*Рис. 11.11. В примере приложение, осуществляющее обслуживание, дает диапазон оценки, слишком широкий для того, чтобы быть полезным пользователю*

### В чем заключается нарушение

Компании кабельного телевидения заработали за это нарушение немало обвинений, но есть и другие поставщики услуг, пользующиеся дурной славой за указание необоснованно широких диапазонов оценок времени. Это нарушение не связано с неопределенностью, потому что налицо некоторый уровень определенности в том, что обещанная услуга будет предоставлена в указанных временных границах. Проблема в самой длине временного диапазона. Чем шире интервал оцениваемого времени, тем больше людей начинают подозревать, что данные оценки объясняются уклончивостью или даже леностью сотрудников службы.

### Имейте в виду

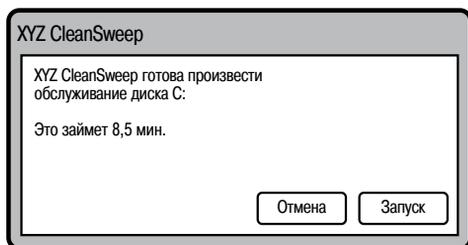
Помните, что компании кабельного телевидения могут выйти сухими из воды с установкой широких диапазонов, потому что потребители в действительности не имеют иного выбора. Представьте другую ситуацию, когда вам нужен водопроводчик. Вероятно, положив трубку, вы будете смеяться, если водопроводчик скажет, что придет между 9 ч утра и 4 ч дня. Когда у пользователей нет выбора, толерантность немного выше (но это не означает, что они не раздосадованы!). Когда выбор есть, толерантность совсем слабая, если вообще присутствует.

## Сверхточность

Слишком высокая точность прогнозирования времени завершения подталкивает пользователей к тому, чтобы подвергнуть ее проверке.

## В чем заключается нарушение

Некоторые центры обработки заказов потребителей дают пользователям автоматизированную оценку того, сколько времени звонящие должны ждать разговора с живым оператором. Какова будет ваша реакция, если автоматизированный голос объявит, что с живым оператором вас свяжут через 3 мин и 35 с? Ничего странного не будет в том, что вы захотите взять в руки секундомер, отсчитывающий заявленное время. Люди тяготеют к особым числам, оценивая время и говоря о нем. (Якоря времени обсуждаются в главе 7.) На примере, показанном на рис. 11.12, UI объявляет, что обслуживание займет ровно 8,5 мин. Использование точных значений оставляет впечатление, что оценки точны, тем самым пользователи приглашаются к тому, чтобы подвергнуть их проверке.



*Рис. 11.12. В этом примере обслуживающее приложение обещает, что процесс продлится ровно 8,5 мин. Объявление точного времени завершения склоняет пользователей к его проверке*

## Имейте в виду

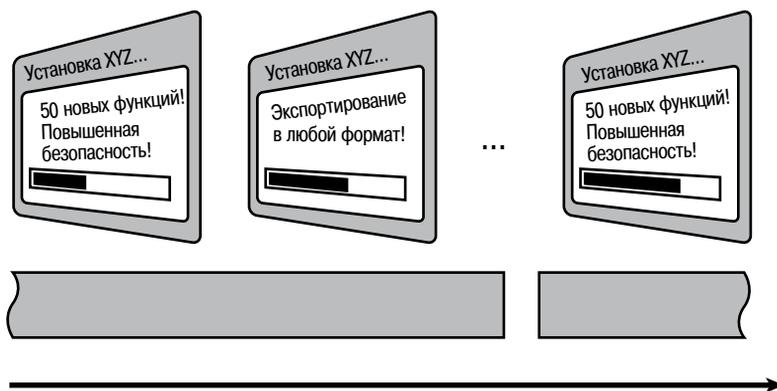
Если пользователь не нуждается в точных значениях длительностей, нет смысла их предоставлять. Даже если оценки верны и вы уверены, что процесс продлится именно столько, в случае, подобном показанному на рис. 11.12, округлите число до следующего большего якоря времени и объявите, что процесс продлится менее этого времени, например «не более 10 мин».

