

Дэвид Трамм



СОФТ ОТСТОИ! И ЧТО С ЭТИМ ДЕЛАТЬ?

По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 5-93286-097-9, название «Софт – отстой! И что с этим делать?» – 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), где именно Вы получили данный файл.

Why Software Sucks...

and what you can do about it

David S. Platt

◆ Addison-Wesley

ПРОФЕ  ИОНАЛЬНО

Софт – отстой!

И что с этим делать?

Дэвид С. Платт



*Санкт-Петербург – Москва
2008*

Серия «Профессионально»

Дэвид С. Платт

Софт – отстой! И что с этим делать?

Перевод С. Маккавеева

Главный редактор
Зав. редакцией
Редактор
Художник
Корректор
Верстка

*А. Галунов
Н. Макарова
В. Овчинников
В. Гренда
О. Макарова
О. Макарова*

Платт Д.

Софт – отстой! И что с этим делать? – Пер. с англ. – СПб: Символ-Плюс, 2007. – 248 с., ил.

ISBN-10: 5-93286-097-9

ISBN-13: 978-5-93286-097-7

Дэвид Платт, за плечами которого 20-летний опыт программирования и преподавания, утверждает, что современное ПО – отстой. Оно не защищено и позволяет программам злоумышленников проникать из Интернета в наши компьютеры. Оно ненадежно и ломается в самый ответственный момент, уничтожая плоды наших долгих трудов и не давая средств к их спасению. Им трудно пользоваться, потому что приходится ломать голову над тем, как выполнить простейшие операции. Дэвид Платт объясняет, почему программы могут так разочаровывать и даже оказываться опасными, а также предлагает несколько способов борьбы с этим. Изложение содержит много примеров и сдобрено юмором. Не многие компьютерные книжки способны заставить громко смеяться. Но Дэв не только смешит, он делится очень интересными наблюдениями и взглядами, излагая их в ясном и занимательном стиле. Он призывает пользователей сообща бороться с производителями плохого ПО и приглашает единомышленников на свой сайт www.suckbusters.com.

ISBN-10: 5-93286-097-9

ISBN-13: 978-5-93286-097-7

ISBN 0-321-46675-6 (англ)

© Издательство Символ-Плюс, 2007

Authorized translation of the English edition © 2006 Pearson Education Inc. 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) 324-5353, edit@symbol.ru. Лицензия ЛП N 000054 от 25.12.98.

Налоговая льгота – общероссийский классификатор продукции
ОК 005-93, том 2; 953000 – книги и брошюры.

Подписано в печать 22.11.2007. Формат 70x90¹/₁₆. Печать офсетная.

Объем 15,5 печ. л. Тираж 2000 экз. Заказ N

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

*Моим дочерям Люси Катрине Платт
и Аннабел Роуз Платт*



Оглавление

Похвальные отзывы	10
Об авторе	12
Благодарности	13
Введение	15
<i>Глава 1 Кого считать идиотами?</i>	23
Что было раньше	24
Почему и по сей день программы — отстой	25
Контроль или легкость эксплуатации?	27
Меня не интересует, как работает ваша программа	29
Плохая функция и хорошая функция	33
Прекращение работы и демонстрация идиотизма	37
Испытание на подопытных животных	39
Какова ситуация и что можно сделать	42
<i>Глава 2 Запутавшись в паутине</i>	44
С чего это началось	45
Как это действует	47
Почему с отстоем до сих пор не покончено	50
Проект, ориентированный на клиента, против проекта, ориентированного на сервер	53
Кто поднимет мне веки?	59
Очевидно, что — нет!	65
Заставки, Flash и анимация	68

Тестирование на живых организмах	71
Что мы можем сделать	73
<i>Глава 3 Защитите меня</i>	<i>77</i>
Как было раньше	78
Почему с отстоем до сих пор не покончено	79
Что должны были бы знать программисты	83
Человеческий фактор	88
Бюджет неудобств	91
Пользователь ленив	95
Психологическая атака	98
Заключительное слово о безопасности	103
Что мы можем сделать	104
<i>Глава 4 Ты кто такой?</i>	<i>107</i>
С чего это началось	107
Почему с отстоем до сих пор не покончено	108
Несовместимые требования	109
Ладно, делать-то что?	116
<i>Глава 5 Ну, что смотришь?</i>	<i>129</i>
Да, им известно, кто вы такой	129
Отстой крепчает беспримерно	132
Пользователи не знают, в чем состоят риски	135
Что становится известным прежде всего	137
Доить клиентов с помощью cookies?	139
Чепуха заявлений о неразглашении информации	147
Заметая следы	149
Загадка Google	150
Решение	154
<i>Глава 6 Десять тысяч гиков, помешавшихся на джолт-коле</i>	<i>157</i>
Посмотрите на них в их естественной среде обитания	157
Все эти гики	158
Кто говорит, когда и о чем	161
Это ярмарка	165
Гики, поколение «Next» — передача эстафеты	169

<i>Глава 7</i>	<i>Так кто они такие, эти ненормальные?</i>	175
	Номо Logicus.	176
	Отравление тестостероном	176
	Контроль и удовлетворенность	178
	Сотворение моделей	180
	Гики и качки	182
	Жаргон	184
	Мозги и препятствия.	186
	Семь привычек гиков	188
<i>Глава 8</i>	<i>Microsoft: ни с ней жить нельзя, ни без нее</i>	193
	Они правят миром.	193
	Я и они	194
	Что было раньше	197
	Почему отстой продолжается.	198
	Куда ни кинь — везде клин	202
	Мы любим их ненавидеть	206
	Plus ça change	210
	Трудности роста	214
	Что мы можем сделать	217
	В заключение	223
<i>Глава 9</i>	<i>С этим надо что-то делать</i>	225
	1. Купите	226
	2. Расскажите	231
	3. Высмейте	233
	4. Доверяйте	235
	5. Сплачивайтесь	238
	Эпилог	241



ПОХВАЛЬНЫЕ ОТЗЫВЫ

«Я только что закончил читать лучшую компьютерную книжку [«Why Software Sucks»] и хочу сказать о ней доброе слово. Включите ее в свой список обязательного чтения, если у вас есть программы, если вы любите программы, не любите программистов или даже сами программист, потому что мистер Платт (который преподает программирование) вознамерился сбить спесь с тех, кто считает, что если он может написать программе, то может сделать ее простой в обращении... Книга смешная, но она также служит важным предупреждением тем софтверным компаниям, которые намереваются сократить свои расходы на поддержку клиентов. Если вас мучает вопрос «Почему хорошие программисты пишут такие ужасные программы?», то ответ на него вы найдете в этой книге.»

— Джон Мак-Кормик (*John McCormick*),
ведущий колонки *Locksmith* на *TechRepublic.com*

«Скажу сразу, не многие компьютерные книжки способны заставить меня громко смеяться. Но Дэйв Платт не только смешит, он также делится некоторыми очень интересными наблюдениями и взглядами, излагая их в ясном и занимательном стиле. Это тоже встречается не часто!»

— Генри Лейтнер (*Henry Leitner*), зам. декана
по информационной технологии и старший преподаватель
вычислительных наук, Гарвардский университет

«Книжка для всех угнетенных пользователей компьютеров, написанная понятным языком.»

— *Стейси Барателли, парикмахер автора*

«Уникальный подход Дэвида к трепещущим проблемам разработки ПО заставил меня по-новому взглянуть на этот процесс. Если вас заботит качество программ, которыми вы пользуетесь или которые разрабатываете, прочтите эту книгу.»

— *Дэйв Чэппелл (Dave Chappell), глава Chappell & Associates*

«Я начал читать эту книгу на работе, но прекратил, не дочитав до конца первой страницы. Я не в силах был спрятать ухмылку на лице! Наслажусь ей, когда вернусь домой и найду укромное местечко для чтения.»

— *Цукаса Макино (Tsukasa Makino), ИТ-менеджер*

«Дэвид объясняет (причем так, что его понимает даже моя теща), почему программы, которыми мы сегодня пользуемся, могут так разочаровывать и даже иногда оказываться опасными, а также предлагает несколько способов борьбы с этим.»

— *Джим Броссо (Jim Brosseau), Clarrus Consulting Group, Inc.*



Об авторе

Дэвид С. Платт руководит Rolling Thunder Computing (www.rollthunder.com), учебно-консультационной фирмой. Имея более чем двадцатилетний опыт работы программистом, он преподает разработку программ в Колледже непрерывного обучения Гарвардского Университета и в различных компаниях по всему миру, часто выступает на конференциях. Он автор девяти книг, в том числе «Introducing Microsoft .NET, Third Edition»¹, «The Microsoft Platform Ahead»² и «Understanding COM+» (все вышли в Microsoft Press), а также многочисленных журнальных статей и бюллетеней. В 2002 году Microsoft присудила ему звание «Легенды программирования» (Software Legend). Живет в Ипсуиче, шт. Массачусетс.

¹ Дэвид С. Платт «Знакомство с Microsoft .NET». — Пер. с англ. — Русская Редакция, 2001.

² Дэвид С. Платт «Платформа Microsoft 2005». — Пер. с англ. — Русская Редакция, 2004.



Благодарности

Всякая книга — результат коллективного труда. При работе над этой у меня была превосходная группа поддержки. В Addison-Wesley все поняли и поддержали идею книги, написанной для пользователей компьютеров, а не для программистов, как обычно. Вместо «это не наш профиль» прозвучало «здорово, это для нас отличная возможность».

В первую очередь благодарю моего редактора Питера Гордона (Peter Gordon), который после первого прочтения заметил, что ему не часто попадают рукописи по компьютерам, которые заставляли бы громко смеяться. Спасибо и другим сотрудникам Addison-Wesley, в число которых входят Курт Джонсон (Curt Johnson), Ким Бедигхаймер (Kim Voedigheimer), Джули Нагил (Julie Nahil), Одри Дойл (Audrey Doyle), Анна Попик (Anna Popick) и Эрик Гэрулей (Eric Garulay). Я благодарю своего друга и коллегу-автора Дэвида Чэппелла (David Chappell) в первую очередь за то, что он направил меня в Addison-Wesley. Хочу также поблагодарить своего агента Алекса Гласса (Alex Glass) из Trident Media Group и автора Кэтрин Коултер (Catherine Coulter), направившую меня к нему.

Написанное здесь было вдохновлено чтением книги «Complications: a Surgeon's Notes on an Imperfect Science» (Осложнения: заметки хирурга о несовершенстве науки) Атула Гаванди (Atul Gawande), представляющей собой взгляд хирурга на его собственную профессию. Я подумал: «А почему бы мне не написать подобную книгу о моей профессии, разработке программного обеспечения? У меня получится веселее». В его книге

есть глава «9000 хирургов», в которой рассказывается о поведении хирургов на их ежегодной конференции и которая послужила непосредственным источником моей главы «Десять тысяч гиков, помешанных на джолт-коле».

Я должен поблагодарить всех своих друзей, коллег, клиентов и студентов — тех, кто прочел первые наброски и высказал ценное для меня мнение. Я многому учусь, работая с вами. И всех разработчиков, которые пишут хорошие программы, благодаря кому я могу показать что-то со словами: «Вот правильная вещь, вот что мы можем на самом деле». И всем разработчикам отстойного ПО тоже спасибо, потому что без них мне не над чем было бы смеяться, и книга получилась бы пресной.

Наконец, хочу поблагодарить свою семью: жену Линду и дочерей Аннабель Роуз и Люси Катрину.



Введение

Современное программное обеспечение — отстой. Приличных слов, чтобы выразить этот факт, нет. Оно не защищено и позволяет программам-преступникам проникать из Интернета в наши жилища. Оно ненадежно и ломается в самый ответственный момент, уничтожая плоды наших многочасовых трудов и не давая средств к их спасению. Работать с ним трудно, потому что приходится ломать голову над тем, как выполнить простейшие операции.

Наверно, вы не впервые это слышите. Вы же не идиот, хотя небезызвестная серия книг с желто-черной обложкой ставит перед собой цель убедить вас в обратном. Нынешнее программное обеспечение отвратительно, и вы всегда это знали. Поэтому вы и рассмеялись, увидев название книги. И ведь этот смех не был произвольным, правда? Вы же не сказали себе, прочтя название: «Ха, забавно, надо бы посмеяться»? Нет. Заглавие ударило по нервам, как запах, посылающий сигнал через кору к рептильному мозгу, и вы разразились произвольным безудержным смехом. Когда к вам возвратилось сознание, вы, возможно, подумали что-то вроде «Здорово! Этот малый (умеющий хорошо выражать свои мысли, с рекомендациями, красивый и, судя по всему, к тому же скромный) говорит дело, выражая на бумаге то, что я всегда думал. Притом вовремя». Так оно и есть. В этой книге посредством нетехнического, свободного от жаргона языка рассказывается, почему возникла такая ситуация и можно ли ее как-то изменить. «Черт возьми, надо купить!

И несколько экземпляров для друзей». Отличная мысль. Разве я не сказал, что вы не идиот?

Пятнадцать, даже десять лет назад люди не пользовались программами в повседневной жизни. Мои родители не посылали электронные поздравительные открытки к дням рождения своих внучек и не рассчитывали получить от меня таким же способом цифровые фотографии. Они подводили сальдо своих чековых счетов в бумажной тетради и вели расписание своих дел чернилами в бумажных календарях с милыми картинками. Те немногие, кто регулярно работал с программами, обычно вынуждены были это делать по роду своей работы, например туристические агенты пользовались системами бронирования авиабилетов. У них были маленькие специализированные приложения, для которых требовались дорогое фирменное оборудование, продолжительное обучение и текущая поддержка, а какое-либо применение их для других задач было невозможно. Всемирная паутина World Wide Web была технически сложным изобретением кабинетных ученых. Большинству даже не было известно о ее существовании, не говоря уже о возможности красть с ее помощью музыку или загружать непристойные картинки.

Все совершенно изменилось — по меркам общественного развития практически за ночь, и довольно незаметным для нас образом. Мои родители теперь пользуются для управления чековыми счетами финансовым пакетом, который оплачивает счета электронным способом (поэтому у них никогда не кончаются почтовые марки) и автоматически показывает электронные вклады и оплаченные чеки, когда они поступают в банк. Они теперь хотят не только получать по электронной почте фотографии внучек, но и смотреть живое потоковое видео. Предоставив домашним пользователям неограниченный высокоскоростной доступ примерно за \$40 в месяц, Web сделала то, что обещали, но не осуществили поборники атомной энергии — сделать ее слишком дешевой, чтобы стоило мерить количество. Интернет проник всюду настолько глубоко, что штат Пенсильвания убрал с автомобильных номеров свое прозвище «Штат — Краеугольный Камень» (The Keystone State) и заменил его веб-адресом штата *www.state.pa.us* (рис. 1). Флорида пошла дальше Пенсильвании, сохранив прозвище и заменив название штата веб-адресом *MyFlorida.com* (рис. 2). А потребность в туристических агентах сильно сократилась, поскольку у обычного человека есть дешевый и простой доступ к информации, который раньше требовал дорогого и сложного оборудования.



Рис. 1. Автомобильный номер штата Пенсильвания с веб-адресом вместо прозвища штата

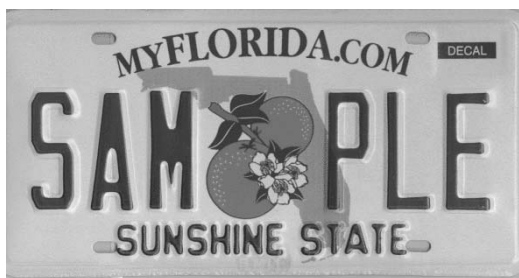
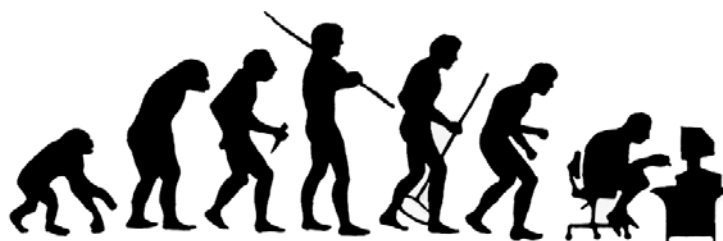


Рис. 2. Автомобильный номер штата Флорида с веб-адресом вместо названия штата

Сегодня мы окружены морями программного обеспечения, но у большинства пользователей нет никакого представления о том, как оно создается и почему оно работает так, а не иначе. Известно только, что оно очень нам нравится. Каждый может рассказать какую-нибудь жуткую историю, и не одну, связанную с программами, вроде всех этих авиационных кошмаров с многочасовыми ожиданиями в самолете на взлетной полосе или ошибочной отправкой багажа в Тимбукту. Вы весь день работали с некой программой, не зная, что нужно явно сохранять рабочий документ, а потом эта чертова программа грохнулась, и вы все потеряли. А случайная последовательность битов на экране в тот момент была очень похожа на средний палец. С момента выбора в 1982 году журналом «TIME» в качестве «человека года» персонального компьютера репутация последнего опустилась. Карикатура «Деволуция» хорошо отражает чувства многих пользователей (рис. 3).



Something, somewhere went terribly wrong

Рис. 3. Похоже, пользователи изменили свое мнение

Программное обеспечение может не быть отстоем и не должно быть им, но на практике оно такое, какое есть. Одна из причин этого в том, что программисты, архитекторы и менеджеры, занимающиеся его изготовлением, понимают своих клиентов гораздо хуже, чем следовало бы, и в сравнении с разработчиками в других отраслях. Их продукты отстойны не потому, что разработчики чего-то не умеют, а потому что они не знают, что нужно делать. Они оторваны от своих клиентов («жертв» — говорят некоторые) — тех, кто обеспечивает им зарплату, — и обычно этого не создают. Часто они решают не те задачи, добавляя новые функции, которые не нужны никому кроме них самих, и нанося ими вред всем пользователям. Если бы они понимали, что за люди их пользователи, то принимали бы другие, лучшие решения.

Например, приложения Microsoft Office, такие как Word и Excel, дают пользователю возможность перетаскивать главное меню (с кнопками File, Edit и Help) с его обычного места в верхней части окна к любому краю экрана или даже оставлять плавающим поверх документа (рис. 4). Я ни разу не встретил ни единого человека и не слышал о таком — подчеркиваю, ни единого, — кто когда-либо воспользовался этой функцией, даже среди моих знакомых разработчиков Office. Вы спросите, что плохого в том, что программа позволяет это делать? Есть несколько отрицательных последствий. Когда я тянусь мышью, чтобы щелкнуть по меню File, я иногда немного промахиваюсь, особенно если перепью кофе, и в результате начинаю перетаскивать панель меню по экрану. Приходится прервать работу, оттащить панель обратно, припарковать на место и полминуты клясть предков тех дебилов, которые написали эту идиотскую «функцию». Ме-

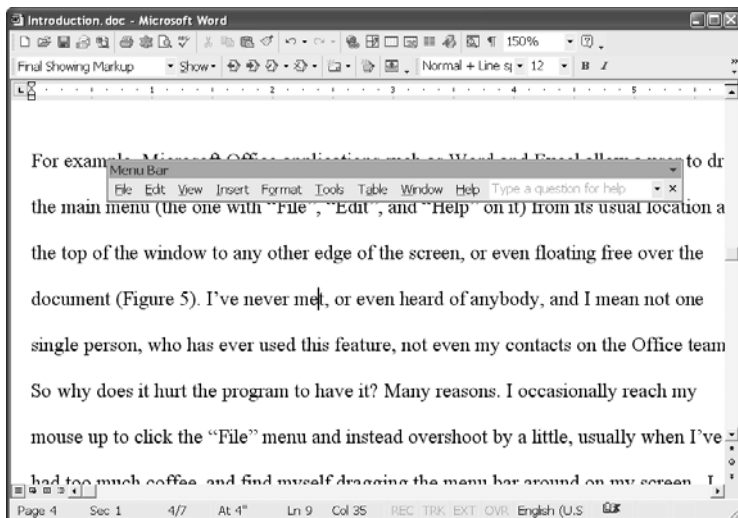


Рис. 4. Главное меню Microsoft Word, парящее над документом. Сделать-то можно, но кому это надо?

лочь, но два раза в день по 30 секунд, умноженные на миллиард пользователей, составят около 27 человеческих жизней, выбрасываемых впустую ежедневно.¹ Моя производительность, как и производительность большинства пользователей, станет выше, если эту функцию совсем убрать из программы, чтобы мы не тратили зря время и не теряли концентрации внимания, занимаясь такой чепухой. Кроме того, лишние команды программы («код», как называют их программисты), с помощью которых обеспечивается эта функция, увеличивают вероятность возникновения аварийных ситуаций и уязвимостей в системе защиты, так же как дополнительные движущиеся детали в любом механическом устройстве делают его менее надежным. Жизнь была бы лучше, если бы Microsoft взяла те деньги, которые были потрачены на проектирование, написание, тестирование, отладку, документирование и поддержку этой глупости, и сожгла их, а лучше — отдала мне. Хуже всего то, что такие контрпродуктивные безделушки расходуют ресурсы разработки, которые можно и нужно бы-

¹ Миллиард минут — это 694444 дней, 1901 год или чуть больше 27 раз по 70 лет.