## Повторяющиеся циклы

Распространена практика заполнения длительных процессов, в особенности установок программ, путем отображения в UI последовательности слайдов с заранее подготовленной информацией. Однако, когда последовательность заканчивается и начинается сначала, это снижает пользовательскую толерантность (рис. 11.13).

## В чем заключается нарушение

Всем знакомо состояние ожидания телефонной связи с прослушиванием записи типа «Ваш звонок важен для нас». Вначале это немного ободряет, а затем приходит раздражение от бесконечного повторения одной



*Рис. 11.13. В этом примере во время установки приложения отображается последовательность слайдов, характеризующих новые возможности программы. Однако, когда последовательность заканчивается, на экран возвращается первый слайд*

и той же фразы. То же происходит в кинотеатрах при повторении перед сеансом слайдов с рекламой. Люди начинают использовать заметные повторения как отметки для измерения проходящего времени.

Многие длительные задачи, подобные показанной на рис. 11.13, включают слайд-шоу, которое повторяется до тех пор, пока выполняется задача. Несмотря на то что вначале это является эффективным, потому что отвлекает внимание пользователей от реальной длительности процесса, повторы в действительности снижают толерантность, так же как это делает фраза «Ваш звонок важен для нас».

### **Имейте в виду**

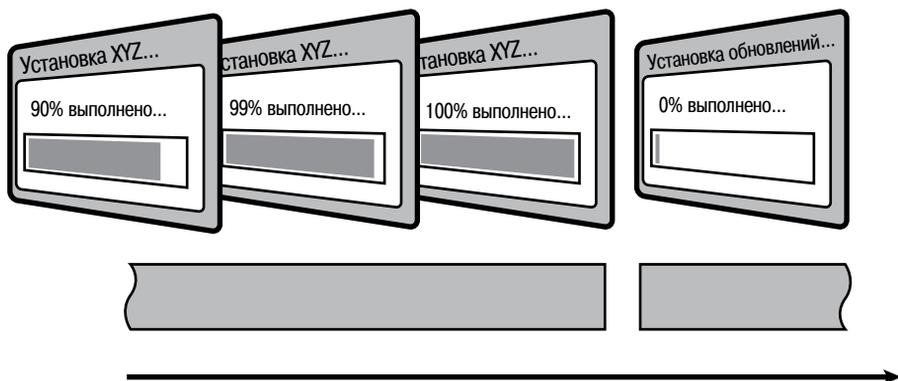
Несмотря на то что решения, подобные слайд-шоу, показанному на рис. 11.13, разработаны ради того, чтобы время проходило быстрее, они могут давать обратный эффект, если информация не значима для пользователя. Отвлечение, как было отмечено в главе 10 «Методы», работает, когда возможно какое-то пользовательское взаимодействие или когда представляемая информация наполнена смыслом для пользователя и важна для него.

## **Неожиданное дополнение**

Если пользователи не предупреждены о процессе, который обязательно последует за выполняемым, их толерантность к неожиданному процессу сильно понижается (рис. 11.14).

### **В чем заключается нарушение**

У знаменитого комика Джерри Зайнфелда есть забавная шутка о зале ожидания в офисе врача: «Они наконец вас вызывают... и вы думаете,



*Рис. 11.14. В этом примере пользователь ожидает, что установка завершится, как только будет выполнено на 100%. К несчастью, имеется дополнительная установка программы обновления, о которой пользователя не оповестили. Толерантность к этому неожиданно возникшему процессу низка*

что увидите доктора, но не видите его, не так ли? Нет. Вы попадаете в следующий зал ожидания – меньший». Люди хотят морально подготовиться к долгому ожиданию, прежде чем оно начнется. Если они удивлены дополнительным временем ожидания, почти всегда это равносильно исчерпанию запаса толерантности.

В среде человеко-машинных взаимодействий яркие примеры такого нарушения обнаруживаются во многих установках приложений, где требуется ряд установок технологий поддержки. Если пользователи не предупреждены о том, что потребуется последовательность установок, они могут предположить, что первая установка является единственно необходимой. Когда неожиданно запускается второй процесс, пользовательская толерантность к этому «сюрпризу» сильно понижается.

### **Имейте в виду**

Испытанный трюк торговцев состоит в том, чтобы пригласить потенциального покупателя войти в магазин и предложить ему нечто мелкое, прежде чем говорить о большой покупке. Некоторые коллеги-преподаватели печально известны тем же – заверениями, что в экзаменационном билете есть лишь пять вопросов, без упоминания того, что каждый вопрос содержит десять частей! Этот принцип никогда не должен использоваться при проектировании программного обеспечения – не следует до поры до времени умышленно скрывать то, что для вашего решения требуются дополнительные процессы. Проявляйте тщательность в подборе того, что необходимо приложению или решению, чтобы оно было функциональным или удобным.

## Задержка использования

Если люди обнаруживают, что нельзя начать пользоваться программой из-за того, что загружены не все ее функции, многие из которых им сейчас и не нужны, то толерантность снижается.

### В чем заключается нарушение

Предположим, что вы подходите к магазину или закусочной за несколько минут до открытия. Вы, конечно, увидите персонал, готовящийся к началу рабочего дня. Если вами владеет страстное желание выпить кофе, то каждая секунда наблюдения за персоналом, который бездельничает, лишая вас возжеленной порции кофеина, поистине мучительна.

Можно найти аналогичный пример в мире компьютерных программ. Многие приложения запускаются в определенной последовательности, и как только все компоненты загружены, приложение можно использовать (рис. 11.15). Чем сложнее приложение, тем длиннее будет эта последовательность. Чем дольше время запуска, тем больше пользователям приходится ждать возможности работы с приложением. Если пользователи видят, что требуемые компоненты уже загружены, но еще недоступны или что загружены и доступны ненужные им компоненты, это оказывает негативное воздействие на толерантность.



*Рис. 11.15. Показанное здесь приложение загружает компоненты, после чего переходит в состояние полной готовности. Это раздражает тех пользователей, которые в действительности хотят выполнить простую задачу, но для получения возможности работать с приложением вынуждены ждать его полной загрузки*

### Имейте в виду

Составьте список функций или задач, которые пользователь может выполнить с помощью приложения, и задайте им приоритеты, соответствующие вероятности их использования (то есть определите, какова

вероятность того, что они понадобятся пользователю в первую очередь). Например, для почтовой программы проверка наличия новых сообщений, создание нового сообщения, поиск существующего сообщения должны иметь в списке высокий приоритет. Постарайтесь загружать в первую очередь компоненты с высоким приоритетом и предоставить пользователям возможность взаимодействовать с приложением, пока подготавливаются части, характеризующиеся низким приоритетом.

## Выводы

Эта глава знакомит вас с пятнадцатью распространенными нарушениями, которые можно предотвратить или исправить. Первые восемь – это нарушения восприятия, потому что они ведут к восприятию фактических длительностей как более продолжительных, чем на самом деле. Остальные семь нарушений являются нарушениями толерантности, потому что они вынуждают пользователей становиться менее терпеливыми, менее снисходительными и менее толерантными к ожиданиям и задержкам.

## Кроличья нора

### Оценивание с занижением и завышением

Goodin, R. E., J. M. Rice, M. Bittman and P. Saunders. The time-pressure illusion: Discretionary time vs. free time. *Social Indicators Research*, 73, 2005. 43–70.

Hancock, P. A. and J. L. Weaver. On time distortion under stress. *Theoretical Issues in Ergonomics Science*, 6, 2005. 193–211.

Loftus, E. F. Time went by so slowly: Overestimation of event duration by males and females. *Applied Cognitive Psychology*, 1, 1987. 3–13.

Mattes, S. and R. Ulrich. Directed attention prolongs the perceived duration of a brief stimulus. *Perception & Psychophysics*, 60, 1998. 1305–1317.

Roy, M. M., N. J. S. Christenfeld and C. R. M. McKenzie. Underestimating the duration of future events memory incorrectly used or memory bias. *Psychological Bulletin*, 131, 2005. 738–756.

Yarmey, A. D. Retrospective duration estimations for variant and invariant events in field situations. *Applied Cognitive Psychology*, 14, 2000. 45–57.

## Ожидание и удовлетворенность потребителей

Antonide, G., P. C. Verhoef and M. van Aalst. Consumer perception and evaluation of waiting time: A field experiment. *Journal of Consumer Psychology*, 12, 2002. 193–202.