Рис. 5. Так выглядел Clippy

ло потратить на то, что действительно важно для большинства пользователей, например, чтобы программа реже загибалась, и, если это случилось, не терялась их работа. Верх расточительности и создания контрпродуктивных функций — это Clippy (рис. 5), говорящая, танцующая и надоедливая скрепка для бумаг в Microsoft Office (добавлена в 1997 году, деактивирована по требованию народа в 2002).¹

Программное обеспечение и впредь будет отстойным, если мы не примем мер. Наши автомобили стали безопаснее (благодаря надувным подушкам и системам антиблокировки тормозов), надежнее (благодаря лучшему проектированию и сокращению отказов) и удобнее (благодаря плеерам компакт-дисков и подставкам для чашек), только когда этого требовали покупатели, приобретающие автомобили, в которых все это было, и игнорировавшие остальные. Мама всегда учила меня, что не стоит осуждать что-либо, если не имеешь представления, как сделать его лучше. У меня много хороших идей, которыми я поделюсь с вами, показывая глупость некоторых проектных решений в современных программах. Эта книга не задумана как сборник советов, но я раскрою все известные мне приемы ослабления самых ужасных результатов плохого проектирования — например, как отключить диалоговое окно подтверждения («Вы уверены?»

¹ Я не одинок в своем осуждении этой гнусной горлумоподобной твари. Как написали юристы Даля Литвик (Dahlia Lithwick) и Брандт Гольдштейн (Brandt Goldstein) в книге «Me v. Everybody: Absurd Contracts for an Absurd World» (Я против всех: абсурдные договоры в абсурдном мире), Workman Publishing, 2003: «3. Условие, касающееся маниакальной бумажной скрепки с бровями. Должно быть удалено/отключено/уничтожено все, что дает возможность этой маленькой глупой твари прыгать в правом нижнем углу моего экрана, издавать звуки, которые можно назвать лишь «механическим пуканием», и непрерывно подмечать: «Я вижу, что ты пишешь письмо с требованием выкупа...» или решать, что я хочу все свои документы преобразовать в электронные таблицы, а письма — в нумерованные списки».

Точно уверены? Точно-точно-точно уверены?»), которое появляется при отправке документа в мусорную корзину Windows. Главная задача книги — показать, что мы с вами можем сделать, чтобы наши голоса были услышаны в программной индустрии, и она прекратила выпуск отстоя, желательнее скорее. Ведь получилось же с Clippy?

В то же время я должен сообщить вам, что разработка программного обеспечения во многом отличается от разработки физических объектов — чего-то такого, что, скажем, можно уронить себе на ногу. Если, допустим, вы делаете стол из дерева, то свойства материала мешают выбрать одни варианты конструкции и вынуждают выбирать другие. Например, понятно, что нельзя паять, но можно крепить винтами, и толщина должна быть достаточно большой, чтобы стул не прогибался. Программное же обеспечение, напротив, предполагает очень мало внутренних ограничений. Гибкость материала здесь чрезвычайно велика. Как писал Фредерик Брукс в классической книге о разработке программного обеспечения «Мифический человек-месяц» (Addison Wesley, 1995), которую я бы предложил ему переименовать в «Мистическую гико-неделю» соответственно сокращению сроков проектов в нынешние времена, программисты работают «почти исключительно с чистой мыслью». Я всегда говорил, что программирование — это попытка затолкать дым в бутылку. Поэтому я объясню вам, какие вещи действительно трудно заставить делать программу по самой природе (например, успешно переносить внезапное выключение питания или быть совместимой с прежними версиями¹), а какие обусловлены просто безмозглым проектным решением тупого разработчика, который ни черта не соображает.

Я преподаю разработку программного обеспечения на факультете непрерывного образования Гарвардского университета и в различных компаниях во всех концах света. Я написал девять книг для программистов и руководителей программных разработок, а также много статей для журналов и информационных бюллетеней. Несмотря на эти недостатки, я обещаю, что мои тексты не будут скучными. Я старательно стремился избегать технического жаргона. Например, вы не встретите в этой книге слова

¹ В любимой шутке программистов говорится: «Как Творцу удалось создать мир всего за шесть дней? У него не было предыдущих инсталляций, совместимость с которыми необходимо было обеспечить». Мой редактор заметил, что Он также сэкономил на документации.

«гигабайт»¹, вместо которого я стану употреблять слова «четверть пространства жесткого диска», что на момент написания было примерно равноценно.

В сравнении с прочими моими книгами эту было удивительно легко писать. Покойный спортивный комментатор Ред Смит (Red Smith, 1906—1982) любил говорить: «Писать легко. Просто вскрываешь себе вены и пишешь». Надеюсь, вы уже заметили, что я весьма эмоционально отношусь к некоторым затрагиваемым здесь темам. Когда я заканчиваю свои шумные речи («Windows XP — приличный для своей цены продукт. Windows 98 — это кусок сами знаете чего, не стоящий затрат на то, чтобы его выкинуть к черту»), мои слушатели часто говорят «Платт, хватит ходить вокруг да около и скажи, что ты думаешь на самом деле». На что я отвечаю: «Если вы меня обвиняете в том, что я говорю о них то, что вижу, то я признаю себя виновным по этому пункту». Эта книга не просто попытка объяснить вам положение; это мое обращение к небесам в желании, чтобы здравомыслие возоблагодало в этой безумной профессии. Надеюсь, что, прочтя эту книгу, вы присоедините свой голос к моему.

¹ Ну да, только один раз и только здесь.

ГЛАВА ПЕРВАЯ

Кого считать идиотами?

«Ее никто не купит, — усмехнулся я, увидев в магазине название книги. — Кто на глазах у всех станет покупать книгу, которая возвещает всему миру, что он идиот? Это все равно, что покупать презервативы, украшенные надписью „для самых маленьких“».

Но все мы теперь знаем, что из этого вышло. «DOS для чайников» и парная к ней «Windows для чайников» стали абсолютными бестселлерами книг по компьютерной тематике. Идея вышла далеко за пределы компьютеров, и появились такие книги, как «Вино для чайников», «Аквариумы с соленой водой для чайников» и «Рак груди для чайников». В этой серии было продано более 100 млн. экземпляров, если верить книге «Как опубликовать свою книгу для чайников», которую я купил, чтобы найти издателя для того труда, который вы сейчас держите в руках.¹

Компьютеры заставляют людей чувствовать себя тупыми. Грамотные, образованные люди не могут заставить этот проклятый ящик делать то, что им нужно, и вместо того чтобы двинуться на Microsoft с факелами и вилами и вздернуть там чучело Билла Гейтса, они клянут себя, говоря: да,

¹ Результаты бывают неприятными. В октябре 2003 года комиссия по безопасности потребительских товаров сняла с продажи «Изготовление свечей и мыла для чайников», поскольку некорректные инструкции по смешиванию некоторых химикалий были чреваты получением ожогов. Не знаю, как следует в этой связи относиться к книге о раке груди или другой из той же категории — «Рак простаты для полных идиотов».

наверное, я тупой. В обществе, в котором никто никогда не виноват, в котором человек, проливший на себя кофе, подает иск к ресторану, где он это сделал, заставить пользователей винить себя за что-то — потрясающее достижение, хотя, возможно, не это было главной целью производителя. Почему программисты создают приложения, которые заставляют людей испытывать такие чувства, и почему люди смиренно терпят такие оскорбления от своих компьютеров?

Что было раньше

Разработчиков самых первых компьютерных программ не заботило, легко ли пользоваться их продуктами. Решать стоявшие перед ними задачи, например заставить принтер правильно выводить слова на бумагу, было столь сложно, что не оставалось ни времени, ни денег, чтобы облегчить жизнь пользователя. Время работы компьютера стоило чрезвычайно дорого — гораздо дороже, чем время пользователей. Заставлять человека учить сложные команды вместо того, чтобы потребовать от компьютера выводить их список в виде меню, было оправданно экономически. Сейчас соотношение цен стало обратным, но каждый работник компьютерной отрасли, кому сейчас за 30, вырос в тех старых условиях. Это, несомненно, отразилось на нашем сегодняшнем мышлении, как бы мы ни стремились расстаться с прошлым. Вспомните своих пожилых родственников, выросших во время Великой депрессии 1930-х годов: они даже сегодня не понимают, как можно выбрасывать носок, в котором всего одна дырка.

Первые пользователи компьютеров были готовы к тому, что их ждут трудности, не уступающие проблемам водителей авто в начале XX века, и их опасения обычно оправдывались. Почти все пользователи сами были программистами. Они редко испытывали потребность или даже желание облегчить положение. Мы воспринимали сложности — нехватку компьютерного времени, непонятность команд, низкое качество документации — так же, как те пионеры-мотористы воспринимали двигатели, заводящиеся рукояткой или постоянные проколы шин. Все были в одинаковых условиях. Мы были счастливы одной возможностью выполнять сложные вычислительные задачи (составление таблиц по результатам переписи населения, взлом кодов противника), так же как они были счастливы, что избавились от необходимости каждый день выгребать из конюшни навоз. Нам нравилось возиться со своими программами, находя им такое применение,

которое не предполагалось их разработчиками, так же как первым механикам нравилось копаться в своих двигателях. Генри Форд рассмеялся бы в лицо тому, кто сказал бы, что в его модели «Т» не хватает держателя для чашки.

В те времена бытовало представление, что облегчать работу с программами не следует. Если программу трудно написать, значит, и пользоваться ею должно быть тяжело, чтобы плоды программистских трудов могли пожинать только те, кто своими интеллектуальными усилиями доказал на это право. Даже сегодня я с удивительной нежностью вспоминаю гордость, которую испытал, когда сделал открытие, что команда для печати документа на той первой солидной вычислительной системе, с которой я работал (в 1975 году, на первом курсе колледжа), называется не Print или P, а Q, потому что вы ставите документ в очередь — очередь для печати. Я узнал волшебное слово. Я входил в круг избранных. Я был Умен!

Но когда аппаратная часть подешевела и компьютеры переместились из охлаждаемых кондиционерами застекленных зон, где над ними колдовали верховные жрецы, на верстаки умельцев-любителей, а после и на рабочие столы обычных служащих и в дома обычных людей, они обязаны были стать проще в обращении. Поэтому разработчикам приложений пришлось вкладывать время и деньги в разработку таких программ, с которыми пользователи действительно могли бы работать. Почему же результаты оказались такими жалкими?

Почему и по сей день программы — отстой

Та часть компьютерной программы, которая взаимодействует с человеком-пользователем, то есть получает от него команды и входные данные, выводит сообщения и результаты работы, называется **интерфейсом пользователя**. Разработка интерфейса пользователя, так же как и многие другие области вычислительной науки, — искусство, требующее специальных умений и знаний, о котором большинство программистов не имеет никакого понятия. Программистами они стали потому, что научились общаться с микропроцессором — кремниевой микросхемой, составляющей основу вычислительной машины. Но интерфейс пользователя по определению предназначен для общения с совершенно другим программно-аппаратным устройством — живым человеком. Нет ничего удивительного в том, что искусство беседы с логичным, безошибочным и глупым чипом

совершенно отлично от искусства беседы с нелогичным, склонным ошибаться и разумным человеком. Однако считается, что тот, кто преуспел в первом искусстве, автоматически должен справиться со вторым. Обычно это не так, и сам программист этого почти никогда не сознает. Вот поэтому интерфейсы пользователя, созданные программистами, обычно отвратительны — во всяком случае с точки зрения того бедолаги, который мучается с этим хламом.

Почему так происходит? Ведь раз программисты умеют программировать, значит, определенный уровень интеллекта у них есть. В большинстве своем они неплохо ладят с этой кремниевой микросхемой, иначе бы их довольно быстро уволили и им пришлось бы подбирать себе другую профессию, в которой они могли оказаться полезнее обществу, например заняться кровельными работами. Почему же, принимаясь за разработку интерфейса пользователя, они превращаются в слабоумных идиотов? Причина проста, и она всегда лежит в основе любых мыслимых случаев взаимного непонимания: они не знают своих пользователей.

Каждому программисту кажется, что ему достоверно известно, чего хотят пользователи. В конце концов, он же ежедневно с утра до вечера работает с компьютером и не может этого не знать. Он говорит себе: «Если я сделаю такой интерфейс пользователя, который нравится мне, то и пользователи его полюбят». *Ничуть не бывало!* Если только программист не пишет свои программы для таких же прожженных технарей, как он сам, пользователям это не понравится. Своим ученикам, которых я учу программировать, я советую зарубить на носу — наряду с фразами типа «мусор на входе, мусор на выходе» и «всегда снимай колоду» — Первый Закон Платта, Единственный Закон Проектирования Пользовательского Интерфейса:

ПОЗНАЙ СВОЕГО ПОЛЬЗОВАТЕЛЯ, ИБО ОН НЕ ТАКОЙ, КАК ТЫ

В качестве простейшего примера возьмем программу управления личными финансами типа Quicken или Microsoft Money. Пользователь имеет с ними дело по несколько часов раза два в месяц. Естественно, ему трудно вспомнить, как управляться с этой программой, — в отличие от той, с которой он работает ежедневно. Поэтому подсказки и помощь должны быть детальными — такими, которые покажутся назойливыми и раздражающими человеку, не вылезавшему из этой программы (например, программисту). Программист не в состоянии поставить себя на место этого

пользователя. Программист очень хорошо знает свою программу и не может себе представить, что не все такие же, как он.

В своем заблуждении, что все люди похожи на них, программисты совершают две главные ошибки, проектируя интерфейс пользователя. Они придают большее значение степени контроля, а не легкости использования, и стремятся к тому, чтобы можно было делать сложные вещи, вместо того чтобы сделать простые вещи простыми. И кроме того, они полагают, что пользователей интересует внутреннее устройство их программы, что не соответствует действительности. Мне приходилось совершать обе ошибки, и я сожалею о заблуждениях своей молодости.

Контроль или легкость эксплуатации?

Каждый раз, читая лекции в какой-нибудь компании, я спрашиваю у слушателей, многие ли из них ездят на машинах с ручной коробкой передач (как я сам). Обычно руки поднимают около половины. Тогда я спрашиваю, многие ли *приехали бы* на машине с ручной передачей, если бы им это позволили жены, или они поехали на машине с автоматической коробкой, потому что превращаются в брюзгливых старикашек вроде меня. Обычно руки поднимает половина из оставшихся.¹ «Теперь согласитесь, — говорю я, — что ручная передача сложнее в освоении и эксплуатации, чем автоматическая, но обеспечивает лучшую управляемость?» Они чувствуют, что здесь какая-то ловушка, но обычно еще не могут ее распознать, поэтому соглашаются, но неохотно. «А теперь скажите, какой, по вашему, процент машин с ручной передачей продается в США?» Они неловко ежатся и обычно говорят что-то вроде «мало, процентов 30». Размечтались. По разным оценкам от 10 до 14 процентов. Пусть будет 12 с половиной, для простоты.

Этот опрос показывает, что шесть из восьми программистов ценят незначительное преимущество в степени контроля и эффективности настолько высоко, что, потратив 25 или более тысяч долларов на своего железного коня, они готовы на постоянную дополнительную работу в тече-

¹ Сделайте такую проверку в своей компании. Благодаря ей вы, возможно, узнаете нечто новое о своих пользователях. Потом зайдите на сайт этой книги (www.whysoftwaresucks.com) и сообщите мне полученные результаты. Спасибо.

ние всего срока службы автомобиля, чтобы этим преимуществом пользоваться. И лишь один из восьми человек среди всего населения принимает такое же решение при тех же условиях выбора. А на самом деле еще меньше, потому что все эти шесть программистов также входят в эту одну восьмую. Процент нормальных людей, стремящихся выполнять лишнюю работу, близок к нулю. Программисты ценят контроль. Пользователи ценят легкость эксплуатации. Ваш пользователь не такой, как вы.

Вот пример, показывающий, как можно все испортить. Когда-то справочная служба АТ&Т была простой и удобной. Вы спрашивали чей-нибудь номер, и автоматический голос отвечал «Нужный вам номер: 555-1212. Пожалуйста, запишите его». Если вы не положили трубку, номер произносился еще раз, и можно было проверить, правильно ли вы записали его. Просто. Легко. Невозможно ошибиться. Отлично. Затем в АТ&Т ввели возможность автоматического набора этого номера. Голос произносил: «Нужный вам номер 555-1212 можно автоматически набрать за дополнительную плату 50 центов. Нажмите 1, если вы согласны, и 2 в противном случае». Простота сохранилась, но появилась новая более мощная функция для тех, кто готов был за нее заплатить. Тот, кому она была не нужна, мог просто положить трубку. Потом какому-то айди-ойту [*sic*, см. примечание¹] пришла в голову абсолютно дикая мысль. Когда я в прошлый раз хотел воспользоваться справочной АТ&Т, мне было сообщено: «Нужный вам номер можно автоматически набрать за дополнительную плату 50 центов. Нажмите 1, если вы согласны, или 2 в противном случае». Служба не называет номер, пока я не нажму одну из кнопок. Мне пришлось отнять трубку от уха, найти глазами клавиатуру (а лет после 45 это уже труднее), положить карандаш, который я держал в руке, чтобы

¹ Происхождение слова связано с курсом, который я однажды читал. Некий слушатель слабо успевал, потому что не слишком утруждал себя, и взял привычку писать мне длинные и эмоциональные электронные письма, жалуясь на несправедливость. Конечно, он был очень расстроен. С другой стороны, трудно воспринимать всерьез того, кто якобы учится на последнем курсе престижного университета и не научился правильно писать слово «идиот». Обычно он писал мне: «Платт, вы айдиойт. И тот, кто ставит оценки, — айдиойт, и тот, кто мне посоветовал этот курс, — айдиойт, и я — айдиойт, потому что послушал его». С тех пор мы с коллегами пользуемся словом *idoit* (произносится «айдиойт», или «идуа», если вы француз) для обозначения людей настолько невежественных, что они не могут даже правильно написать «идиот».

записать номер, нажать нужную кнопку, снова взять карандаш и приложить трубку обратно к уху. Только после этого мне сообщили, что нужный номер — 555-1212. Сложная и мощная операция возможна, но простая операция перестала быть простой. Разработчик этой системы, очевидно, ценит контроль выше простоты, но гарантирую, что пользователи придерживаются другой точки зрения. Того, кто придумал такую операцию, следовало бы заставить проделывать ее 500 раз в день. К концу недели он бы застрелился.

Мой оператор сотовой связи Verizon, напротив, поднял простоту пользования на новую высоту. В Verizon сообразили, что почти все, кто обращается в справочную службу, хотят кому-то срочно позвонить, так почему бы не помочь? Когда я звоню в справочную службу с сотового телефона, автоматический голос произносит: «Номер 555-1212. Соединяю вас». Все происходит автоматически, без каких-либо действий или даже мыслей с моей стороны. Новый номер сохраняется на моем телефоне в списке последних набранных, поэтому я могу при желании поместить его в свой справочник. Те немногие, кто собирался только записать номер, могут просто повесить трубку, что они и так сделали бы. Простые вещи оказываются простыми. Сложные и мощные — тоже простыми. Такая схема настолько же хороша, насколько плоха схема AT&T.¹

Меня не интересует, как работает ваша программа

Вторая ошибка, которую допускают программисты при разработке пользовательского интерфейса, заключается в их стремлении заставить пользователя разобраться во внутренней логике своей программы. Вместо того чтобы приспособить интерфейс к способу мышления пользователя, программист хочет заставить его приспособливаться к своему способу мышления. Более того, программист считает такой подход правильным. «Но ведь так работает моя программа», — говорит он, недоумевая, как у кого-то могла возникнуть мысль сделать иной интерфейс.

¹ Я только что прочел, что Verizon собирается передавать на ваш телефон маршрут проезда к тому месту, телефонный номер которого вы запросили, установив ваше положение с помощью встроенного в телефон чипа GPS. Очень рассчитываю, что они ничего при этом не усложнят и не испортят.

Приведу пример, показывающий, что я имею в виду. Откройте Блокнот Windows или любую другую программу-редактор и наберите с клавиатуры произвольный текст. Теперь выберите в главном меню пункты File→Exit или щелкните по крестику в правом верхнем углу. Окно сообщения, которое вы увидите, показано на рис. 1.1.

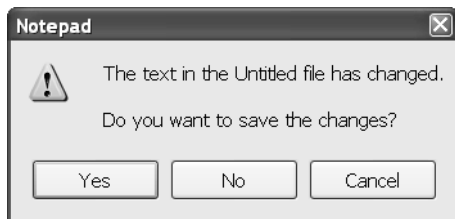


Рис. 1.1. Блокнот спрашивает у пользователя, нужно ли сохранять изменения

О чем она спрашивает, эта программа? Говорит, что какой-то файл изменился, но я нигде не видел никаких файлов. Какие, к черту, изменения, которые нужно сохранять?