David, M. M. and J. Heineke. How disconfirmation, perception and actual waiting times impact customer satisfaction. *International Journal of Service Industry Management*, 9, 1998. 64–73.

Houston, M. B., L. A. Bettencourt and S. Wenger. The relationship between waiting in a service queue. *Psychology & Marketing*, 15, 1998. 735–753.

Pruyn, A. and A. Smidts. Customers' reactions to waiting: Effects of the presence of 'fellow sufferers' in the waiting room. *Advances in Consumer Research*, 26, 1999. 211–216.

Unzicker, D. K. The psychology of being put on hold: An exploratory study of service quality. *Psychology & Marketing*, 16, 1999. 327–350.



# Алфавитный указатель

## В

Bringing Design to Software, 45

## С

CRC (контроль с помощью циклического избыточного кода), 116

## D

D-уровни (обнаружение различий во времени), 89

D0-сценарии (нулевое количество длительностей), 89

D2-сценарии (две длительности), 89

Dealing with Darwin, 95

## E

ESD-TR-86-278 (стандарты на время отклика), 67

## F

Flow: The Psychology of Optimal Experience (М. Чиксентмихайи), 144

## J

jnd, наименьшее заметное различие (Just Noticeable Difference), 91

## M

MIL-STD 1472F (стандарты на время отклика системы), 65

## N

Nielsen Norman Group, 26

## T

TAM (Technology Acceptance Model), 28

## W

WOMM, глашатаи рынка (word-of-mouth marketing), 126

## A

анимирование переходов, 83

аппаратное обеспечение и время отклика системы, 64

## Б

беспокойство (нарушение восприятия), 205

буферизация и предложение (управление толерантностью), 188

## В

Вебер, Э. Г. (Weber, E. H.), 90

Вергеймер, М. (Wertheimer, M.), 84

вид отображения, 107

визуальные объекты (индикация выполнения), 109

влияние протяженности действия

и тренированности пользователя, 166

внесение определенности, 134  
 воспринимаемая длительность, 35, 43  
 восприятие, 28  
     времени, 27  
     производительности, 28  
 впечатление (потокосное состояние  
 пользователя), 142  
 временные константы, 74  
     воспринимаемого времени  
     обработки, 74  
     времени элементарной задачи, 74, 80  
     немедленного времени  
     отклика, 74  
 временные ракурсы, 122  
     в режиме реального времени, 124  
     предварительный, 122  
     ретроспективный, 126  
 время  
     в академических дисциплинах, 23  
     изменение величины, 27  
     планирование, 27  
     проектирование  
         примеры плохого  
         проектирования, 22  
         причины для, 25  
     субъективное ощущение, 24  
     шкалы, 23  
 время дня как фактор толерантности, 49  
 время ожидания, завышение, 43  
 время отклика, 55  
     максимально приемлемое, 80  
     пользователя, 57  
     системы, 58, 62  
     слишком быстрые отклики, 81  
 время реакции, 22  
     выбора, 23, 59  
     простой, 59  
 выбор  
     единицы продвижения (индикация  
     выполнения), 111  
     индикации выполнения, 107  
     класса индикации выполнения, 105  
 вынуждающая реактивность, 80  
 выражение времени, 127

## Г

гештальтпсихология, 204  
 гипотеза ослабления, 203  
 границы, оценивание времени, 131

## Д

двойная лакмусовая проверка, 102  
 Дворак, Дж. К. (Dvorak, J. C.), 113  
 деление надвое, 90  
     посредством среднего  
     геометрического, 97  
 день недели как фактор  
 толерантности, 49  
 диалоги как человеко-машинные  
 взаимодействия, 55  
 диапазоны, оценивание времени, 130  
 динамическая индикация  
 выполнения, 102  
 дифференциация, нейтрализация, 89, 97  
 длительность  
     D0-сценарии (нулевое количество  
     длительностей), 89  
     D2-сценарии (две длительности), 89  
     непрерывные длительности  
     (управление восприятием), 178  
     нисходящие длительности  
     (управление восприятием), 175  
     оценивание длительностей  
     в управлении толерантностью, 183  
     широкий диапазон оценки  
     (нарушение толерантности), 209  
 Дондерс, Ф. К. (Donders, F.C.), 22  
 допустимая регрессия, 89, 94  
 достоверность полученной  
 информации, 154  
 доступность, 148  
 дробь Вебера, 91

## Е

единицы времени (индикация  
 выполнения), 112, 135, 136  
 единицы продвижения (индикация  
 выполнения), выбор, 111  
 единицы работы (индикация  
 выполнения), 115  
 единственное число для единиц  
 времени, 135

## З

задержка использования (нарушение  
 толерантности), 213  
 закон  
     Вебера, 90

## закон

- Вебера–Фехнера, 90, 131
- влияния, 126
- Фирордта, 43
- Хика–Хаймана, 61
- запустить-и-забыть (управление восприятием), 181
- затраченное время, представление, 111, 112, 125

**И**

- извлечение (память), 41
- изменение величины времени, 27
- измерение воспринимаемых длительностей, 160
- иллюзия движения, 84
- иллюзия наполненной длительности, 182
- имена торговых марок продуктов как фактор толерантности, 50
- индикация выполнения, 101, 102, 134
  - визуальные объекты, 109
  - выбор, 107
  - выбор типа данных, 116
  - единицы времени, 112
  - единицы работы, 115
  - класса А, 104
  - класса В, 104
  - класса С, 104
  - класса D, 104
  - основанная на тексте, 107
  - проектирование, 107
- индикация затраченного времени (нарушение восприятия), 200
- индикация как фактор толерантности, 47
- интериндивидуальный подход, 166
- интерпретации задержек, 73
- интраиндивидуальный подход, 166
- информирование (управление восприятием), 179
- искажения
  - времени, типы, 42
  - длительности, типы, 42
  - памяти, 41
- использование как фактор толерантности, 46
- использование примеров, 142

**К**

- Кард, С. (Card, S.), 74
- карта потокового состояния пользователя, 140
- качественные данные (индикация выполнения), 117
- классификация индикации выполнения, 102
- когнитивный контроль, 142
- кодирование (память), 41
- количественные данные (индикация выполнения), 116
- комикс Симпсоны, 56
- компромисс обучение/удобство использования, 26
- конструирование времени, причины для, 25
- конструктор времени, 29
- контекстные контрольные данные (управление толерантностью), 190
- контроль с помощью циклического избыточного кода (CRC), 116
- корпорация MITRE, нормативы на время отклика системы, 66
- кривая обучения, 23
- критерий типа ненамного, 96
- кроссмодальное сравнение, оценивание толерантности, 165
- культура как фактор толерантности, 50

**М**

- максимально приемлемое время отклика, 80
- матрица якорей времени, 129
- мгновенная реактивность, 75
- методы замеров реальных длительностей, выбор, 159
- микровыражения, 73
- Миллер, Р. (Miller, R.), 64, 80
- множественное число для единиц времени, 135
- мода как фактор толерантности, 50
- модель восприятия технологии, 28
- модель шкалы цен (управление толерантностью), 184
- Мур, Дж. (Moore, G.), 95
- мысленный эталон, 43
- мягкость пользователя как фактор толерантности, 48

## Н

надежность полученной информации, 154  
 нарушение восприятия, 195  
   беспокойство, 205  
   индикация затраченного времени, 200  
   наблюдение за чайником, 196  
   негативная оценка, 199  
   не отпускающее ожидание, 197  
   перегруженность информацией, 203  
   фрагментированные длительности, 204  
   эффект Барнабуса, 201  
 нарушение толерантности, 206  
   задержка использования, 213  
   нарушение обещаний, 208  
   неожиданное дополнение, 211  
   неопределенность, 207  
   повторяющиеся циклы, 210  
   сверхточность, 209  
   широкий диапазон оценки длительности, 209  
 настройка, измерение воспринимаемой длительности, 162  
 негативная оценка (нарушение восприятия), 199  
 незамедлительная реактивность, 76  
 неисключительная реактивность, 74  
 нейтрализация дифференциации, 89, 97  
 нелинейная индикация выполнения (управление восприятием), 176  
 неоднократные сбои как фактор толерантности, 47  
 неожиданное дополнение (нарушение толерантности), 211  
 неопределенная индикация выполнения, 102  
 неопределенность (нарушение толерантности), 134, 207  
 не отпускающее ожидание (нарушение восприятия), 197  
 не прерывающая реактивность, 78  
 непрерывные длительности (управление восприятием), 178  
 Нильсен, Я. (Nielsen, J.), 74  
 нисходящие длительности (управление восприятием), 175