Дело в том, что Блокнот обычно применяется для редактирования документов («файлов» на техническом жаргоне), находящихся на жестком диске. Когда вы открываете документ, Блокнот копирует его с диска в память компьютера. Когда вы добавляете или удаляете текст с клавиатуры, программа изменяет содержимое этой копии в памяти. (В данном примере мы не открывали существующий документ, а программа создала в памяти новый документ, назвав его «Безымянный».) После того как вы завершите работу с документом, программа должна записать его из памяти обратно на диск, и эта операция называется «сохранить файл». Если этого не сделать, вся ваша работа пропадет, и вы очень рассердитесь.

Программист написал программу, которая работает подобным образом (копирует документ с диска в память, изменяет копию, находящуюся в памяти и записывает ее обратно на диск), потому что для него это было проще всего. И он выбрал совсем неплохой способ. Чтение или запись символов с диска (вращающихся металлических пластинок) происходит в тысячи раз медленнее, чем те же действия с памятью (электроны движутся со скоростью света), поэтому возможно, что такой способ организации внутренней логики программы действительно самый лучший.

Но интерфейс пользователя, который сделал этот программист, просто отображает внутреннюю логику программы. Что в этом плохого? Автор программы заставляет вас разбираться в том, как она работает. А вам не нужно знать или думать о том, как устроена программа внутри, чтобы успешно пользоваться ею, так же как для управления автомобилем вам не нужно знать или думать о том, впрыскивается в двигатель топливо или поступает из карбюратора.

Обычно ваши мысли идут не таким путем, какой предлагает эта программа. Большинство людей представляет себе редактирование документа в компьютере аналогичным работе с карандашом и бумагой (вы еще помните, что это такое?). Вы делаете пометки карандашом, и они остаются на бумаге. То, что вам не нужно, вы стираете резинкой. Если документ вам больше не нужен, вы комкаете лист и выбрасываете его. Бумага хранит всю вашу работу, если только вы не потратите дополнительную энергию, чтобы уничтожить ее. Но Блокнот предлагает вам другое. Каждый новичок в компьютерах на этом попадается — говорит «нет», и Блокнот выкидывает всю вашу работу — хорошо, если ее немного. В конечном счете пользователь привыкает думать, как компьютерная программа, точнее, как программист, который написал эту чепуху. Алан Купер, гуру в области проектирования интерфейсов пользователя, определяет «грамотного пользователя компьютера» как того, кто «набил себе столько шишек, что у него одеревенела шкура и он больше не чувствует боли».

Вопрос и ответ на него были бы намного понятнее, если бы текст окна диалога гласил: «Выбросить все, что ты только что сделал?» Это абсолютно тот же самый вопрос, но заданный с точки зрения пользователя, а не программиста. Но программист думает только о том, что делает программа, то есть о записи на диск, и спрашивает, нужна ли запись. Автор требует, чтобы вы встали на его место, даже не попытавшись встать на ваше. Иначе он сформулировал бы вопрос по-другому. Иначе он увидел бы нелепость своего вопроса и создал более удачный интерфейс пользователя, даже если бы программа, которая за ним скрывается, продолжала работать по-прежнему.

Программа управления личными финансами Microsoft Money устроена правильно. Тот, кто ее проектировал, понимает, что мысленной моделью для пользователя служит чековая книжка, поэтому он и показал на экране опись чеков (рис. 1.2). Пользователь-новичок чувствует себя здесь привычно и удобно (ну, относительно). Чек, с которым вы работаете в дан-

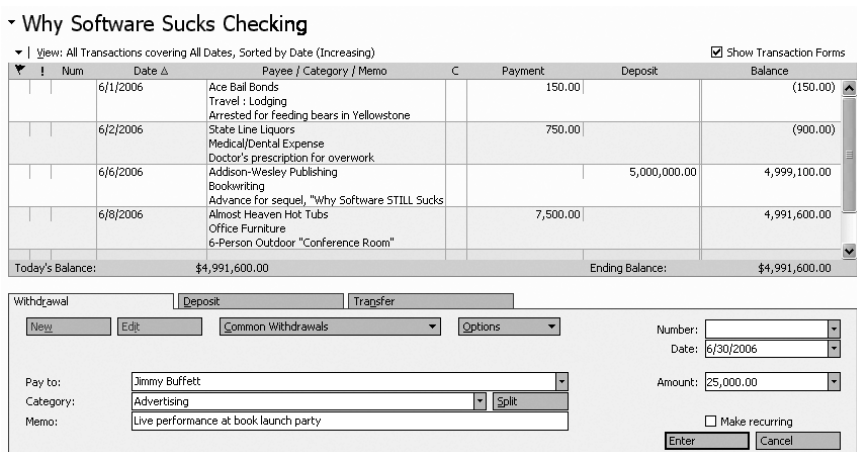


Рис. 1.2. Интерфейс пользователя *Microsoft Money*, похожий на настоящую чековую книжку

ный момент, выделен другим цветом. Вы вводите в чек нужные данные и нажимаете Enter. Чек перемещается вверх, цвет его становится таким же, как у остальных, а в рабочей области появляется новый пустой чек. Если включен звук, раздастся «динь-динь», как в кассовом аппарате.¹ Программа не спрашивает у вас, нужно ли сохранить чек. Само нажатие клавиши Enter говорит программе, что вы хотите сохранить информацию. Если вы потом передумаете и захотите поменять дату чека или совсем удалить его, вам придется щелкнуть по этому чеку и ввести новые данные. Спрашивается, когда эта программа читает данные с диска и пишет их обратно? Не знаю и знать не хочу. И вам это знать ни к чему. Интерфейс пользователя этой программы следует вашей мысленной модели, а не заставляет вас что-то изучать и разбираться с внутренними конструктивными решениями программиста.

Такой тип проектирования интерфейса пользователя гораздо лучше. Я, будучи пользователем, не хочу думать о самой программе. Я хочу ду-

¹ Этот звук, конечно, анахронизм. Где вы сейчас услышите такие позвякивания у кассового аппарата? Современные кассы попискивают и шумят, как компьютеры, каковыми они и являются. Но этот звук сохранился, как сохранилось выражение «повесить телефонную трубку», хотя мы всего лишь нажимаем кнопку.

мать о работе, которую делает для меня программа, например достаточно ли у меня денег, чтобы оплатить этот счет? Еще один гуру в области дизайна интерфейса Дональд Норман (Donald Norman) очень хорошо выразил это представление в названии одной из своих книг — «The Invisible Computer» (Невидимый компьютер), MIT Press, 1999. В идеале я вообще не должен думать о программе.

Это одна из важных причин, по которым с программами трудно работать, и они заставляют вас чувствовать себя дураком. Вас принуждают думать по-программистски, даже если вы не программист и не собираетесь им быть. А вам это и ни к чему. Не надо быть механиком, чтобы водить машину, врачом, чтобы принять аспирин, и мясником, чтобы поджарить гамбургер. Вы платите за этот продукт свои деньги, заработанные тяжелым трудом. Это программист должен подстраиваться под вас, а не наоборот.

Плохая функция и хорошая функция

А вот еще один пример того, как программисты уродуют интерфейс и заставляют пользователей чувствовать себя тупыми. Выберите на рабочем столе Windows любой документ. Теперь нажмите клавишу Delete. Если только вы не сумели ранее отключить эту функцию, перед вами появится окно подтверждения, показанное на рис. 1.3, спрашивающее, действительно ли вы хотите удалить этот файл.

Был хоть один раз, когда вы сказали «ой, я не собирался это делать, спасибо за предупреждение» и нажали No? Может быть, вы видели, как это делает кто-то другой, или слышали про такой случай? Лично я — нет. Подтверждениями так часто злоупотребляют, что по иронии судьбы они стали совершенно бесполезными. Это окно постоянно кричит «волк», как пастушок в басне Эзопа, поэтому никто не обращает на него внимания,

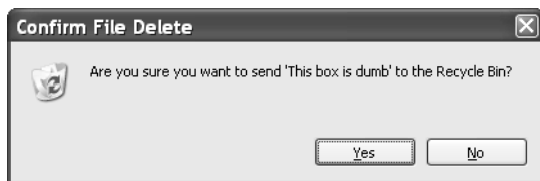


Рис. 1.3. Бессмысленное окно подтверждения от мусорной корзины Windows

даже когда предупреждение относится к файлу, который вы действительно не собирались удалять. Вы так часто с этим встречались, что уже не обращаете внимания. Действуя на автопилоте, вы автоматически нажимаете кнопку Yes. Безопасности вы не получаете. Никакой. К счастью, именно это диалоговое окно можно отключить.¹ Есть масса других, от которых вам избавиться не удастся, хотя их быть не должно. Совсем. Нигде. Никогда.

В жизни вы совершаете разные операции, и при этом у вас не требуют подтверждения. Когда вы поворачиваете ключ зажигания, ваша машина не спрашивает у вас, действительно ли вы хотите завести двигатель. В супермаркете служащий не спрашивает у вас, действительно ли вы хотите купить эти товары, когда вы выкладываете их на ленту у кассового аппарата. Вспомните, сколько еще книг вы купили на Amazon.com после того, как обнаружили их фирменную функцию «заказ одним щелчком».²

Зачем программисты все время просят подтверждений? Они делают это потому, что им кажется, будто их пользователи запутались и не понимают, каковы будут последствия того, о чем они только что попросили программу. Не исключено, что так оно и есть — с учетом ужасного качества остальных частей интерфейса пользователя. Но запрос подтверждения не решает этой проблемы. Если пользователь был сбит с толку, когда изначально дал ту команду, которая вывела диалоговое окно подтверждения, то после его появления он запутается еще больше. Поскольку программа показывает нежелание выполнять то, что ей сказано, он начинает думать, что сделал какую-то ошибку. Применение окон запроса подтверждения избавляет программистов от того, чтобы а) четко объяснить пользователю последствия его действий, чтобы он не попытался сделать то, что ему не нужно, и б) предоставить возможность восстановить ситуацию, если пользователь действительно совершит то, о чем позднее пожалеет.

¹ Щелкните правой кнопкой по Корзине, выберите Свойства во всплывающем меню и сбросьте флажок Показывать запрос подтверждения.

² Мало кто знает, что в первом прототипе этой функции программисты Amazon выводили диалоговое окно подтверждения с вопросом «Вы действительно хотите заказать этот товар одним щелчком?», когда пользователь щелкал по кнопке «заказ в 1 щелчок», тем самым превращая его в 2-щелчковый процесс. Они бились за сохранение этой функции, пока не последовало прямое указание президента Amazon Джеффа Безоса убрать ее и сделать действительно процедуру одного щелчка.

Но что если пользователь действительно совершает ошибку? Если, например, вы положите на транспортер у кассы фонарик и комплект батареек, которые для него не годятся, разве внимательный служащий не заинтересуется, действительно ли они вам нужны? Разве не должен хороший интерфейс пользователя оберегать нас от таких ошибок? Конечно, должен, и прелесть компьютерных программ в том, что они в состоянии это делать. Но это достигается не с помощью задаваемых каждый раз глупых вопросов «а вы уверены, что действительно хотите сделать эту ерунду, о которой только что меня попросили?» В хорошем интерфейсе пользователя такой проблемы вообще не должно возникать. Например, на веб-странице, где продается фонарик, может быть флажок «вместе с батарейками». По умолчанию он должен быть установлен, потому что без батареек фонарик не будет работать. Если у покупателя уже есть достаточное количество батареек такого размера, он может сбросить флажок. А еще лучше, если фонарик будет упакован уже с батарейками внутри, чтобы быть готовым к работе в тот момент, когда вы его достанете из упаковки, не задумываясь о батарейках. Толковый разработчик интерфейса пользователя подумает об этом еще до начала программирования. Если программисту кажется, что необходимо показать окно подтверждения, то я вам гарантирую, что он напорточил где-то в интерфейсе, иначе оно не понадобилось бы. Он, наверное, и не пытался обойтись без него, ему, наверное, и в голову не пришло такое. Окно подтверждения — это подпорка для ленивого или несведущего программиста, расплачиваются за которую все пользователи. К тому же она не действует.

Но разве не было бы желательно подтверждать намерение совершить некое деструктивное действие типа удаления файла? Нет. Еще одна из причин, по которым не требуется подтверждать намерение завести двигатель или купить товар в супермаркете, состоит в том, что эти операции легко и просто отменить, если вы решите, что сделали ошибку. Вы просто выключаете зажигание или возвращаете ненужный товар. Компьютерные программы могут легко и просто делать копии документов или участков памяти. Благодаря этому программисты могут предоставлять пользователям возможность отменять выполненные действия. Функция Undo (Отмена) — одна из самых полезных в арсенале разработчика интерфейса пользователя. По моим представлениям, это самое большое отдельное достижение в проектировании интерфейса пользователя со времени появления мыши.

Когда вы нажимаете кнопку Yes в окне подтверждения на рис. 1.3 (или если вы совсем отключили это окно), Windows не стирает документ в компьютере, а перемещает его в другую область диска под названием Корзина (Recycle Bin), аналогичную известному мусорному ящику на Макинтошах. Если после этого вы передумаете и пожелаете получить документ обратно, то можете достать его из Корзины, если только перед тем не очистили ее. Как правило, вам действительно нужно убрать файл. Гораздо эффективнее исправлять относительно небольшое количество ошибок, которые действительно случаются (например, можно выбрать для удаления не тот файл, неточно переместив указатель мыши, — со мной такое бывает), чем пытаться предотвратить такие ошибки, раздражая пользователей показом окна подтверждения при каждом удалении, особенно если на него привычно не обращают внимания. Ложка лекарства не стоит бочки профилактики.

Функция «отменить» может применяться не только к операциям с файлами, но и внутри приложений. Обычно она располагается в меню Edit (Правка) вместе с парной к ней Redo (Повторить), которая, разумеется, отменяет отмену. У меня обычно и пяти минут не проходит, чтобы я не отменил какое-нибудь действие: либо напечатал дурацкую фразу, либо текст переместишь не туда, куда нужно. Программисты, которые реализуют такую функцию, — настоящие друзья пользователя. При случае я всегда угощаю их пивом, и вам советую поступать так же. Заставить эту функцию действовать так, чтобы у пользователя не возникало никаких проблем при ее вызове, невероятно трудно (как известно, «чтобы было легко, нужно немало потрудиться»): всего лишь нажать Ctrl-Z (мизинцем и средним пальцем левой руки) — и вот оно вернулось, как будто ничего не произошло. Программа должна просить подтвердить только те операции, которые она не сможет отменить. А отменить она должна уметь все.

Подлинная прелесть Undo в том, что она позволяет исследовать программу. Иногда бывает непросто понять, как действует незнакомая вам программа, ориентируясь только по коротким надписям в пунктах меню и миниатюрным картинкам на кнопках в панели инструментов. Но команда Undo есть, поэтому можно поэкспериментировать, проверяя разные варианты, и не бояться что-либо испортить, чего нельзя будет исправить несколькими нажатиями клавиш. Можно подвигать абзац и оценить, не будет ли смотреться лучше в другом месте, а если нет, то вернуть его обратно. Программисты часто относятся к вводу неправильных данных как к действиям идиота, не удосужившегося сесть и прочесть руководство.

И они ошибаются. Это основной метод, используемый человеческим родом при обучении. Приложение с функцией Undo становится обучающим, а не устрашающим. Оно признает и еще больше укрепляет человеческую природу пользователя. Неспособность реализовать эту функцию надлежащим образом — смертный грех.

Прекращение работы и демонстрация идиотизма

Если Undo реализована правильно, деструктивная операция в системе остается одна, а именно очистка Корзины. Кое-кто скажет, что уж у этой-то операции должно быть диалоговое окно подтверждения, и оно действительно есть. Но даже тут окно подтверждения служит лишь защитой против другого скверного проектного решения — расположения пункта контекстного меню Explore (Проводник) рядом с пунктом Empty Recycle Bin (Очистить Корзину). Одно неточное движение мыши, проскочившей три строчки меню вместо двух, и вы получаете одно вместо другого. Это плохо. Если это единственное деструктивное действие в системе, то очистка Корзины должна осуществляться особым образом, больше ни для чего не используемым, например одновременным нажатием двух кнопок мыши («взятием аккорда») или щелчком при одновременном нажатии какой-либо клавиши. А еще лучше было бы очищать корзину автоматически, например удаляя из нее файлы по истечении определенного промежутка времени, величину которого можно было бы задавать, начиная, скажем, с месяца, чтобы пореже приходилось опустошать корзину вручную. Вот бы дома иметь такие мусорные бачки! Окна с запросами подтверждения не должны появляться нигде и ни при каких обстоятельствах. Программист, который их создает, не справляется со своими обязанностями и не должен заниматься разработкой интерфейсов пользователей.

Худшие свои черты интерфейсы, разработанные программистами, проявляют при показе пользователю сообщений об ошибках. Как раз сегодня я, вместо того чтобы работать над этой главой, залез на домашнюю страницу *CNN.com* и захотел сохранить ее у себя на диске. Я выбрал в меню веб-браузера File→Save. Появилось диалоговое окно, сообщавшее о ходе операции сохранения — 5% сохранено, 15% и т. д. вплоть до 99%. Затем это окно исчезло и выскочило то, которое показано на рис. 1.4.

Это окно — творение полного идиота. Почему нельзя было сохранить веб-страницу, и можно ли исправить ситуацию? Была ли страница каким-

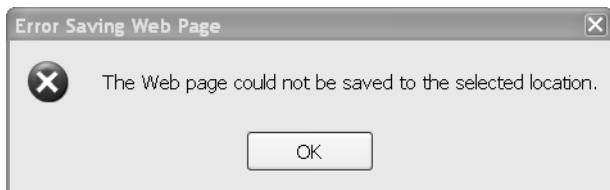


Рис. 1.4. *Очень глупое диалоговое окно*

то образом защищена от копирования, как пытаются иногда делать на сайтах профессиональных художников? Или сервер оказался недоступен? Почему индикатор выполнения дошел до 99 процентов, если операция была обречена? Где те 99 процентов страницы, которые были сохранены, как о том сообщила программа? Пусть это не вся страница, но я предпочел бы 99 процентов, нежели ничего. Куда она исчезла? В окне сказано, что страницу нельзя сохранить по выбранному адресу; надо ли это понимать так, что по другому адресу ее можно было сохранить? Если да, то где и как это узнать? Если нет, то причем тут адрес? Броузер уже сумел удовлетворительно показать мне эту страницу, поэтому я захотел ее сохранить. Почему он не может сохранить те данные, которые фактически показывает? В окне ничего не сказано относительно того, как узнать, в чем проблема и где найти дополнительные сведения. А в качестве ответа предлагается только кнопка ОК. Нет, для меня совсем не «ОК», если операция не выполнена и программа не может объяснить почему. Даже заголовок окна «Ошибка сохранения веб-страницы» вводит в заблуждение. Я не совершал ошибок. Я сделал то, что программа мне позволяла. Это программа сделала одну ошибку, не сумев сохранить мою страницу, а потом другую, не сумев объяснить причину. Создать такую неразбериху с помощью полутора десятков слов — верх идиотизма, с которым мне приходилось встречаться.

Алан Купер, уже упомянутый мной гуру проектирования интерфейса пользователя, описывает подобные ситуации как «прекращение работы и демонстрация идиотизма». Если сохранить страницу нельзя, броузер должен был это знать и предупредить меня, дав какое-то объяснение, — желательно не в виде очередного дурацкого окна. Может быть, при показе этой страницы нужно сделать неактивным пункт меню «сохранить» или изменить текст на «сохранение невозможно — страница защищена», чтобы я знал, в чем дело. Если нельзя сохранить страницу целиком, программа

должна сохранить то, что может, и сообщить мне о пропущенном, — опять же не вынуждая меня щелкать по очередному дурацкому окну. Возможно, на сохраненной странице следует поместить какие-то маркеры в тех областях, которые сохранить не удалось, и показывать красные крестики, как это делает браузер для ненайденных элементов страницы.

Проведя некоторые изыскания, я смог восстановить картину происшедшего. В настройках моего браузера я заблокировал некоторые виды раздражавшего меня контента. Браузер показывает их как пустые области, что нравится мне гораздо больше, чем дурацкие пляшущие рекламные объявления, которые там должны были быть. (Об идиотизме пляшущей рекламы мы поговорим в другой главе.) Когда программный код, который должен был сохранить страницу, добрался до этих элементов страницы и обнаружил, что они заблокированы, он не пропустил их, как это сделала часть программы, ответственная за отображение страницы. Код поперхнулся в этом месте, прервал всю процедуру вместо сохранения того, что можно было, и прекратил работу, продемонстрировав описанный выше идиотизм.

Что нужно, чтобы не чувствовать себя идиотом, когда тебе показывают такое непонятное окно? Нужно знать, что здесь не ваша вина, а неспособность программиста справиться со своими обязанностями; что пользователь не обязан понимать такие глупые сообщения. Полезно также представить себе, как, схватив этого программиста за горло, вы наносите ему коленом удары в пах. Здесь отсылаю вас к советам в конце этой главы и в заключительной главе этой книги.

Испытание на подопытных животных

Программист не выпустит продукт, не протестировав его внутренней работы (ОК, *не должен выпустить*). Почему же тогда он считает, что можно обойтись без тестирования интерфейса пользователя и не проверять, смогут ли пользователи с ним работать? Потому что ему самому продукт нравится, и он сам может с ним работать, значит, и у других получится. Как мы уже видели, такое подсознательное предположение почти всегда ошибочно. Если пользователь не может разобраться, как работать с программой, компьютер превращается в очень дорогое пресс-папье. Проверка интерфейса пользователя, которая называется **тестированием юзабилити**, — процедура сложная и дорогая, но необходимая.

Недостаточно просто отдать пользователям программу и потом поинтересоваться, понравилась ли им она. Часто они не могут вспомнить, какие действия совершали, или не хотят рассказывать о возникшей проблеме, стыдясь, что не смогли с ней справиться, или не хотят обидеть программиста, сказав ему, что результат последних двух лет его профессиональной деятельности — полное баракло. (Последней проблемы у меня, как вы догадываетесь, не возникает.)

Чтобы выяснить ситуацию, программист должен знать точные действия пользователя при его взаимодействии с интерфейсом. С чего начинает пользователь? Каковы его последующие действия? Сколько раз ему нужно попытаться что-то выполнить, прежде чем он разберется, как правильно действовать? Сколько времени требуется, чтобы обратить внимание на некую функцию?

При этом наблюдение за пользователем не должно оказывать влияния на его поведение. Это означает, что пользователь должен находиться в отдельном помещении и располагать только теми вспомогательными средствами (например, электронной документацией или Google), которые у него будут в реальных условиях. Наблюдать за пользователем нужно через зеркальную стену, вести видеозапись действий и регистрировать их программным образом, чтобы знать, какие он нажимал клавиши и что делал мышью при работе с приложением. В некоторых лабораториях даже применяют устройства для слежения за направлением взгляда пользователя, смотрящего на экран.

Когда вы это сделаете, на вас снизойдет просветление. Алан Купер в своей классической книге о проектировании интерфейса пользователя «About Face: The Essentials of User Interface Design» (IDG Books, 1995) пишет: «[Профессионалы юзабилити] затаскивают программистов в темную комнату, где они смотрят через зеркальное окно, как несчастный пользователь сражается с их программой. Поначалу программисту кажется, что у пользователя какое-то заболевание мозга. Наконец, в результате длительного и мучительного наблюдения программист вынужден отступить перед фактами. Он признает, что интерфейс пользователя нуждается в доработке и торжественно обещает ее выполнить».

К сожалению, слишком часто тестирование юзабилити откладывается до поздних этапов разработки, едва ли не до самой поставки продукта. Как правило, работа отстает от графика, поэтому тестирование юзабилити часто вообще опускают. Если же тестирование дает полезную информа-

цию, то в графике работ часто уже не остается времени на исправления.¹ Тестирование юзабилити следует проводить рано, в идеале вообще до начала программирования.

В некоторых компаниях полагают, что широкие испытания перед самым выпуском позволяют получить продукт с лучшими потребительскими свойствами. Например, в Microsoft принято «съесть то, чем собираешься кормить свою собаку» («dog-fooding»). Для этого перед выпуском продукта в свет его передают реальным пользователям внутри компании, например пересаживают секретарш на новую версию Word для Windows. При этом выявляются некоторые ошибки, но они оказываются логическими ошибками программирования, типа арифметических, нарушающими работу программы. Искать конструктивные ошибки, особенно в сфере юзабилити, слишком поздно. Да, собачий корм будет немного вкуснее, чем без такой проверки. Но он не превратится в корм для кошек, а проверка происходит слишком поздно, и проверяющие не успевают выяснить, что потребителями корма станут не собаки, а кошки, а то и жирафы.

А вот пример правильного подхода. Однажды я консультировал страховую компанию, в которой писали Windows-программу, чтобы заменить дорогие терминалы IBM. Довольно необычно для страховой компании, что они провели испытание юзабилити, о котором я сейчас рассказывал. И они все сделали грамотно — с видеозаписью и наблюдением через зеркало. Выяснилось, что пользователи в целом довольны приложением и удовлетворены его работой. Но они по привычке нажимали Enter для перехода из одного поля ввода в другое, как это было принято в терминалах IBM, вместо клавиши Tab, как в приложениях Windows. Разработчикам предложили изменить поведение программы. После тщательного размышления разработчики решили, что хотя технически такое изменение сделать очень легко, оно не сильно облегчило бы жизнь пользователям, вопреки надеждам последних. Конечно, можно было заставить приложение работать по-старому. Но все новые коммерческие приложения Windows, которые планировалось устанавливать, будут работать по-другому, и пользователи сойдут с ума, переключаясь с одной схемы на другую по

¹ В одной из своих книжек для программистов я сформулировал закон Платта «экспоненциального роста оценки», который гласит, что «всякий программный проект требует втрое больше времени против самой тщательной оценки, даже если она получена с учетом этого закона».

несколько раз на дню. Разработчики все же убедили пользователей смириться с изменением привычного стиля. После неизбежного периода протестов пользователи успокоились, чему способствовало и ужасное положение на рынке труда в тот момент. Я не утверждаю, что программисты должны впихивать в глотку пользователям то, что тем должно быть полезно. Обычно это не удается, но здесь был особый случай. Я привожу эту историю с моим клиентом как пример хорошего тестирования юзабилити. Они провели необходимое тестирование. Они получили результаты. И они сделали правильные выводы из полученных результатов. Почаще бы компании так поступали.

Какова ситуация и что можно сделать

Что из этого следует для нас, бедных пользователей? Подытожу сказанное:

1. Вы не глупы. Интерфейсы пользователя действительно скверные, чего быть не должно.
2. Плохи интерфейсы, потому что их разрабатывают программисты, не понимающие разницы между собой и пользователями.
3. По причине, указанной в п.2, интерфейсы намеренно усложнены, и авторы считают, что пользователям это должно нравиться, что является ошибкой (см. п. 1).

Интерфейс пользователя можно сделать гораздо лучше, если привлекать специалистов по юзабилити в самом начале каждого программного проекта. Обычные программисты в этом отношении малопригодны, чтобы не сказать хуже. Кто-то должен встать на защиту молчаливого большинства пользователей, которым глубоко наплевать на технологию как таковую, которым просто нужно выполнить задачу и после этого заниматься своими обычными делами. Это я и пытаюсь делать на всех обсуждениях дизайна, где мне приходится бывать. «Вы, как те конструкторы дрелей для сети Home Depot, — говорю я программистам, — спорите, что лучше: шариковые подшипники, роликовые или пневматические, и каждый утверждает, что именно его решение нужнее всего на свете покупателю. Все не так. Покупателю не нужна ваша дрель сама по себе, нисколечко. Не была нужна и никогда не будет. Он идет в Home Depot не потому, что ему нужна дрель, а потому что ему нужны отверстия. Если бы он мог купить коробку отверстий для своей стены, он бы и не касался вашей дрели

и был бы вполне счастлив. (Помните Ринго в *Yellow Submarine*? «I've got a hole in my socket...» — «у меня в кармане дырка».) Ваша дрель — лишь неизбежное зло на пути вашего пользователя к отверстиям. А вот теперь спросите себя сами и честно ответьте: какие отверстия нужны вашему пользователю и как ваша программа может помочь ему получить их лучше, быстрее и дешевле?»

Прочтя эту главу, вы можете не хуже всякого другого объяснить производителю программного обеспечения, что вам нравится, а что нет. Структура интерфейса программы — это не откровение свыше, дарованное на горе Синай. Она создается на основе конструктивных решений программистов и других разработчиков, и эти решения вполне могут быть иными. Пишите им электронные письма, и почаще. Объясняйте, что вам нравится и что нет. Требуйте, чтобы они не совали вам под нос этот запрос подтверждения, а реализовали бы хорошую функцию Undo.

Больше всего на свете программисты боятся выглядеть глупыми. Они готовы быть уродами, импотентами, мучителями детей и животных, но только не тупыми. Да смилостивится к нам Господь. В их представлении о себе интеллект ценится превыше всего. Если вы хотите, чтобы программист что-то сделал, постарайтесь, чтобы он глупо выглядел на публике, если не сделает этого.

Итак, когда в очередной раз вы встретите плохой интерфейс пользователя, остановитесь и рассмотрите его. Поработайте с ним немного, выясните точно и конкретно, чем он вам не нравится и чего бы вам хотелось от него. Пошлите сообщение о плохом дизайне в Зал позора «Hall of Shame» — веб-сайт, специально созданный для таких целей. Можно начать с веб-сайта этой книги www.whysoftwaresucks.com. Затем пошлите письмо компании-изготовителю с извещением о том, как вы их выставили на позор. Чем крупнее их глупость, обнаруженная вами, тем больше они будут раздосадованы ее публикацией. Тогда, *возможно*, до них дойдет, что их пользователи не такие, как они сами.

ГЛАВА ВТОРАЯ

Запутавшись в паутине

Всемирная паутина огромна. (Разозленный читатель: «Это сколько я заплатил, чтобы узнать об этом?» Автор, подсчитывающий гонорар: «А что? Разве не огромна?») Система, которая изначально связывала несколько десятков ученых-физиков, превратилась в нечто, уже названное «становым хребтом человеческой расы». Сегодня персональные компьютеры используются главным образом для поиска информации в Сети, и она постепенно проникает в наши телевизоры, сотовые телефоны и даже холодильники.¹

Но потенциал Сети реализуется не полностью, поскольку многие программисты и дизайнеры, создающие сайты и управляющие ими, не понимают назначение Сети, принципы ее работы и решаемые задачи, даже если претендуют на такое понимание. Любой пользователь может рассказать десяток историй, связанных с неудачным опытом общения с Всемирной паутиной, аналогичным неудачному опыту работы с настольными приложениями. Вы полчаса вводите данные на нескольких страницах и в итоге получаете сообщение «сервер перегружен, повторите попытку позднее», теряя при этом все результаты своей работы. Либо браузер за-

¹ Интернет-холодильник LG GR-D267DTU оборудован автоматической морозилкой для льда, встроенным ПК с сенсорным экраном и высокоскоростным соединением с Интернетом. Лично я никогда им не пользовался, но он, судя по обзорам, хорошо интегрируется с подключаемыми к Интернету СВЧ, кондиционером воздуха и стиральной машиной того же производителя.

сыпает, не дав никакого ответа, и вы так и не знаете, удалось ли вам купить билеты на Брюса Спрингстина. И все это время вы боитесь, что какие-нибудь негодяи доберутся с помощью Интернета до ваших личных дел и украдут ваши деньги или идентификационные данные. В этой главе мы поговорим о том, с какими сложностями сталкиваются пользователи при работе с сайтами, аналогично тому, как в первой главе мы разбирались с настольными приложениями. В последующих главах мы обсудим безопасность и защищенность в обеих областях.

Многие не различают термины World Wide Web (WWW, Всемирная паутина, или Сеть, или Веб) и Интернет, хотя они означают совершенно разные вещи. Интернет — это сеть передачи данных самого общего типа, по которой можно обмениваться любыми данными между любыми двумя компьютерами на свете. World Wide Web — это специфический тип данных, распространяемых через Интернет, страницы, которые человек читает в браузере типа Internet Explorer или Firefox. Возникновение WWW поверх Интернета — грандиозная история. Попробую рассказать ее в четырех разделах, избегая технических деталей.

С чего это началось

Сеть зародилась в CERN, крупной научной физической лаборатории в Женеве. Работавшие в ней ученые стали ради экономии бумаги и места размещать документы, касающиеся своих исследований, на объединенных в сеть компьютерах, а читать эти документы на своих мощных и дорогих рабочих станциях они могли с помощью некоей программы — раннего варианта браузера. Затем в 1990 году некто по имени Тим Бернерс-Ли сказал: «Я часто вижу, что одни документы ссылаются на другие. Поскольку все они размещены в сети, почему бы не сделать так, чтобы при чтении можно было одним щелчком перескочить к упомянутому документу? Назовем такую ссылку, скажем, ... ссылкой („link“»). Разработчики документов договорились относительно текстовых символов, которыми следовало обозначать ссылку и задавать ее адрес, а также модифицировали программу-браузер, чтобы она реагировала на щелчки по ссылкам. Так появилась Паутина. Но она еще не стала всемирной.

Поскольку научные документы часто ссылались на документы из других лабораторий, ученые захотели соединить системы хранения документов CERN и других лабораторий. Относительно просто удалось заста-

вить системы хранения документов общаться между собой посредством Интернета — сети общего назначения, которая уже соединяла их и применялась в то время преимущественно для электронной почты. Теперь пользователи могли переходить из одного документа в другой без каких-либо хлопот и даже не зная, где физически располагаются эти документы. Вот так Паутина стала всемирной. Но тогда она еще не пользовалась популярностью.

Подлинный расцвет Сети начался примерно в 1994 году, когда количество серверов выросло от 500 в начале года до 10000 в конце. Произошло это отчасти благодаря усовершенствованной программе-браузеру под названием Mosaic, позднее развившейся в Netscape Navigator, разработанной опять-таки в научной физической лаборатории. Это был первый хороший (или почти приличный) браузер для таких недорогих платформ, как Windows, которая тогда как раз начала свое широкое распространение в домашних компьютерах и на рабочих местах. Учебные заведения и предприятия, уже пользовавшиеся электронной почтой Интернет, стали размещать в сети свои документы и читать чужие, а домашние пользователи начали заниматься тем же самым с помощью модемов для телефонных линий. При этом увеличение количества пользователей вызывало увеличение объема контента, а растущий контент привлекал новых пользователей. Так начался стремительный рост Всемирной паутины.

Как бы это ни было неприятно тем ученым, усилиями которых возникла Сеть, за ее феноменальным ростом стоят те же силы, что управляют практически любой (неакадемической) деятельностью людей: деньги и секс (часто объединяясь друг с другом). Благодаря Сети потребителю стало гораздо легче найти и получить любые данные, будь то текст, музыка, графика или что-либо еще, из любого места. Иногда владелец данных берет плату за сами данные. Распространители порнографии с огромным успехом эксплуатируют эту бизнес-модель, ибо даже сегодня лишь редкие пользователи готовы платить за какой-либо иной контент в Сети. Ряд организаций, среди которых выделяются «Wall Street Journal» (как считают некоторые, тоже порнография, но особого рода) и «Oxford English Dictionary», делают то же самое, но с ограниченным успехом, и основные компании звукозаписи могут придти к ней, если когда-нибудь проявят благоразумие. Но сегодня владелец данных чаще зарабатывает на том, что благодаря легкости доступа к этим данным через Интернет снижает издержки существующих бизнес-процессов, например, сокращая работ-

ников в системах заказа авиабилетов или слежения за доставкой почты. Вот так Сеть становится вездесущей.

Как это действует

Изначально предполагалось, что в Сети можно будет просматривать статические страницы, содержащие текст и иногда графику. «Статическими» я называю страницы, которые заранее — до запроса пользователем — создаются автором с помощью, скажем, текстового процессора, наподобие газетных или журнальных статей. Когда пользователь запрашивает веб-страницу, вводя ее адрес в браузере или щелкая по ссылке, сервер, на котором находится эта страница, считывает ее с диска и отправляет пользователю. У сервера запрос пользователя не вызывает никаких затруднений: он просто находит и возвращает статью, которую автор написал ранее, как на рис. 2.1.

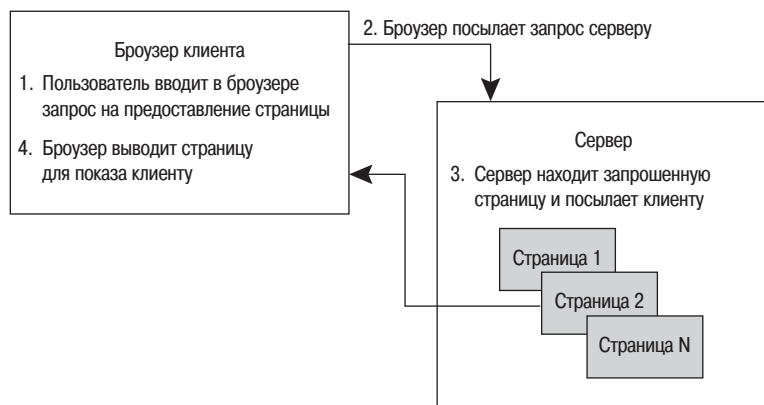


Рис. 2.1. Всемирная паутина, общение человека с человеком

В соответствии с этим механизмом Сеть обеспечивает общение одного человека с другим. Такие страницы можно сравнить с бумажными брошюрами, а Веб — со сверхскоростной почтовой службой. Эта простая архитектура очень полезна. Например, у моего местного кинотеатра есть небольшой сайт, на который я иногда заглядываю, чтобы узнать, что идет (например, «Город счастья, штат Техас» — в маленьком тexasском городке двух беглых заключенных по ошибке принимают за организаторов кон-

курса талантов), почитать о предстоящих событиях и найти ссылки на киноафиши и рецензии. Кинотеатру не нужно тратить деньги, губить деревья ради рассылки бумажных буклетов, а мне ломать голову, вспоминая, куда я их засунул и не выкинула ли их жена. Другой пример: когда в последний раз вы видели печатное руководство по компьютеру? Компании гораздо дешевле держать руководства в сети, чем печатать на бумаге, а пользователю гораздо дешевле загрузить документ из сети, чем оплачивать доставку почтой. Пользователи не должны ждать, пока придет документ; проще обновлять версии и легко хранить старые для тех, кому они могут понадобиться. Настоящие «гики» часто жаловались, что электронное руководство нельзя читать, лежа в постели, но развитие беспроводной связи вполне удовлетворительно решило эту проблему (но не семейные проблемы, вызываемые чтением компьютерных руководств в постели, — если только это не брак между двумя гиками).

Такой подход приемлем для многих типов документов. Расписания пригородных поездов, меняющиеся несколько раз в год. Газеты и журналы. Справочники служащего или студента. Ежегодные отчеты компаний. Буклеты, рекламирующие места отдыха, музеи и аттракционы. Горы правительственных постановлений. Еще более высокие горы порнографии. Возможно, наивысшим достижением Сети стало ее функционирование в качестве быстрой и недорогой почтовой службы.

Но людям, как известно, всегда всего мало. У той цепи в мозгу, которая требует «еще!», нет никаких ограничителей на входе. Как бы замечательно ни было то, что вы получили, через пять минут вы автоматически произносите: «Да, отлично. Еще, пожалуйста. И побыстрее». Рассматривая статические документы, пользователи задумались: «Если можно получить статью по физике или список кинофильмов на экране, а не на бумаге, почему нельзя посмотреть выписку по банковскому счету? Или обольстительные движения в танце живота?»

Пользователю непонятно, что особенного в этих его новых пожеланиях, но для программиста они представляют иную и гораздо более сложную вычислительную задачу. Банк не может каждый день создавать новые страницы для каждого возможного способа представления каждого существующего счета наподобие того, как это делается в газете для каждой статьи. Слишком много счетов и слишком много способов их просмотра: с сортировкой по номеру чека, или плательщику, или сумме; с сортировкой в порядке убывания или возрастания; показать данные за последние

30, 90 или 120 дней и т. д. Вместо этого сервер банка запрашивает у пользователя входные данные — в данном случае номер счета и вид желаемой сортировки, после чего находит счет в базе данных и создает заказную веб-страницу с подробным счетом, как показано на рис. 2.2.



Рис. 2.2. *World Wide Web, общение человека с компьютером*

Здесь уже происходит общение человека не с другим человеком, а с компьютером. Это деятельность совершенно иного рода. Существует — или существовало в прошлом — много видов работы, заключавшихся в разговоре с клиентом по телефону, вводе запроса клиента в компьютер и зачитывании клиенту по телефону полученного на экране компьютера ответа. Примерами таких операций служат слежение за перемещением почтового отправления в FedEx, покупка авиабилета и перевод денег с одного банковского счета на другой. Эти «контактные точки с людьми» обходятся очень дорого. Один мой клиент, работавший в финансовой сфере, сообщил мне как-то, что обработка одного денежного перевода оператором обходилась его компании в 25 центов, а самостоятельный перевод клиентом в режиме онлайн — в 10 центов.¹ «Если за день таких операций

¹ Кажется, что это невероятно много, если оператор зарабатывает каких-нибудь 10 долларов в час, но при ближайшем рассмотрении стоимость быстро возрастает. Необходимо оплачивать пособия, например страховые. Необходимо обучение. Необходимо подготовить замену на время отпуска или потери трудоспособности (см. страховку). Необходимо нанять инспектора и послать его на курсы обучения признакам сексуального домогательства и периодически отправлять на переподготовку (см. страховку). Необходим отдел кадров, который будет следить за проведением всех этих мероприятий и документировать их. Нужно арендовать здание, где все они будут сидеть, и оплачивать их счета за электричество и телефон. И т. д. и т. п.