## О

объективное оценивание времени, 38  
 объекты сравнения как фактор толерантности, 46  
 объем, получение объективной информации, 39  
 объем внимания, 110  
   золотых рыбок, 80  
 однократная задержка (управление восприятием), 189  
 ожидаемая реактивность, оценивание толерантности, 163  
 ожидание, 43  
 оперативное управление, 36  
 определенная индикация выполнения, 102  
 оптимизация потокового состояния пользователя, 143  
   ощущение контроля, 149  
   соответствие между умениями и сложностью задачи, 144  
 опыт  
   время отклика пользователя, 84  
   как фактор толерантности, 46  
 оставшееся время  
   оценивание времени, 132  
   представление, 112  
 относительная природа реактивности, 72  
 отношение Вебера, 91  
 отображение  
   завершения процесса, 135  
   содержательной информации, 134  
 отсчет последних секунд (управление толерантностью), 191  
 оценивание времени, 127  
 оценивание длительностей, 43, 129  
   в управлении толерантностью, 183  
   границы, 131  
   диапазоны, 130  
 оценивание длительностей, 43, 129  
   оставшееся время, 132  
   фрагментированные длительности (нарушение восприятия), 204  
   широкий диапазон оценки (нарушение толерантности), 209  
   якоря времени, 128  
 оценивание толерантности, 163  
 кроссmodalное сравнение, 165

ожидаемая реактивность, 163  
продуцирование, 165  
экспериментирование, 164  
ощущение контроля, в оптимизации  
потокowego состояния  
пользователя, 149

## П

первый закон сервиса Майстера, 29  
перегруженность информацией  
(нарушение восприятия), 203  
планирование времени, 27  
планирование эксперимента  
влияние протяженности действия  
и тренированности  
пользователя, 166  
повторяющиеся попытки как фактор  
толерантности, 47  
повторяющиеся циклы (нарушение  
толерантности), 210  
показатели как фактор  
толерантности, 46  
показатели реактивности, 74  
получение информации, 154  
влияние протяженности действия  
и тренированности  
пользователя, 166  
для оценивания толерантности, 163  
кроссмодальное сравнение, 165  
ожидаемая реактивность, 163  
продуцирование, 165  
экспериментирование, 164  
достоверность, 154  
надежность, 154  
о воспринимаемых  
длительностях, 160  
настройка, 162  
репродуцирование событий, 162  
словесное оценивание, 160  
о реальных длительностях, 156  
определение, 157  
пользовательские оценки, 160  
пользовательские оценки, 166  
пользовательская навигация, 141  
пользовательская толерантность, 19,  
35, 37  
пользовательские оценки  
измерение воспринимаемых

длительностей, 160  
настройка, 162  
репродуцирование событий, 162  
словесные оценки, 160  
измерение реальных  
длительностей, 160  
при сборе информации, 166  
порог  
негодования, 45  
толерантности, 45  
последовательное раскрытие, 115  
поток  
взаимодействия, 141  
деятельности, 141  
заданий, 141  
пользовательского интерфейса, 141  
потокое состояние пользователя, 139  
в виде карты, 140  
в виде траектории, 141  
восприятие и толерантность, 139  
впечатление, 142  
оптимизация, 143  
ощущение контроля, 149  
соответствие между умениями  
и навыками и сложностью  
задачи, 144  
потокые каналы, 144  
правило 20%, 92  
предварительный временной ракурс, 122  
предлог между в представлении  
времени, 136  
предметы для сравнения, 44  
представление времени, 121  
предубеждение как фактор  
толерантности, 50  
пример  
Костко, 18  
с двойным щелчком (слишком  
быстрый отклик системы), 82  
с задержками клавиатуры, 62  
с компаниями кабельного телевидения  
(нарушение толерантности), 209  
удобства использования  
внутрикорпоративной сети, 26  
принцип лезвия Оккама, 73  
прихоти как фактор толерантности, 50  
программное обеспечение и время  
отклика системы, 64