совершается 5 миллионов, то суммарная разница весьма ощутима», — добавил он. Если вы звоните в FedEx по телефону, чтобы узнать, где посылка, и разговариваете с человеком-оператором, акционеры FedEx должны пожалеть, что вы вообще ее отправили, потому что эта сделка стала для них убыточной. Недорогие авиалинии, зарабатывающие сегодня деньги или, по крайней мере, теряющие их не так быстро, как прежние перевозчики, отличаются высокой долей билетов, продаваемых через дешевые сайты, и сокращением дорогих кассиров-людей. Тут возникает много способов заработать деньги для тех предприятий, которые смогут проявить достаточную проницательность.

А проницательность им необходима. Потому что, как я сказал, этот динамический случай представляет собой гораздо более сложную вычислительную задачу. Часто запрашиваются конфиденциальные данные, поэтому сервер должен установить личность пользователя (провести **аутентификацию**, см. главу 4) и решить, есть ли у него право доступа к запрашиваемым данным. Вы хотите узнать сальдо своего банковского счета, но не хотите, чтобы его узнала ваша разгневанная бывшая жена? Сервер должен также запоминать действия пользователя, переходящего от одной страницы запроса к другой, чтобы программа продажи авиабилетов могла в итоге правильно определить причитающуюся сумму в зависимости от выбранного маршрута следования. Сервер и браузер должны шифровать конфиденциальные данные, передаваемые по сети, чтобы ловкий сынишка вашего соседа не смог прочесть их с помощью перехватчика сетевых пакетов. И все это сервер должен сделать корректно, за считанные секунды, прежде чем вам надоест ждать и вы пошлете его к черту, сняв трубку телефона или перейдя на сайт конкурирующей фирмы.

Почему с отстоем до сих пор не покончено

Конечно, пользователи не знают о сложности вычислительных задач, которые должны решаться на сайте, а если бы и знали, их бы это несколько не взволновало. Их волнует только то, что они видят — лицо сайта, открытое публике, интерфейс пользователя. У веб-сайтов интерфейсы пользователя оказываются никуда не годными по той же причине, что и интерфейсы программ: веб-дизайнеры не знают своих пользователей, а потому предполагают, что те должны быть похожи на них самих. Но они не похожи, а дизайнеры этого не понимают, и результаты их непонимания

оказываются у всех на виду. Поскольку кажется, что спроектировать веб-сайт очень просто, люди, которые вообще не должны к нему прикасаться (менеджеры, торговцы) считают, что могут разработать хороший интерфейс: «Я знаю, что нужно пользователям, потому что я сам — пользователь». Это все равно, что считать себя способным залечить канал в зубе только потому, что вам пришлось на себе испытать эту операцию.¹ Программисты плохо справляются с разработкой интерфейса пользователя, но эти любители — еще хуже. И оправданий тому нет. На разработку хорошего веб-сайта уйдет столько же труда, сколько на разработку плохого. Но работать надо с умом, а это случается гораздо реже, чем хотелось бы.

Гуру веб-дизайна Винсент Флэндерс (Vincent Flanders) однажды написал у себя на сайте,² что главная причина плохих веб-разработок — «уверенность в том, что вы или ваш веб-сайт кому-то интересны». Он совершенно прав. Я втолковываю своим слушателям: «Вы не интересны пользователю. Не были интересны и никогда не будете. Ваша мама может заинтересоваться, если вытерпит вид вашего сайта дольше нескольких секунд, а кроме нее — никто». Если только разработчик сайта не создает его для своей мамы (а есть много семейных сайтов, разработанных именно с такой целью), он должен твердо усвоить, что пользователь пришел к нему на сайт не для того, чтобы осчастливить его. И даже его мама, желающая поглядеть на фотографии своих внуков, не станет долго терпеть плохо спроектированный веб-сайт.

Пользователь приходит на веб-сайт, чтобы удовлетворить какую-то свою потребность. Вероятно, он хочет а) найти какую-то полезную для своей повседневной деятельности информацию, например расписание движения поездов, маршрут проезда или местонахождение посылки; б) купить некий продукт или услугу, например книги, или музыку, или путешествие, или все мыслимое и неммыслимое на eBay; или в) развлечься — обычно с помощью людей, на которых не слишком много одежды.

Веб-дизайнеры часто повторяют ошибки дизайнеров настольных приложений, делая сложные вещи возможными, вместо того чтобы делать

¹ Бывшая судья Верховного суда США Сандра Дэй О'Коннор в своей автобиографии «Lazy B» описывает, как один скотник на ранчо ее семьи проделал это над собой. Из ее книги неясно, бросил ли он заниматься скотиной, чтобы открыть свое зубоврачебное дело.

² www.webpagesthatsuck.com

простые вещи простыми, и требуя от пользователя разбираться во внутреннем механизме, вместо того чтобы приспособлять интерфейс к образу мысли пользователя. Дизайнеры настолько увлекаются реализацией всех мыслимых и немислимых украшательств, что забывают о целях пользователей, которым вовсе не интересен процесс, а интересно получение результата с минимальными усилиями. Если бы можно было, щелкнув каблучками и произнеся волшебное слово, купить новую книгу для своей библиотеки или узнать, где в данный момент находится почтовая посылка, пользователь так и сделал бы, вместо того чтобы запустить свой компьютер и производить какие-то действия с веб-сайтом.

Разработчики веб-сайтов сталкиваются не только с такими классическими трудностями, но и с еще двумя, которые не встречаются разработчикам настольных приложений. Первая связана с тем, что посещение веб-сайта представляет собой событие, которое случайно — в гораздо большей мере, чем установка программы на компьютере. Фактически любой пользователь впервые попадает на веб-сайт с поисковой машины, или введя адрес, или щелкнув по ссылке, найденной где-то в рекламе. Пользователь изначально не знает, чем ему может быть полезен этот веб-сайт. Домашняя страница сайта должна зрительно сообщить об этом пользователю в течение нескольких первых секунд, пока он не скажет «да пошли вы...» и не перейдет на следующую ссылку в списке результатов поиска. Чуть позже в этой главе я расскажу о двух сайтах, из которых один хорошо справляется с этой задачей, а другой — плохо.

Вторая трудность — навигация по сайту. Настольная программа обычно оснащена панелью инструментов и меню, содержащим команды и расположенным в верхней части окна приложения, и пользователи знают, что там они найдут список команд. Команды тоже более или менее стандартизованы: Файл, Правка, Справка и т. д. Но при просмотре веб-сайта меню и панель инструментов принадлежат браузеру (Назад, История, Останов и т. д.), поэтому навигация и передача команд веб-сайту (перейти на страницу поиска авиарейсов, найти рейсы из Балтимора в Сент-Льюис в четверг) должны осуществляться каким-то иным образом. Структура навигации по сайту (куда я могу попасть в пределах этого сайта, как это сделать, в каком месте я смогу решить проблему, из-за которой пришел сюда) должна быть очевидна пользователю после 2–3-секундного взгляда, но такие сайты встречаются редко. Примеры, иллюстрирующие хорошие и неудачные в этом смысле решения, я также приведу.

Прежде всего хороший веб-сайт минимизирует объем работы, который должен выполнить пользователь, особенно связанной с самим сайтом, а не с задачей, за решением которой он пришел на сайт. Отличная книга по веб-дизайну Стива Круга возвещает эту идею своим названием: «Не заставляйте меня думать»¹ («Steve Krug, Don't Make Me Think», New Riders Publishing, 2000). Посмотрите, как одна группа разработчиков последовала этому совету, а другая — нет. Интересно, какой из сайтов вы бы предпочли?

Проект, ориентированный на клиента, против проекта, ориентированного на сервер

Всемирная паутина по определению является всемирной (прервите меня, если я слишком углублюсь в технические подробности). Пользователь, находящийся, скажем, в Швеции, получает доступ к Google так же легко, как пользователь, находящийся в Соединенных Штатах. По этой причине Google перевел свою домашнюю страницу на многие языки, от африкаанс² до зулусского, часто при помощи добровольных переводчиков.³ Когда пользователь в Швеции вводит в адресной строке браузера «www.google.com», сервер Google определяет, что запрос пришел из Швеции, с потрясающей сообразительностью решает, что пользователь, скорее всего, говорит по-шведски и автоматически посылает ему шведскую домашнюю страницу Google (рис. 2.3).

¹ Стив Круг «Веб-дизайн: книга Стива Круга или „Не заставляйте меня думать!“», 2-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2008.

² Африкаанс (бурский язык) — язык, родственный нидерландскому, на котором говорят живущие в ЮАР африканеры, или буры (люди белой расы, потомки голландских колонистов, высадившихся в 17 в. на мысе Доброй Надежды), а также «цветные», т. е. люди смешанного бурско-негритянского происхождения. Наряду с английским является официальным языком ЮАР. — *Примеч. ред.*

³ Можно только удивляться, что громадная и высокодоходная компания Google умудряется привлекать бесплатных добровольцев, чтобы делать свою работу, но это так. Должно быть, добровольцы считают, что Google никогда не станет тратить деньги, чтобы перевести свой сайт на их родные языки, такие как бирманский (по данным Google.com на момент написания книги работа выполнена на 14%), санскрит (68%) и клингон (96%).



Рис. 2.3. Домашняя страница Google.com по-шведски

На этой странице не только предлагаются услуги Google по-шведски, она содержит и прямую ссылку на англоязычную домашнюю страницу Google на тот случай, если человек, обращающийся к Google из Швеции, не говорит по-шведски, или если швед захочет попрактиковаться в английском, или Google неправильно определит страну, из которой пришел запрос. Все прочие языки Google находятся на расстоянии двух щелчков с помощью ссылки Språkverktyg или трех щелчков, если пользователь сначала пойдет на английскую страницу, где та же ссылка имеет название Language Tools. Если пользователь не станет менять язык интерфейса, Google автоматически запомнит выбранный язык на этом компьютере (с помощью cookies, о которых говорится в главе 5) и покажет домашнюю страницу на этом языке в следующий раз, откуда бы ни пришел запрос. (Работающим в Швеции гастарбайтерам из Малайзии и Таиланда эта функция нравится.) Выбор языка оказывается правильным почти всегда, почти для каждого пользователя, не требуя от него никаких размышлений и усилий, а чтобы исправить ошибочный выбор (на будущее), требуется от одного (как правило) до трех (максимум) щелчков мышью. Такая автоматизация повышает удобство пребывания в Сети для всех пользователей, которые обычно этого и не замечают, что служит примером хорошего

дизайна. Я выражаю высшую похвалу, доступную рецензенту: «работает, как должно».

Откуда Google знает, из какой страны пришел запрос? Каждый запрос веб-страницы всегда содержит так называемый **IP-адрес**, сетевой идентификатор компьютера пользователя. Это нечто вроде автоматического определителя номера при запросе веб-страницы.¹ Так же как коды стран в телефонных номерах, эти адреса раздаются интернет-провайдерам центральным административным органом. Так же как по коду страны можно узнать, где находится телефон, по IP-адресу можно выяснить, какому провайдеру он принадлежит и в какой стране находится. После этого сервер может послать страницу на языке, соответствующем этому местонахождению. Домашняя страница для многоязычной страны (Канады, Бельгии, Швейцарии) может при первом посещении предложить пользователю выбрать язык и запомнить его выбор или поступить еще изящнее — например, если запрос поступил из канадской провинции Квебек, то предложить страницу на французском языке, а если из Британской Колумбии — то на английском.

Для сайта этой книги (www.whysoftwaresucks.com) я написал образец страницы, демонстрирующей этот прием и определяющей страну пользователя. Я создал работоспособную страницу примерно за полчаса, так что это не очень сложно, но чтобы довести ее до профессионального уровня, пришлось бы поработать дольше.

Так какого же *черта* UPS.com не сделает что-нибудь аналогичное? Думаю, вы поразитесь, узнав, как UPS осложняет людям жизнь. На сайте этой службы сообщается, что United Parcel Service доставляет ежедневно 14 миллионов посылок, из которых 90% приходится на США. UPS.com получает 145 миллионов хитов в день, в среднем около десяти на каждую посылку. Пользователи могут следить за перемещением посылок и планировать отправку, не общаясь с человеком-оператором. UPS получает существенную экономию благодаря сокращению числа операторов, о чем я уже говорил выше.

¹ В главе 5 я расскажу, как помешать этой технологии определить ваш адрес, если вы хотите путешествовать по Интернету инкогнито (это не очень сложно, так же как и предотвратить показ функций caller ID номера вашего телефона).

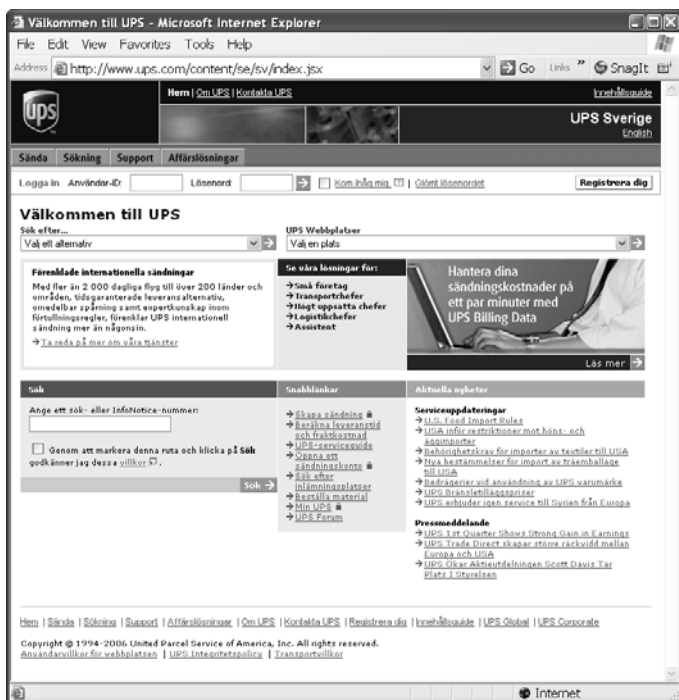


Рис. 2.4. Шведская страница UPS.com



Рис. 2.5. Глобальная домашняя страница UPS.com

Так как UPS осуществляет доставку в любую точку света, ей было бы естественно иметь сайт для пользователей в Швеции, и такой существует (рис. 2.4). Но попасть на него далеко не так просто, как на шведскую страницу Google. Если просто ввести в браузере «www.ups.com», находясь в Швеции или любом другом месте, вы увидите страницу, показанную на рис. 2.5, которая предлагает вручную выбрать свое местонахождение. UPS.com не определяет ее автоматически, как Google, и, в отличие от Google, вообще не станет с вами разговаривать, пока вы не укажете свое местонахождение. Чтобы попасть на шведскую страницу UPS.com, надо открыть окно выбора страны, щелкнув по нему, затем ввести S, чтобы перейти к нужной букве в списке, потом 26 раз нажать «стрелку вниз», чтобы добраться до Sweden (есть много стран, начинающихся с S), наконец щелкнуть по списку, чтобы закрыть его, и затем щелкнуть по стрелке «вперед», чтобы перейти куда надо. Только после этого вы увидите веб-страницу UPS для Швеции.^{1,2,3}

Не слишком любезное отношение к посетителям из Швеции, не так ли? Тридцать щелчков в интерфейсе пользователя, прежде чем просто начать то, зачем вы сюда пришли, — в сравнении с нулем щелчков на Google? Это не тот способ, которым компания может привлечь посетителей на свой веб-сайт, чтобы они не звонили дорогостоящим операторам. И UPS пренебрежительно относится к затратам времени не только своих

¹ Один большой гик заметил, что рассудительный швед мог бы нажать T и сразу попасть в Tadjikistan, а оттуда тремя нажатиями стрелки вверх переместиться до Sweden. Предложение добираться до Швеции через Таджикистан — полная противоположность проектированию под девизом «не заставляйте пользователя думать».

² Другой рассудительный швед мог бы попробовать сразу после S нажать W, чтобы перейти к SWeden, как это помогает в толковых программах типа Outlook. К сожалению, список стран на UPS.com глуповат и не может делать такие вещи. При вводе W он интерпретирует ее не как вторую букву слова, а как новую первую и переносит вас в страну под названием Wallis and Futuna (французская территория в южной части Тихого океана между Фиджи и Самоа — еще дальше от Швеции, чем Таджикистан).

³ Все-таки если нажимать клавиши (в данном случае S, w, e) достаточно быстро, то страну удастся выбрать сразу. Кроме того, список стран прокручивается при помощи мыши, т. е. 26 нажатий — это слишком даже для флегматика. — *Примеч. ред.*

шведских клиентов. Пользователям из США, составляющим 90% клиентов, нужно не менее четырех кликов, чтобы выбрать свою страну. Если по умолчанию просто показывать страницу для США, 90% пользователей будут избавлены от процедуры выбора, а остальным от этого хуже не станет.

Общаясь с человеком-служащим, мы бы такого не потерпели. Представьте себе, что вы приходите на почту, чтобы отправить посылку, и клерк сообщает вам: «Сначала нажмите 30 раз на эту кнопку и выберите язык, на котором вы хотите со мной разговаривать». Как минимум, вы скажете ему: «Послушай, если ты слабоумный и не понимаешь, что мы находимся в такой-то стране и я поздоровался с тобой на таком-то языке, почему бы тебе хотя бы не попытаться говорить на нем, пока я тебя не придушил?» Почему мы покорно терпим такие оскорбления от компьютеров? Совершенно не обязательно это делать.

Едва ли веб-программисты UPS так слабы, что сделали сайт, обрабатывающий 145 миллионов хитов в сутки, и не знали, что по IP-адресу пользователя можно установить страну. То есть они приняли это решение сознательно. Возможно, они по недомыслию решили обеспечить больший контроль за счет простоты применения, как мы видели это в предыдущей главе: «Вот круто! Из любой точки можно попасть на страницу для любой страны! Пусть у всех будет такая возможность!» А может быть, они, как многие программисты предыдущих поколений, решили взвалить на пользователей вину за все ошибки. Однако если хотя бы в каждом десятом из этих 145 миллионов обращений пользователь перейдет на страницу выбора страны и выбор страны займет в среднем пять секунд, то в UPS.com ежемесячно будет уходить из-за этой глупости целая человеческая жизнь. Они не ценят время своих пользователей. Возможно, им никогда и не приходило в голову, что его надо ценить.

Разработчики UPS.com скажут вам, что можно запомнить страну, установив флажок «запомнить этот адрес» на заглавной странице или поставив в браузере закладку. В последнем случае на закладки расходуется дефицитное место. В первом приходится запоминать страну, еще не увидев страницу, на которую вас приведет этот выбор (кому это надо?). Хуже всего, что в обоих случаях вам придется заниматься инфраструктурой сайта, а не тем делом, ради которого вы пришли на этот сайт. Таким способом невозможно произвести хорошее впечатление на нового клиента. Еще хуже придется пользователям в Швейцарии (от 32 до 34 щелчков,

в зависимости от выбранного языка) и Сирии (35 щелчков) — ну, это, конечно, если никто из них ничего не слышал о прокрутке. Если жители последней страны когда-нибудь обратят свою злобу не на газетные карикатуры, а на юзабилити веб-сайтов, то UPS не поздоровится.

Вы, кто делает сайты для UPS, дорогие мои, послушайте старого дядюшку Платта. У Google это сделано хорошо. У вас — плохо. У Google прекрасный выбор по умолчанию, у вас — никакого. Если пользователь выбирает не такой язык, какой задан по умолчанию, Google запоминает его автоматически. А вы требуете, чтобы пользователи что-то делали, чтобы вы запомнили этот язык. Вы не уважаете своих клиентов — людей, выпекающих хлеб и строящих жилье для ваших детей. Я знаю, что вам это безразлично, потому что вы гики, которые не умеют общаться с людьми (совершенно избыточное описание). Но я знаю другие слова, которые взволнуют вас гораздо сильнее: ваш дизайн выставляет вас дураками в сравнении с другими программистами. Пользователи, обыкновенные юзеры, издеваются над вами, потому что я показал им, какое глупое решение вы приняли. На вашем месте я бы исправил положение.

Кто поднимет мне веки?

В предыдущей главе я писал о потребности программистов осуществлять контроль, об их стремлении делать сложные вещи возможными, а непростые — простыми. Они часто требуют, чтобы пользователь разбирался в их внутренней модели программирования и приспособливался к ней, вместо того чтобы приспособливать свою программу к образу мыслей пользователя. Расскажу о двух сайтах, из которых один делает это правильно, а другой неправильно.

Сеть FedEx Kinko's располагает более чем тысячей копировальных салонов на территории США. Главное, за чем клиенты приходят на сайт компании — это выяснить, где находится ближайший к ним магазин. Веб-дизайнеры Kinko's максимально облегчили поиск этой информации. На домашней странице (рис. 2.6) клиента просят ввести почтовый код места, вблизи которого он хочет найти магазин (в нижней части рисунка, слева от центра; в цвете выглядит лучше). Страница результатов (рис. 2.7) показывает пять ближайших магазинов, контактную информацию, перечень услуг, предоставляемых в каждом магазине, географическую карту с нанесенным на нее местоположением магазинов и ссылку, по которой можно



Рис. 2.6. Домашняя страница FedEx Kinko's с простым определителем места

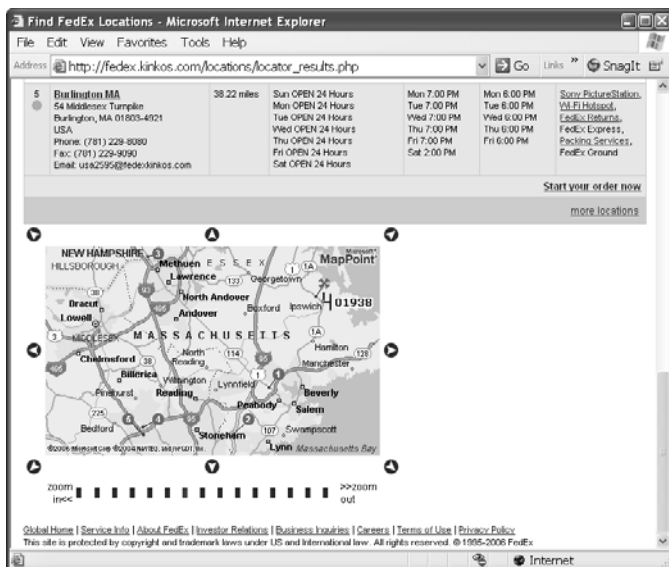


Рис. 2.7. Страница ответа определителя места магазинов Kinko's

получить сведения о том, как проехать к выбранному магазину. Ответить на элементарный вопрос «где ближайший магазин» очень просто, а ответы на более сложные вопросы, типа «есть ли в этом магазине нужные мне сервисы и как туда проехать», получаются автоматически при ответе на простой вопрос. Это ничуть не хуже, чем все виденное мной в этом роде, и близко к лучшему из всего, что я могу себе представить.

Сравните это с сайтом Starbucks.com известной сети кофеен. Тут тоже есть определитель места, но чтобы попасть на него, надо уйти с домашней страницы. Когда я захожу туда (рис. 2.8) и ввожу свой почтовый код 01938, страница результатов сообщает мне (мельчайшим шрифтом, хотя мой браузер настроен на показ самого крупного шрифта), что «в пределах заданной вами степени близости не был найден ни один магазин; если вы увеличите радиус поиска, могут найтись магазины в вашем районе; повторить поиск?» (рис. 2.9).

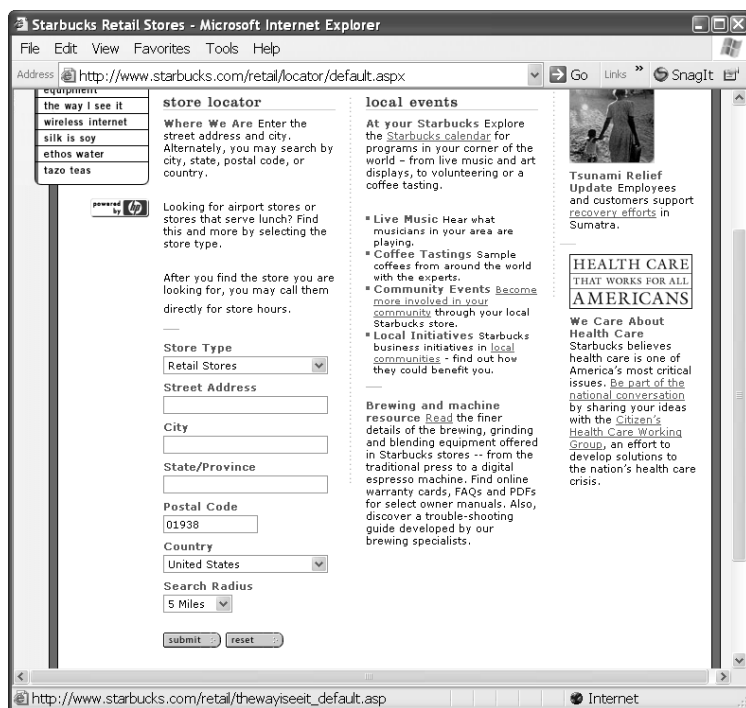


Рис. 2.8. Страница определителя места Starbucks.com с контролем радиуса поиска

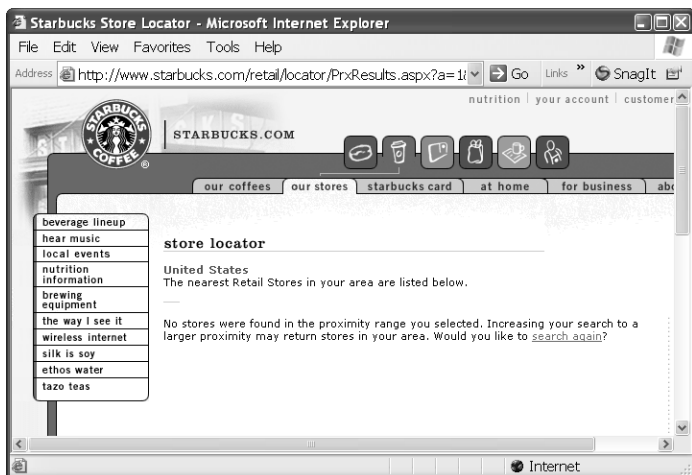


Рис. 2.9. Starbucks.com сообщает, что вблизи меня нет магазинов Starbucks

Какого черта? Я не выбирал никакого «радиуса поиска». Я просто ввел свой почтовый код. Эта кнопка с надписью «Радиус поиска» имеет какое-то отношение к «степени близости»? Если да, то почему не употребить одинаковые слова, а если нет, то о чем вообще речь? Почему шрифт такой, что невозможно читать, хотя я настроил браузер на тот шрифт, который я могу читать? А главное, почему вообще я должен думать обо всей этой чепухе? Я с утра еще не выпил кофе, поэтому я пошел на Starbucks.com, чтобы найти этот чертов магазин. Я задал простейший вопрос, а в ответ мне предлагаются многосложные слова, расшифровав которые, я смогу понять, почему мне не дали ответ, а потом я должен возиться с настройками, чтобы все-таки получить его. Почему просто не сказать, где находится ближайший магазин? Идите куда подальше, я лучше выпью чаю. И валиума. По-моему, он мне необходим.

Веб-программисты Starbucks допустили ошибку, выставив напоказ внутреннюю логику своей программы и заставив пользователей приспосабливаться к ней. Невидимо для пользователя оба веб-сервера, Kinko's и Starbucks, получают географические данные с третьего сервера — MapPoint, показанного на рис. 2.10. MapPoint знает, где находятся эти и многие другие предприятия, умеет находить предприятия вблизи указанного места, генерировать карты и описания проезда. Мне приходилось работать с MapPoint в приложении такого типа. Известно, что в подобных опера-



1. Пользователь посылает свое местонахождение на сайт Kinko's или Starbucks с просьбой сообщить ему, где ближайший магазин.
2. Веб-сервер запрашивает у MapPoint местонахождение магазинов и карты.
3. Веб-сервер получает у MapPoint информацию и представляет ее в виде красивой веб-страницы.
4. Веб-сервер возвращает пользователю страницу с перечислением ближайших к нему магазинов.

Рис. 2.10. Веб-сайты Kinko's и Starbucks, отыскивающие ближайшие магазины с помощью службы MapPoint

циях поиска MapPoint требует задать радиус поиска в пределах от 1 до 100 миль. Это разумно с точки зрения внутреннего программирования MapPoint, потому что для нахождения всех точек Starbucks на территории США потребуется много вычислений, а пользователям все равно нужны только одно-два ближайших местонахождения. По этой причине и появилась кнопка «Радиус поиска» в нижней части страницы определителя местонахождения Starbucks: она задает расстояние, в пределах которого будет вести поиск MapPoint. Kinko's автоматически определяет, каким должен быть радиус, и передает его MapPoint, не беспокоя пользователя,¹ а вот Starbucks заставляет пользователя думать и сбивается, если пользователь задаст неправильную (с его точки зрения) величину.

Наверное, программисты Starbucks считают, что чем больше контроля над поиском, тем мощнее и круче. На самом деле это бесполезно и раздражает. Никто не станет спрашивать: «А есть ли Starbucks в радиусе пяти миль? А десяти? Пятнадцати?» Зато многие интересуются по нескольку раз на дню: «Пить хочется, нужно встряхнуться, пусть даже это будет $2^{1/2}$ доллара за чашку эспрессо, которая стоит 15 центов. Где тут ближе

¹ Какое значение берет Kinko's и как они его рассчитали? Не знаю и знать не хочу. И вам оно ни к чему. Я просто хочу кофе. Обращаюсь к Starbucks: «Хватит задавать дурацкие вопросы, просто скажите, где кофе?»

всего Starbucks?»¹ Интерфейс пользователя Kinko's уважает ход мыслей клиента, а Starbucks — нет. Если Starbucks желает сначала поискать в радиусе 5 миль (в Нью-Йорке, для почтового кода 10021, это 166 магазинов) — прекрасно, но пусть делает это тихо, не требуя от пользователя вводить данные, и пусть автоматически расширит поиск, если все-таки ничего не найдет, и пусть не беспокоит пользователя, пока не сделает все возможное самостоятельно. Если и тогда не найдется магазинов, то пусть выразит пользователю сочувствие, скажем, напечатает примерно такой текст: «Бедолага, ни единого Starbucks в радиусе 100 миль. Вот тут ссылка — можешь сделать там заказ по почте, и завтра же мы пришлем тебе его с FedEx. (Их пункты легко найти даже там, где нет наших.) Или щелкни здесь и перейдешь в наш центр франчайзинга, где ты узнаешь, как открыть СОБСТВЕННЫЙ магазин Starbucks. Вам там такой явно необходим. Только не пропей всю прибыль». Может быть, программисты Starbucks, не в меру накофеинившиеся даже по программистским меркам, не могут себе представить, что на земле есть место, находящееся больше чем в пяти милях от магазина Starbucks. (Ну сколько еще мне повторять? Ваш. Пользователь. Не такой. Как Вы.)

Представьте себе такое письмо, отправленное Джиму Дональду, президенту Starbucks: «Дорогой Джим, мне пришлось три раза вызывать определитель местонахождения на вашем веб-сайте, чтобы найти ближайший магазин Starbucks, потому что этот магазин в 11 милях от меня. Желательно было бы сделать это за один раз. Сколько же мне биться головой об стену, чтобы вы взяли у меня мои деньги? Почему вы не сделаете все так же просто, как у Kinko's? Пусть ваш веб-дизайнер купит книгу «Софт — отстой! И что с этим делать?» и прочтет ее, если умеет читать. А я тем временем отправляюсь к вашему конкуренту Peet's Coffee».

¹ Как пишет Тим Хартфорд в книге «The Undercover Economist» (Тайный экономист), Oxford University Press, 2006: «Устанавливая весьма различные цены на продукты примерно одинаковой стоимости, Starbucks выявляет покупателей, которых мало заботит цена. Starbucks не знает точного метода выявления расточительных клиентов, поэтому предлагает им богатый выбор роскошных веревочек, с помощью которых можно удавиться.»

Очевидно, что — нет!

Предоставляемые сайтом функции и возможности должны быть очевидны для посетителя. Он сразу должен видеть, чем этот сайт может быть для него полезен и что от него для этого требуется. Лучший в мире пример тому Google (рис. 2.11) — вероятно самый признанный в Сети сайт. Пользователь видит, понимает и получает все мгновенно. Сайт прост и понятен — безо всяких вертящихся и анимированных логотипов, столь любимых этими типами из маркетинга (см. главу 6) и якобы «обеспечивающими узнаваемость нашего бренда». Сложные и редко используемые либо экспериментальные функции типа текстовых сообщений открываются через ссылку «еще...» и не отвлекают посетителя от базовых операций. Все настолько просто, что понятно даже при нахождении на шведской странице, которая была показана на рис. 2.3. Мне трудно себе представить, как можно улучшить эту страницу.

Полный контраст с Coke.com (рис. 2.12). Что можно сделать, попав на этот сайт? Видимо что-то можно, иначе бы его не было, но что? Непонятно. Надписи «The Coca-Cola Company», «Coca-Cola Worldwide» и «Coca-Cola In the USA» не говорят мне решительно ничего. Что делают эти кнопки, непонятно, пока не наведешь на них мышшь; прием, который

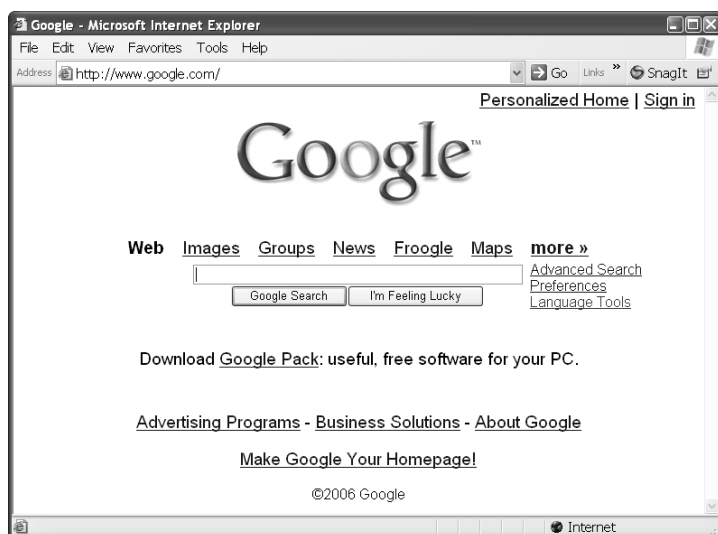


Рис. 2.11. Домашняя страница Google на английском: приятно посмотреть



Рис. 2.12. Домашняя страница *Coke.com*. Для чего этот сайт? Непонятно. Наверное, он что-то может, иначе бы его не было

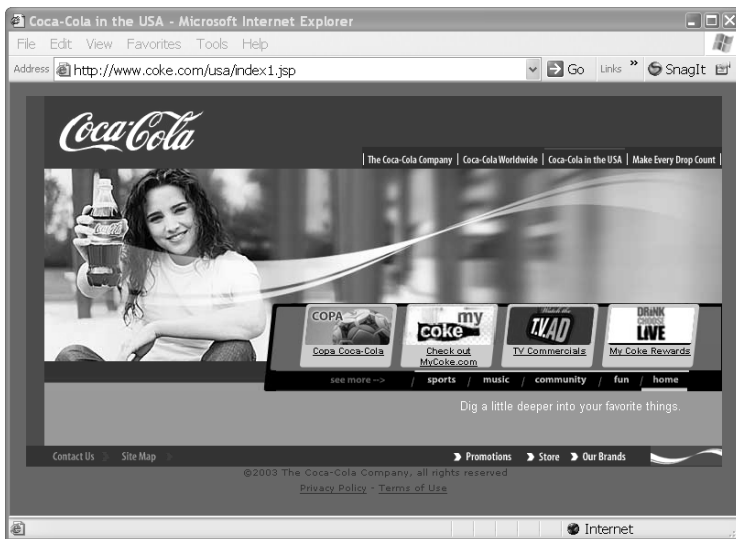


Рис. 2.13. Страница *Coke USA*. Все равно непонятно, что она делает и зачем она мне может понадобиться

Винсент Флэндерс называет «путешествием в неведомое». Даже если последовать надписи «USA» (рис. 2.13), все равно неясно, чем может быть полезен этот сайт. Ссылки для навигации размещены всюду — вверху, внизу, слева, справа и даже в центре. За исключением ссылки «TV ads», трудно догадаться, что они делают, и мне жаль тех, кто испытывает такую нехватку TV-рекламы, что ищет ее и тут. Есть ли тут бесплатная музыка, контракты по закупке газированной воды, спонсорство, сведения о пищевой ценности (как же!), сведения об Олимпийских играх, на поддержку которых Соке тратит так много средств? Что-то не видно. Нет и окна для поиска или ссылки на него, так что и этим способом не найти того, что нужно. Непонятно вообще, с какими целями сделан этот сайт.

Полная противоположность сайту главного конкурента — Pepsi.com. На первой же странице (рис. 2.14) полно всего. Музыка. Спорт. Реклама. Опровержение ложных слухов. Химический состав, рингтоны, фильмы. Окно поиска, если не видно, где лежит то, что мне нужно. В прежней



Рис. 2.14. Видно, что может сделать эта страница. Возможно, ничего из этого мне не требуется, но я хотя бы вижу, что тут есть

версии разыгрывались пожизненные билеты на Суперкубок.¹ Я вижу, чем этот сайт может быть мне полезен и где начать искать то, что нужно. Этот дизайн гораздо лучше. Здесь тоже не все хорошо, о чем скажу позже, но, по крайней мере, на главной странице видно, чем можно заняться на сайте.

Винсент Флэндерс считает, что в Соке путают веб-дизайн с сексом. «В реальном мире предварительная любовная игра — это необходимость. Надо создать настроение, продемонстрировать нежность и обольстительность. Чудесно. Но во Всемирной паутине нет места предварительным играм. В них нет необходимости. Они мешают. Грубо говоря, „бах-трах-спасибо“. Посетителей сайта не нужно соблазнять или создавать им настроение. Они пришли с определенной целью, и чем быстрее получают то, за чем пришли, тем приятнее им будет».

В Google это понимают. Я прихожу на сайт Google, чтобы найти другие сайты, и получаю их быстро и без лишней суеты. Соке занимается ерундой, скрывая главное (если там есть, что скрывать). Перси поступает гораздо лучше, с одним исключением. В книге этого не видно, но центральная картинка на сайте Перси вращается так, что у меня голова кружится. На эту неприятную особенность мы теперь и обратим особое внимание.

Заставки, Flash и анимация

Две худшие технологии в арсенале веб-дизайнера — это заставки и анимация. Страница-заставка (splash page) — это домашняя страница, которая только отображает сообщение «вы попали на наш сайт» и ничего больше. Полезная работа, которую может выполнить сайт, — его «настоящая» домашняя страница — становится доступной после перехода по ссылке со страницы-заставки. Примером служит страница Соке (рис. 2.12). Страницы-заставки попусту тратят время пользователя. Много ли посетителей было бы у Google, если бы после ввода в браузере адреса «Google.com» они попадали на страницу, сообщавшую: «Здравствуйтесь, спасибо, что вы выбрали Google. Правда, мы молодцы? А теперь щелкните вот здесь, и вы попадете на страницу, где можно заняться поиском». Ни у каких сайтов не должно быть таких страниц. Никогда.

¹ Матч за звание чемпиона Национальной футбольной лиги (NFL) Соединенных Штатов. Игра и сопутствующее ей празднование стали национальным праздником в США. (http://ru.wikipedia.org/wiki/Супер_Боул).

Точно так же очень часто злоупотребляют анимацией. Обычно для ее воспроизведения требуется подключить к браузеру штуковину под названием Macromedia Flash Player, хотя есть и другие технологии. Если верить одному бывшему сотруднику Macromedia, первоначальным предназначением Flash был показ порнографии — вибрирующего танца живота, о котором я уже рассказывал. Мой редактор не даст глубже развить эту идею, но я думаю, что вы сами в состоянии все прекрасно представить себе. Помимо данного великолепного и достойного применения, есть еще ряд сценариев веб-дизайна, в которых видео идеально подходит для реализации того, за чем пользователь пришел на сайт. Классическим примером служит изготовитель мебели Ikea, сайт которого гораздо более понятно показывает, как выполнить сборку мебели, чем это могла бы сделать любая печатная инструкция. Отлично. Сказочно. Замечательно. Да будет благословенна Сеть.

Но гораздо чаще дизайнеры раскидывают повсюду анимацию только ради нее самой. Многие рекламные объявления на веб-страницах применяют эту технику только потому, что движущиеся объекты притягивают человеческий глаз. Вот скриншот сайта Slate.com (рис. 2.15), на котором

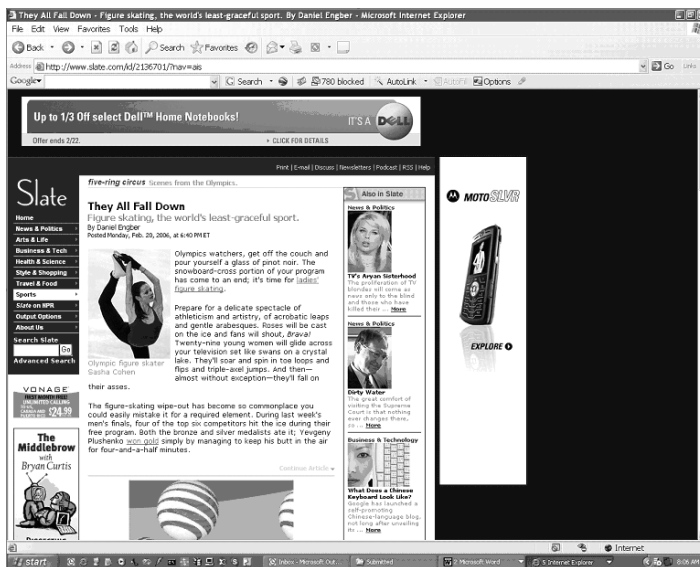


Рис. 2.15. Три плюющих рекламы на одной странице. Попробуйте почитать статьи во время этого представления

баннер Dell сверху, сотовый телефон справа и шары с логотипом AT&T непрерывно дергаются, в то время как я пытаюсь прочесть статью. Я не желаю, чтобы моя газета подпрыгивала, когда я ее читаю. Это особенно противно, если с утра я не выпил кофе (напоминание о неудаче с поиском магазина Starbucks, описанной выше). Злоупотребление движением на экране нервирует пользователей и вынуждает их запрещать показ анимации в настройках своих браузеров, причем этот запрет распространяется как на вредную анимацию, так и на полезную. К счастью, есть продукты, позволяющие легко включать и выключать анимацию. Это делают многие фильтры рекламы и специализированные продукты, например Flashblock с mozdev.org и No! Flash с Baryon Lee.