продуцирование, оценивание  
толерантности, 165  
проектирование и время отклика  
системы, 64  
производительность, восприятие, 29  
психологические ограничения для  
времени отклика системы, 64  
психологическое время, 40  
психология, изучение времени, 22

**Р**

работа с архивными документами, 142  
ракурс в режиме реального времени, 124  
раннее завершение (управление  
восприятием), 173  
Рао, Р. (Rao, R.), 83  
распределение времени, 56, 121  
реактивность, 71  
характеристики, 71  
реальные длительности, замеры, 156, 157  
репродуцирование событий, измерение  
воспринимаемых длительностей, 162  
репутация продуктов как фактор  
толерантности, 50  
ретроспективный временной ракурс, 126  
решение, 31

**С**

Саффо, П. (Saffo, P.), 45  
сверхточность (нарушение  
толерантности), 209  
Селкер, Т. (Selker, T.), 80  
скрытая декомпоновка (управление  
восприятием), 174  
слишком быстрые отклики, 81  
словесные оценки, измерение  
воспринимаемых длительностей, 160  
соответствие между умениями  
и навыками и сложностью задачи  
(оптимизация потокового состояния  
пользователя), 144  
соотношение скорость/точность, 60, 84  
стандарты как фактор толерантности, 46  
статическая индикация выполнения, 103  
степенной закон научения, 60, 84  
субъективное ощущение задержек, 24

**Т**

тайм-ауты, 134  
таймеры, 134  
теплые перезагрузки, 189  
тип данных (индикация выполнения),  
выбор, 116  
толерантность, 139, 163  
точность для реальных  
длительностей, 157  
траектория потокового состояния  
пользователя, 141  
тревога, 144

**У**

удобство использования, 26  
управление  
восприятием, 36, 171  
запустить-и-забыть, 181  
информирование, 179  
нелинейная индикация  
выполнения, 176  
непрерывные длительности, 178  
нисходящие длительности, 175  
раннее завершение, 173  
скрытая декомпоновка, 174  
упреждающий запуск, 172  
целенаправленное отвлечение, 180  
дифференциацией, 98  
толерантностью, 37, 183  
буферизация и предложение, 188  
контекстные контрольные  
данные, 190  
модель шкалы цен, 184  
однократная задержка, 189  
отсчет последних секунд, 191  
оценивание длительностей, 183  
ценность ожидания, 186  
якоря времени, 185  
упреждающий запуск (управление  
восприятием), 172

**Ф**

фактическая длительность, 35  
факторы толерантности, 45  
вневременные, 47  
временные, 45

Фехнер, Г. Т. (Fechner, G.T.), 90  
фи-феномен, 84  
физиологические ограничения для  
времени отклика системы, 64  
философия, изучение времени, 22  
фрагментированные длительности  
(нарушение восприятия), 204

## Х

холодная загрузка, 189  
хранение (память), 41

## Ц

целевая производительность, задание, 93  
целенаправленное отвлечение  
(управление восприятием), 180  
ценность ожидания (управление  
толерантностью), 186

## Ч

человеко-машинное  
взаимодействие, 17, 18  
временные константы, 74  
время отклика, 55  
диалоговая природа, 55  
Черч, Р. (Church, R.), 97

## Ш

широкий диапазон оценки длительности  
(нарушение толерантности), 209  
шкалы времени, 23

## Э

Эббингауз, Г. (Ebbinghaus, H.), 23  
Экман, П. (Ekman, P.), 73  
экспериментирование, оценивание  
толерантности, 164  
эксперименты с памятью, 23  
эмоциональное состояние как фактор  
толерантности, 49  
эталон, 44  
эффект Барнабуса, 166  
нарушение восприятия, 201  
эффект ореола как фактор  
толерантности, 50

## Я

язык тела как реакция, 73  
якоря времени, 128  
в управлении толерантностью, 185  
границы, 131  
диапазоны, 130  
оставшееся время, 132