Как ни плохи страницы-заставки и ненужная анимация сами по себе, но ужас, производимый их соединением на одной странице доводит пользователей до иступления. Обозреватель Джулия Липман (Julia Lipman) в информационном бюллетене «digitalmass.com» описывает «множество вычурных страниц-заставок, не выражающих ничего, кроме склонности разработчиков потакать своим слабостям». Специалист по дизайну интерфейса пользователя Джаред Спул (Jared Spool) пишет в веб-колонке: «Когда наши клиенты начинают разговоры о Flash-заставках, мы даем им следующий совет. Пойдите к своему местному супермаркету и прихватите с собой клоуна. Пусть этот клоун встанет перед супермаркетом и, как только какой-нибудь покупатель попробует войти внутрь, разыграет перед ним на пару минут маленькое представление, в котором поприветствует пришедшего клиента и разъяснит ему, что хлеб находится в проходе номер шесть и в продаже сегодня есть молоко. Встаньте при этом в сторонке и подсчитайте, сколько человек станет смотреть на клоуна, сколько постарается проскочить мимо него, а сколько даст ему пинка».

Самая назойливая из виденных мной страниц принадлежит Howard-Systems.com (рис. 2.16). Шоу продолжается около минуты, при этом надписи бегают туда-сюда, и пока оно не прекратится, нельзя ничего делать (можно щелкнуть по ссылке «пропустить введение», но она едва заметна). Нельзя даже войти на сайт, щелкнув в середине, что первым делом пытаются сделать большинство посетителей. Для сайтов такого типа был даже сочинен термин *Flash-турбация*. Компания явно сильно потратилась на разработку, и кто-то из высшего руководства, взглянув на результат, высказался в духе: «Прекрасно! Пусть так и будет, людям это должно понравиться». Чем этот парень обкурился? На каких наркотиках сиде-

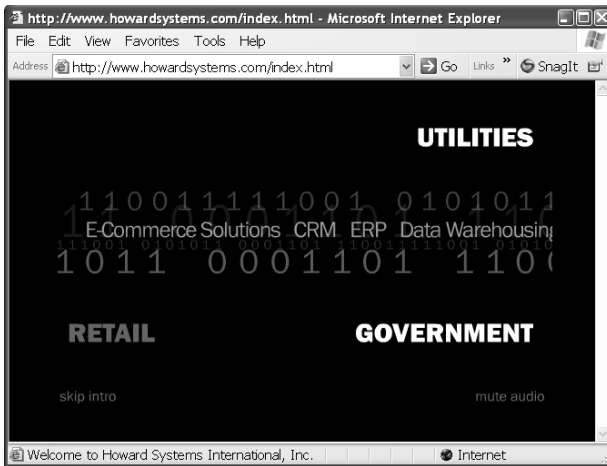


Рис. 2.16. Flash-заставка HowardSystems.com. Худшая из всех, что я видел в жизни, и классический пример flash-турбаии

ла его мать во время беременности? И что за слоган появляется после всей этой отнимающей ваше время ерунды? «У вас есть задача, которую нужно решить **НЕМЕДЛЕННО!**» Да, черт побери, поэтому заткнитесь, прекратите свои прыжки и уйдите прочь, чтобы я мог решить свою задачу!

Тестирование на живых организмах

Веб-сайт необходимо проверять на его фактических пользователях — так же как и другие программные приложения, — и это нетрудно сделать. Нужны лишь видеокамера, какая-нибудь приманка для пользователей (бесплатное пиво, например) и немного терпения. Даете пользователям задание, которое требуется выполнить, смотрите за их действиями и отмечаете, где и как у них случаются неудачи. Как любит повторять Джаред Спул, «Мы просто живем по средствам и ищем проблемы».

Трудно поверить, но иногда невозможно заставить дизайнеров протестировать интерфейс пользователя. Рецензент, назвавшийся «inhollywood», написал на Amazon.com по поводу книги Спула:

«Мне недавно довелось столкнуться с дизайнером крупного коммерческого сайта, которому предложили понаблюдать за пользователями в процессе их работы. Дизайнер отказалась со словами „...мне неинте-

ресно, что люди думают о моем дизайне. Я делала разработку, имея в виду конкретную цель, которую, считаю, достигла. Что нового я могу узнать?»

Удивительное высокомерие! И что же? Пользователи, тестировавшие сайт, заявили, что он запутан, сложен в навигации, труден в поиске и им не хотелось бы с ним работать. Я думаю, что если ее цель была в том, чтобы люди искали другие сайты, то она ее достигла.»

Даже соглашаясь с необходимостью тестирования, дизайнеры плохо его проводят. Они тестируют дизайн на самих себе или на своих ближайших коллегах. Неудивительно, что все проходит «на ура» («мы все прекрасно справляемся с ним, стало быть, все сделано правильно, молодцы ребята!»), но с треском проваливается, когда подключаются реальные пользователи. Разработчикам сайта заранее известно, для чего создан сайт и что он может делать, поэтому они не могут себе представить, каким он кажется тем, кто видит его впервые. Но ведь первейшая задача дизайнера — в течение нескольких секунд дать новым посетителям представление о сайте, и если она решена плохо, то никто уже не станет искать, что еще есть на этом сайте. Дизайнер изначально умеет перемещаться по сайту, поэтому он не может проверить, понятна ли система навигации незнакомому с ней пользователю. Дизайнер заранее знает, как пользователь должен вводить данные — например, не вводить пробелы в номерах кредитных карт, — поэтому он не сможет определить, ясно ли это будет пользователю. И т. д. и т. п.

Разработчики сайтов часто не умеют составлять задачи для тестирования пользователями. Я слышал, как они просят «рассказать, что понравилось и что не понравилось на сайте», вместо того чтобы дать реальную задачу, с которой пользователь может прийти на сайт, вроде «заказать билет на Таллахасси на июль с самой низкой ценой». Джаред Спул пишет о тесте, который он однажды провел для Ikea.com. Дизайнеры Ikea предложили своим испытателям найти книжный шкаф. Естественно, пользователи ввели в окно поиска «книжный шкаф» и быстро нашли нужный товар. Спул же предложил своим пользователям другую задачу: «У вас больше 200 художественных книжек, лежащих в коробках, разбросанных по комнате. Найдите способ привести их в порядок». Пользователи повели себя неожиданным образом. Они стали бродить по торговым залам, вместо того чтобы прибегнуть к текстовому поиску, а если иногда искали

слова, то не «книжный шкаф», а «полки». Пользователям пришлось гораздо труднее, потому что их мысленная модель была совсем не такой, как у дизайнеров сайта. Дизайнеры не узнали бы об этом, если бы не догадались провести тестирование и пригласить посторонних, которые задавали непривычные вопросы.

Иногда мерзопакостность веб-страницы бывает очевидна всем, кроме идиотов из маркетинговой службы. Например, Спул выяснил при тестировании, что анимационный логотип на одном сайте Disney настолько раздражал пользователей, что они «сначала пытались убрать его с экрана, прокручивая страницу, а когда это не удавалось, закрывали рукой, чтобы можно было прочесть текст». Я бы заплатил, и не раз, чтобы увидеть, как этот фильм показывают президенту Disney. Полагаю, он надел бы на дизайнера этого сайта смирительную рубашку и заставил его проходить аттракцион «It's a Small World After All» (Маленький мир)¹ два дня подряд без перерыва, прежде чем позволить ему утопиться.

Что мы можем сделать

Веб-сайты существуют для обслуживания своих пользователей. Вы теперь видите, что можно проектировать хорошие сайты и иногда на них попадать. Если сайт плох, значит, дизайнеры не понимают своих пользователей, хотя должны бы. Мой опыт показывает, что большинство веб-дизайнеров действительно хочет сделать доброе дело и создать хороший сайт. Так помогите им. Обычно на сайте есть ссылка, пройдя по которой, можно послать мнение о сайте его владельцу. Часто хозяин очень внимательно относится к толковым отзывам. Чем они конкретнее, тем лучше, особенно если вы сможете указать на отсутствие желательной вам функции, которая есть на каком-то другом сайте.

Например, недавно я воспользовался такой ссылкой на Alaskaair.com и послал следующее замечание: «Вы показываете рейсы из Бостона в Сиэтл только отсортированными по цене билета. Я выбрал вашу компанию, потому что никто больше не предлагает беспосадочных перелетов по этому

¹ Аттракцион «Small World» объединяет множество персонажей, «поющих» знаменитую песню «It's a small world (After All)», в которой прославляются народы, объединенные глобальной гармонией. http://disneyland.disney.go.com/disneyland/en_US/parks/attractions/detail?name=itsasmallworldAttractionPage

маршруту, но мне трудно найти их в длинном списке всех имеющихся рейсов. Другие сайты, например Orbitz, позволяют ограничить выбор беспосадочными рейсами или хотя бы показать их отдельно, где их можно четко видеть. Вашим сайтом не так удобно пользоваться, как другими, поэтому я склоняюсь к тому, чтобы снять телефонную трубку и поговорить с оператором заказа билетов, который дорого вам обходится. Надеюсь, это письмо поможет вам улучшить свой сайт. Не теряя времени».

Через несколько месяцев после моего письма сайт был исправлен, как показано на рис. 2.17. Теперь на нем есть переключатель, позволяющий сортировать рейсы по количеству посадок, времени вылета или прилета, либо цене. Они сделали даже больше, чем я предложил, добавив другие варианты сортировки с удобным доступом. Конечно, найдется много клиентов, которые захотят отсортировать рейсы по цене, чтобы найти самые дешевые, а другим потребуются первый и последний рейсы в дни вылета и прилета из путешествия соответственно.

Я не знаю, почему в Alaska Airlines сделали это — благодаря моему письму, вопреки ему, и читали ли они его вообще. Но если не сообщить компании, чего вам не хватает на ее сайте, нечего и удивляться, что вы это-

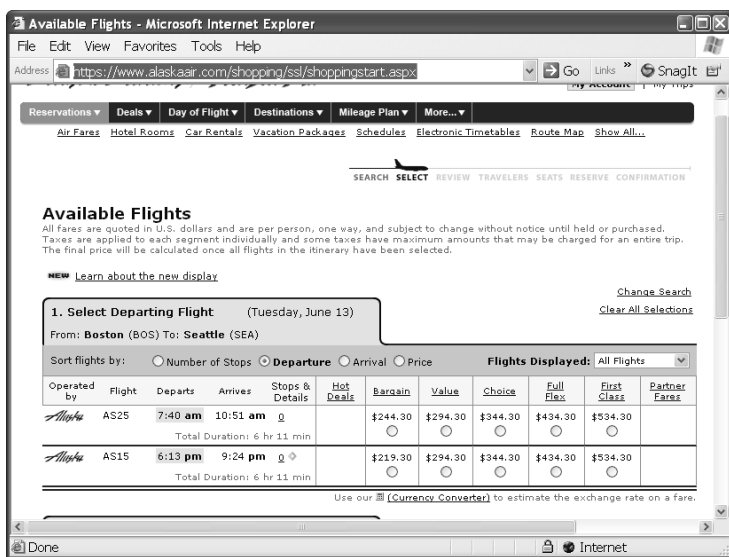


Рис. 2.17. Веб-сайт Alaska Airlines с улучшенными после сделанного замечания функциями

го не получите. И конечно, иногда вы не получите этого, что бы вы ни говорили. Иногда вам нужно прямо противоположное тому, чего хотят 10 других пользователей. А иногда владелец сайта просто не обращает на вас внимания. Я посылал свое мнение и в Starbucks, и в UPS, и ни та ни другая фирма не сделали ничего по поводу описанных мной в этой главе проблем. У меня есть предчувствие, что эта книга все же привлечет их внимание.

Иногда хозяин сайта не проявляет желания получать отзывы. И опять лучшим (или худшим?) примером служит UPS.com. Чтобы добраться до формы, предназначенной для ввода отзыва, надо продраться через пять отдельных экранов с загадочными именами,¹ затем заполнить семь отдельных разделов, и только тогда вам позволят ее передать. (Я помещу эту ссылку на сайте данной книги, чтобы вам было проще завалить их своими отзывами.)

Вежливость всегда полезна, но иногда непросто удержаться от крепких выражений по поводу самых грубых ошибок, особенно если они крадут мое время, когда теряется моя работа. Не удивлюсь, если UPS.com получает в день 100 грубых писем из Швеции, которые начинаются словами «Ушг site съкс, ушг dumb esses». Разработчики сайта не понимают, чем плоха их работа, иначе бы они переделали ее. Простым криком «ваш сайт — отстой» обычно ничего не добьешься, даже если он справедлив (и особенно, если справедлив).

Исключение — если это делает публично признанный авторитет, на сайте с высоким трафиком. Для этой цели существуют сайты «Зал позора», прародителем которых является WebPagesThatSuck.com, руководимый Винсентом Флэндерсом, неоднократно цитированный мною в этой главе. Его превосходная книга «Web Pages That Suck» (Отстойные веб-страницы), Sybex Inc., 1998, написанная в соавторстве с Майком Уиллом (Mike Wills), и ее не менее превосходное продолжение «Son of Web Pages That Suck» (Наследник отстойных веб-страниц), которую он напи-

¹ Зайдите на www.ups.com и выберите свою страну. Теперь выберите маленькую ссылочку *Contact UPS* внизу страницы. Теперь выберите *Internet Technical Support*, а затем *Contact Internet Technical Support*, по какой-то причине настолько отличную от *Internet Technical Support*, что ей нужна собственная страница. Теперь выберите *Email UPS Internet Technical Support*. Заполните форму и нажмите кнопку *Submit*. Не кажется ли вам, что там не очень-то ждут ваших сообщений?

сал в одиночестве, содержат рекомендации, игнорируя которые веб-разработчики подвергают себя большому риску.¹ Вы можете назвать действительно плохие сайты для его веб-колонки «Daily Sucker» (Ежедневный отстой), в которой он ежедневно дает разнос какому-нибудь сайту, разъясняя общественности, что именно в нем плохого и почему. Интересны ощущения веб-мастера сайта, получившего такую награду. Флэндерс не посылает электронные письма на преступные сайты и пишет, что «...некоторые психи любят посылать почтовые сообщения, сообщающие владельцу, что его сайт стал лауреатом премии самой плохой веб-страницы». Можете считать меня психом (вы ведь давно уже это поняли?). Номинируя какой-нибудь сайт, я делаю это, потому что он вызвал у меня раздражение, и когда он получает такую награду, я хочу ткнуть в это носом владельца, умышленно и со злобными намерениями. И я стараюсь послать свое письмо президенту компании, иногда не поленившись напечатать его на бумаге и отправить срочной доставкой UPS (если я не в Швеции в это время). «Примите мои поздравления! — сообщаю я. — Ваш веб-сайт действительно ужасен. В самом деле, во всем Интернете Винсент Флэндерс не смог найти хуже для сегодняшней демонстрации. Итак, вы увековечены на WebPagesThatSuck.com! Приятного дня». Это должно вывести их из полусонного состояния и заставить подумать о чашечке кофе. Если бы только они могли найти поблизости Starbucks... Увы, магазин в шести милях, и Starbucks.com не смог им помочь. Жаль, что Kinko's не торгует кофе. Интересно, справились бы они с этим по-шведски?

¹ Очевидно сходство названий этих книг с данной. Единственная моя связь с Флэндерсом и Уиллзом состоит в том, что я с увлечением читаю их книги и рекомендую другим тоже ознакомиться с их работой.

ГЛАВА ТРЕТЬЯ

Защитите меня

Червь Slammer! Вирус Code Red! Задраить люки! Выключить машины в тот день, когда должен сработать вирус Michelangelo! Шпионские программы регистрируют все ваши нажатия на клавиши и крадут пароли и номера кредитных карт. Взломщики захватывают управление вашим модемом и делают 900 телефонных звонков, которые вам приходится оплачивать. Дня не проходит, чтобы мы не услышали об очередной неприятности, которая может произойти на нашем компьютере.

Когда в детстве я слушал «Петю и Волка», то доводил своего младшего брата до слез, говоря ему, что волк сейчас выйдет из записи и съест его. Пожалуй, я предпочел бы этого хищника тем негодьям, которые могут пролезть через Интернет даже в спальню моей дочери. Страшно? Должно быть страшно. Очень страшно.

В этой главе нет конкретных инструкций по обеспечению безопасности компьютера или сети. Есть много хороших статей об организации межсетевых экранов (брандмауэров), об антивирусных средствах и т. д.¹ Вместо этого я познакомлю вас с процедурой оценки безопасности и встраивания ее в программы. Я расскажу вам об известных препятствиях, которые приходится преодолевать разработчикам программ, и как они справляются с этим (или терпят неудачу). Я надеюсь научить вас самостоятельно отли-

¹ Многие лучшие статьи принадлежат Computer Emergency Response Team в Университете Карнеги-Меллона и лежат на www.cert.org.

чать то, что имеет отношение к безопасности, от чепухи, которую несут безответственные производители в расчете скрыть тот факт, что они не разбираются в том, о чем говорят, и всучивают вам опасные программы.

Как было раньше

В мире всегда были и, вероятно, всегда будут негодяи, готовые вам навредить. Иногда причины бывают экономические: наркоманы хотят украсть из вашей машины радио, чтобы еще раз уколоться, а принадлежащие преступникам автомастерские готовы украсть машину целиком. Иногда причины идеологические: фанатик экологии хочет вывести из строя ваш Hummer, чтобы он больше не смог задавить какую-нибудь пятнистую сову, а террорист хочет поставить на колени Большого дьявола. Но иногда негодяи готовы разбить ветровое стекло только для того, чтобы показать, что они могут это сделать. Подобные риски нельзя устранить полностью, но их можно снизить до приемлемого уровня, для чего, как правило, достаточно избегать мест, где встречаются такие люди. Если же вам приходится там бывать, то вы платите деньги за охраняемую парковку или хотя бы ищите место, которое хорошо освещено и находится на глазах у публики. Если и это невозможно, вы берете такси или едете туда в таком драндулете, который даже Армия Спасения не примет в качестве жертвоприношения.

В те времена, когда настольные компьютеры только появились, преступникам было не так легко в них забраться. И это было хорошо, потому что у ранних версий DOS и Windows, а также выполняемых в них приложений, защиты не было вообще. Разработчиков программного обеспечения — или, точнее, его пользователей — удовлетворяло, что, пока к персональному компьютеру ограничен физический доступ, преступники не могут проникнуть в него и нанести ущерб. Компьютер оказывался примерно в таком же положении, как любое другое дорогое электронное устройство, например телевизор с большим экраном, и защита его осуществлялась аналогичными известными способами. В худшем случае вам пришлось бы купить новый компьютер, если бы старый украли.

И этот подход был эффективным, поскольку компьютеры использовались в основном для выполнения автономных приложений типа игры Zork в текстовом режиме. Соединения с другими компьютерами были редки. Если же они случались, то выполнялись прямым и относительно безопас-

ным образом. Например, когда в 1990 году я пользовался автоматической платежной системой CheckFree, мой компьютер набирал через модем телефонный номер CheckFree. Чтобы перехватить транзакцию, преступнику было необходимо прослушивать аналоговую телефонную линию, что слишком трудно и опасно в сравнении с небольшой денежной суммой, которую удалось бы украсть.

Даже когда в корпоративной среде повсеместно распространились сети и компьютеры стали тесно связаны друг с другом, особых проблем безопасности не возникло, поскольку программы и данные обычно брались из надежных источников. Сотрудники информационных отделов устанавливали новые программы, и вам не приходилось беспокоиться, нет ли среди них программ-шпионов (помимо тех, с помощью которых ваш босс мог следить, не сачкуете ли вы, но тут уж ничего не поделаешь). Документы или почтовые отправления, с которыми вы работали, поступали от ваших же коллег, поэтому можно было надеяться, что в них нет программных закладок (исключая непредумышленные баги, допущенные программистами, — с ними и сейчас ничего не поделаешь). Приходилось ограничивать доступ к определенным конфиденциальным данным, таким как ведомости зарплаты, но тогда было значительно легче это сделать, чем сегодня, и часто решение состояло в размещении конфиденциальных данных в отдельной среде, не связанной с общей сетью компании. (Редактор не позволил мне привести ту фразу, которую произнес инспектор моей компании, когда я попросил его открыть мне учетную запись на его автономной системе заработной платы.)

Почему с отстоем до сих пор не покончено

С появлением Интернета все изменилось. Каждый компьютер в сети оказался соединенным со всеми другими компьютерами в мире. Порядочным людям и, что неудивительно, преступникам тоже стало невероятно легко обмениваться данными друг с другом.

Преступники, действующие в киберпространстве, стремятся нанести ущерб вашему компьютеру по тем же причинам, по каким они желают покалечить ваш автомобиль: чтобы заработать денег, осуществить некую социальную акцию или вообще без причины. Они добиваются этого, запуская свои программы на вашем компьютере, похищая ваши секретные данные или комбинируя то и другое. Есть три основные причины, по которым

такие преступления легче совершить и труднее предотвратить, чем в реальном физическом мире.

Во-первых, благодаря Интернету расстояние перестало служить помехой преступлению. Преступнику, который находится, скажем, в Румынии, не так просто взломать вашу машину, находящуюся в Нью-Йорке, но Интернет не задумавшись направит его зловредные данные на ваш компьютер, если указать его адрес или вы сами нечаянно запросите их, думая, что они представляют интерес и не опасны. Если данные составлены определенным образом и дверь не заперта, он может вломиться к вам так же легко с расстояния в 5000 миль, как если бы находился за соседней стенкой. Френсис Кэйрнкросс (Frances Cairncross) написала книгу под названием «The Death of Distance» (Расстояние умерло), Harvard Business School Press, 2001, в которой обсуждала последствия дешевого и быстрого обмена данными через Интернет, но, кажется, в их числе не было того, о котором идет речь.

Во-вторых, интернет-атаки требуют от атакующего гораздо меньше времени, чем в физическом мире. Чтобы разбить стекло и выгащить радио, нужно несколько минут, и еще несколько часов, чтобы продать его и купить дозу. Взлом компьютера может занять менее секунды. Преступник, действующий в Интернете, может в единицу времени атаковать гораздо больше целей, чем автомобильный вор, что увеличивает вероятность обнаружения уязвимого объекта.

В-третьих, сам Интернет позволяет средствам атаки распространяться быстрее, чем это удастся им в физическом мире. Например, взломщику сейфов трудно передать остальным ворами свое представление о кулачках в замке. В Интернете же достаточно одному преступнику придумать способ взлома, после чего он может сделать программу, эксплуатирующую уязвимость, и почти мгновенно распространить ее среди всех желающих. Вследствие этого множество преступников, далеко не столь искусных, как первооткрыватель, могут воспользоваться инструментом взлома. Атаки стали распространяться гораздо быстрее, как инфекционные заболевания распространяются гораздо быстрее в эпоху реактивной авиации, чем в дни парусных судов.

Все эти факторы действуют совместно. В одной крупной финансовой компании мне говорили, что на их основной сайт совершается более 100 атак в секунду. Мне это количество показалось невероятным. Тем не менее Интернет экономически выгоден. Да, финансовой компании прихо-

дится отражать эти 100 атак каждую секунду и при том обеспечивать приемлемое время отклика для своих клиентов. Но эта компания тратит \$25, если клиент переводит деньги с одного счета на другой с помощью оператора, и всего лишь несколько центов, если это происходит в автоматическом режиме. Когда таких операций проводится за день несколько миллионов, экономия более чем окупает зарплату лучших специалистов по компьютерной безопасности в мире (они действительно работают в этой компании).

Нашим компьютерам приходится существовать в очень скверном мире. Наши требования к безопасности высоки, и программы часто не удовлетворяют им. Абсолютной безопасности достичь невозможно: как я покажу далее в этой главе, компромиссы и затраты окажутся чрезмерными. Но благодаря тщательному планированию и проектированию можно обеспечить приемлемый уровень защиты. Выражаясь фигурально, надо «надеть» на сетевые соединения то, что показано на рис. 3.1.



Рис. 3.1. Вот что в действительности нужно сделать с компьютером

Проблемы безопасности в современной компьютерной индустрии имеют два главных источника. Первый, как обычно, заключается в непонимании большинством программистов того, что касается безопасности. Это специализированная область, а они — универсалы. Мало кто из работающих в настоящее время в этой области изучал проблемы защиты в колледже, потому что этому их там не учили. Лишь сейчас эта дисциплина стала появляться в учебных расписаниях компьютерных курсов, зачастую

(но не всегда) преподаваемая людьми, не имеющими достаточного опыта. Почти все нынешние высшие руководители программной индустрии достигли своего положения в те дни, когда путь к продвижению вверх лежал через создание таких выдающихся функций, как двигающиеся брови скрепки для бумаг. Проблемы безопасности вызывают у них скуку и растерянность. Они мечтают, чтобы эти неряшливые ребята, занимающиеся безопасностью и ведущие себя странно даже по меркам гиков, как-нибудь справились с этим, чтобы сами они снова могли заняться приятными вещами.

Оборонительное мышление противоречит обычному образу мыслей программистов и не любимо ими. Это можно видеть по тому, как они сами описывают то, чем занимаются: «атакуют проблемы». Они расчлняют проблему, разбираются с ней и затем концентрируют свои силы на главном направлении, пока не достигнут успеха. Оборонительное мышление — совсем другое дело, его трудно принять и сохранять. «Оборонительное мышление» даже стало унижительным термином: «У тебя настроение Мажино, ты — безнадежный обыватель».¹ Попробуйте как-нибудь посмотреть живьем хоккейный матч, следя глазами только за защитниками своей команды. Во-первых, вы заметите, что практически невозможно делать это дольше нескольких секунд. Мозг постоянно направляет ваши глаза к движущейся шайбе и действиям вокруг нее. Но если заставить себя смотреть на защитников, то вы увидите, что они ведут совсем другую игру, нежели атакующие форварды. Они ждут, они наблюдают, они предвидят ход событий и заранее выбирают себе позицию, на которую, по их мнению, может переместиться действие. А во втором периоде попробуйте понаблюдать только за голкипером, последней линией обороны. Вам это не удастся, если только вы не мать вратаря или сам вратарь (или то и другое вместе). Нисколько. Поэтому вас не должно удивлять, что те, кто обучен вести атаку и занимался этим всю жизнь, с большим трудом может переключиться на что-либо иное.

¹ Линия Мажино — это ряд стратегических укреплений, построенных во Франции после Первой Мировой войны вдоль границы с Германией, чтобы отразить возможное вторжение. В 1940 году вместо того чтобы совершить самоубийственное нападение на эти укрепления, немецкая армия весьма не любезно обошла северный край линии и через незащищенную Бельгию заняла остальную часть Франции, а затем дождалась, пока защитники линии Мажино сдадутся, лишившись снабжения.

Вторая и более серьезная проблема состоит в том, что даже те программисты, которые разбираются в технологии безопасности, думают на языке этой технологии, а не на языке людей, которые ей пользуются. Пользователя они воспринимают как досадную помеху для своих замечательных уравнений, аналогично очень ученым врачам, воспринимающим пациента как неприятный побочный эффект некоей интересной болезни. Эти программисты считают, что если они правильно рассчитают свои внутренние стратегии, то все будет прекрасно. Ошибка! Безопасность, как и почти все в области вычислений, в большей мере является проблемой человеческого фактора, а не техники, что я покажу вам далее в этой главе. Гуру в области безопасности Брюс Шнейер (Bruce Schneier), о котором я подробнее расскажу ниже, утверждает, что безопасность — это процесс, а не результат. Это цепь, а для всякой цепи прочность определяется ее слабейшим звеном. И я добавлю к этому, что пока в технологии проектирования не будет понята и учтена человеческая природа пользователя, он останется слабейшим звеном, и здесь неизбежно будет рваться цепь. Позвольте привести вам несколько примеров.

Что должны были бы знать программисты

Как я повторил уже несколько раз, безопасность является сугубо специальной областью, о которой средний прикладной программист не имеет представления. Более того, он не сознает своего невежества и его последствий. Он говорит себе: «Ну, я займусь этой безопасностью, когда время придет, — как я это делаю всегда». Так не годится: преступники слишком умны и слишком увлечены своим черным делом. («У них нет подружек, — жаловался мне один знакомый программист. — Они не едят и не спят. Они занимаются только взломом. Разве мы можем победить в борьбе с ними?») Вот пример ошибочных действий некомпетентного программиста. Он взят из проекта, над которым я работал в молодые годы, лет 20 назад. Послушайте мою исповедь.

В то время я работал в компании, ныне усопшей и забытой, которая разрабатывала системы торговли иностранной валютой для крупных банков. Много нам продать не удалось, потому и «усопшая», но несколько систем было поставлено в банки, названия которых вы узнали бы даже сейчас, по прошествии 20 лет слияний и поглощений. Трейдеры начинали рабочий день, как все информационные работники в мире, — вводя с тер-

минала свой идентификатор пользователя и пароль. Эти данные посылались по сети на центральный сервер для проверки, после чего сервер разрешал осуществлять сделки.

Хитрость вот в чем. Существует устройство, называемое анализатором пакетов, с помощью которого каждый может прочесть и записать все данные, проходящие по сети, к которой оно подключено. Во многих случаях такие анализаторы — это просто программы, выполняемые на обычном ПК, которые настраивают и используют сетевое соединение особым образом, не свойственным обычным программам. У них есть многочисленные законные применения, включая аудит безопасности и разработку программ, поэтому запретить их использование нельзя. Кроме того, как наверняка заявили бы мои читатели из числа членов Национальной стрелковой ассоциации, если объявить преступным анализ пакетов, то только преступники и станут им заниматься.

Сеть была внутрибанковской и не имела никаких внешних соединений, поэтому мы не боялись, что ее данные сможет перехватить кто-то, находящийся на другом континенте. Однако любой банкир скажет вам, что наибольшую опасность представляют кражи, совершаемые собственными работниками и что банки устанавливают всевозможные виды защиты и аудита с целью их предотвращения и обнаружения.¹ Перехват данных в сети торговли валютой мог быть осуществлен с компьютера, подключенного к ней кабелем и находящегося на расстоянии какой-нибудь сотни метров. Невозможно найти подключившегося преступника среди всех этих проводов, протянутых через стены и потолки, особенно если пакетный анализатор запущен на компьютере, авторизованном для работы в сети, но не для выполнения этой программы.

Допустим, что мы посылаем пароль по сети открытым текстом, как показано на рис. 3.2. Злоумышленник с анализатором пакетов может про-

¹ Например, однажды я забрел в помещение, где велись торги, чтобы разобраться с забарахлившим компьютером, и застал там трейдеров, обсуждавших что-то конфиденциальное. От меня потребовали оставаться в комнате, пока они не закончат и не объявят результаты рынку, чтобы я не смог использовать услышанное мной в ущерб им. «Поверь нам, — сказали они, — мы не считаем тебя менее надежным, чем кто-либо другой, но бизнес есть бизнес. Ты же не хочешь попасть в число подозреваемых, если произойдет утечка информации. Посиди там, подальше от телефона, почитай газету и держи руки на виду, пока мы не закончим». Так я и поступил.



Рис. 3.2. Пересылка пароля открытым текстом; не очень удачная идея

честь пароль, украсть его, а затем зарегистрироваться в системе как авторизованный трейдер. Очевидно, что этого нельзя допустить. Мой тогдашний начальник был хорошим прикладным программистом и в целом достаточно умен, но в безопасности он не разбирался. Он предложил стратегию, показанную на рис. 3.3. Он решил зашифровать пароль перед отправкой (а) с помощью математической операции, называемой хешированием. Хеш-функция осуществляет перемешивание, и ее легко выполнить в прямом направлении, но невозможно в обратном (примерно как если смешать краски или разбить яйцо). Клиент должен был посылать по сети хешированный пароль, а не открытый текст. Злоумышленник, прочтя пакет, содержащий хеш, не сможет восстановить открытый текст в силу однонаправленности функции хеширования. Сервер же вычислит значение хеша из пароля, который он ожидал получить от этого пользователя (б). Если оно совпадает с присланным, значит, пользователь ввел правильный пароль и потому должен действительно быть тем трейдером, быть которым претендует. Если не совпадает, значит, пользователь выдает себя за другого.

К несчастью, проблема этим не решалась, поскольку значение хешированного пароля остается неизменным, пока не меняется текст пароля. Злоумышленнику достаточно перехватить хешированный пароль и написать



Рис. 3.3. Пересылка хешированного пароля; выглядит убедительно, но уязвима для атаки записи

программу, которая будет воспроизводить то же значение, отправляя его серверу, — тривиальная операция. Я действительно реализовал атаку запись/воспроизведение именно такого типа после того, как моя компания приказала долго жить, когда один клиент пожелал добавить некоторые функции в проект, поддержка которого прекратилась. Ранее я никогда не работал над такими программами, и я не самый гениальный программист, но я потратил всего минут 20, 10 из которых я соображал, куда воткнуть анализатор пакетов. Когда я увидел первые пакеты с данными регистрации, то не поверил, что все может быть так просто. Оказалось, может. Торговая система оказалась совершенно незащитной перед лицом не бог весть какой решительной атаки со стороны не бог весть какого сообразительного атакующего, получившего физический доступ к проводам сети, потому что проектировщик системы не обладал необходимыми знаниями.

Безопасный способ отправки паролей не так трудно спроектировать, если понять проблему, которая вам, уважаемый читатель, теперь ясна. Значение хеша должно случайным образом меняться при каждой регистрации, как показано на рис. 3.4. Клиент посылает серверу запрос на вход в систему, содержащий его имя, но не пароль (а). Сервер отвечает длинным случайным числом, называемым «вызов», которое меняется каждый

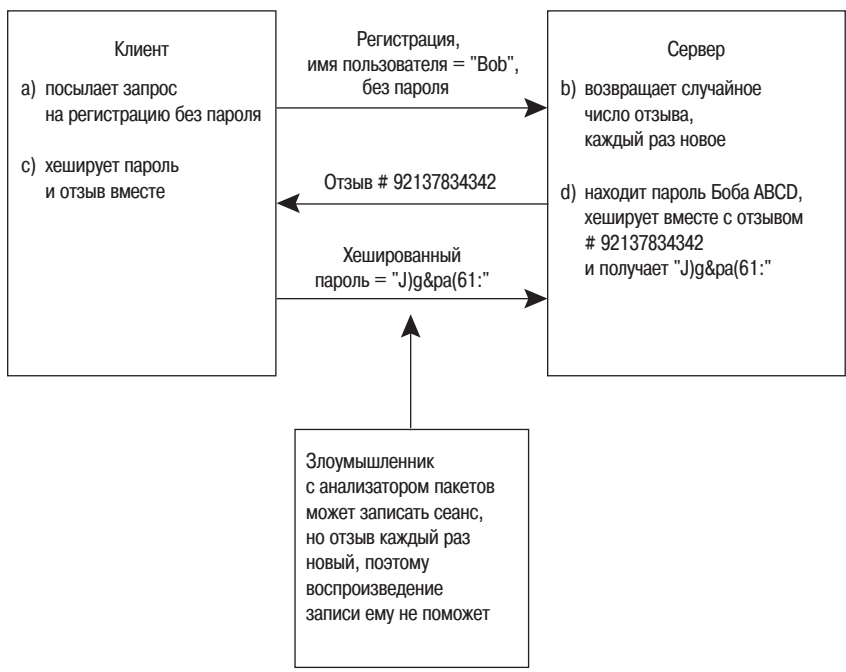


Рис. 3.4. *Протокол, применяемый при аутентификации. Число в отзыве каждый раз новое, поэтому значения хеша каждый раз разные. Атака записи не действует (что не означает безопасности протокола)*

раз (b). Клиент соединяет свой пароль с числом, содержащимся в вызове. Затем он производит хеширование объединенных данных и посылает результат (c), как в предыдущем примере. В это время сервер выполняет такое же объединение и вычисляет результат — так же как в предыдущем случае (d). Если числа совпадают, значит, клиент указал правильные пароль и число вызова, поэтому действительно должен быть тем, за кого себя выдает.

Перехват такого сеанса входа в систему не слишком поможет злоумышленнику. Он не может просто воспроизвести перехваченные данные, как в предыдущем примере, потому что при следующей попытке входа в систему число в отзыве будет другим, как и хеш, содержащий правильный пароль. Остается лишь атаковать систему методом грубой силы. Злоумышленнику нужно взять другой компьютер и перепробовать все открытые пароли, объединяя их с вызовом и хешируя, пока результат не

совпадет с тем, который он перехватил. (Не следует пытаться сделать это на работающей системе, поскольку практически любая схема регистрации, даже такая слабая, заблокирует учетную запись пользователя после нескольких неудачных попыток входа в систему, чтобы предотвратить такую атаку.) Тогда злоумышленнику станет ясно, что он определил открытый пароль пользователя, и он сможет вернуться в банковскую сеть и войти в систему с найденным паролем. Однако если пароль выбран достаточно длинным и случайным, атакующему потребуются тысячи лет, чтобы проверить все альтернативы, и к тому времени найденный им пароль вряд ли ему понадобится.

Такой тип входа в систему, называемый «квитированием» (*handshaking*), легко запрограммировать, если разобраться, что требуется сделать, и он не обременителен по времени при регистрации. Мой босс этого не понимал, как не понимал того, что не понимает этого. Ему не следовало проектировать системы безопасности, как и всем остальным, работавшим в этой компании, включая меня. К счастью, насколько мне известно, ни один из банков более чем за десять лет не воспользовался этой системой (отсюда эпитет «забытая»). Надеюсь, что я прав.

Каждый раз, когда вы читаете, что в некоей программе обнаружена новая брешь в защите, это означает, что один программист допустил ошибку подобного типа, а другой программист ее обнаружил. Обычно ошибки гораздо тоньше, чем та, которую я здесь продемонстрировал, и часто содержатся в иных частях программы, а не в схеме регистрации. Но все они того же типа и возникают по тем же причинам.

Человеческий фактор

Теперь вы немного больше разбираетесь в разработке защищенных программ, по крайней мере, в сравнении с тем малым, который зарабатывал этим себе на жизнь в не столь уж давние времена. (Вас это не пугает?) Но безопасность отнюдь не ограничивается технологией. Брюс Шнайер (Bruce Schneier) написал книгу о криптографии (искусстве составления и взлома кодов и шифров), озаглавленную «*Applied Cryptography*» (Wiley, 1995),¹ которая высоко оценивается специалистами в этой

¹ Брюс Шнайер «Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си». — Пер. с англ. — М.: Триумф, 2002.

области. Через семь лет после ее публикации в другой книге под названием «Secrets and Lies: Digital Security in a Networked World» (Тайны и ложь: цифровая безопасность в мире сетей) он поместил покаяние, начинающееся словами: «Я написал эту книгу, чтобы исправить свою ошибку. [«Applied Cryptography»] описывала математическую утопию. Я выдавал криптографию за Окончательный Ответ™».

«Результат не был удачным. Читатели решили, что криптография — это некий волшебный порошок, которым можно посыпать программу, чтобы сделать ее безопасной... Один мой коллега сказал, что в мире полно плохо защищенных систем, разработанных людьми, которые прочли «Applied Cryptography». Когда он занялся проектированием и анализом систем безопасности для своих клиентов, то «обнаружил, что их слабые места никак не связаны с математикой. Они обусловлены аппаратным и программным обеспечением, сетями и человеческим фактором. Прекрасные математические решения обесценивались в результате плохого программирования, скверных операционных систем или выбора неудачных паролей». Он делает следующий вывод: «Если вы считаете, что технология может решить ваши проблемы безопасности, значит, вы не разбираетесь ни в проблемах, ни в технологии». Полностью согласен».

В предыдущем примере я показал, как программисты делают ошибки, способные скомпрометировать их системы. Теперь я покажу вам, что даже если программист сделает все, что, по его мнению, от него зависит, пользователь все равно сделает систему уязвимой, поступив так, как обычно поступают люди.

В конце предыдущего раздела я рассказал, что злоумышленнику придется взламывать пароль с помощью грубой силы, пропуская все мыслимые пароли через функцию хеширования. Ему не потребуется много времени, чтобы установить, какая функция хеширования применяется в системе: таких, у которых есть необходимые математические свойства, не столь много, и их характерные признаки заметны специалисту. Не это обеспечивает защиту. Защита обеспечивается количеством допустимых паролей, которые придется проверить злоумышленнику. Если пароль состоит из шести букв от АААААА до ZZZZZZ, как это было в нашем случае, то придется проверить более 300 миллионов комбинаций. Если в пароль включить наряду с буквами и цифры, то это количество возрастет до 2 миллиардов. Если взять не шесть символов, а восемь, то количество комбинаций возрастет почти до 3 триллионов. Включите сюда буквы

нижнего регистра и символы пунктуации, и вы получите порядка квадриллиона (единица с пятнадцатью нулями) возможностей. Если проверять по 1000 комбинаций в секунду, полный перебор займет более 31 000 лет. Меняйте пароль 1–2 раза в год и можете забыть об опасности полного перебора.

Но злоумышленник хитер, и он учтет влияние человеческой природы при выборе пароля. Не все пароли встречаются с равными вероятностями. Обычно пароли выбираются людьми, а их голова думает словами. Как бы ни старался человек придумать случайный пароль, у него это не получится. Поэтому сообразительный злоумышленник воспользуется не грубой силой, а «атакой по словарю». Он не станет пробовать сначала АААААА, потом АААААВ, потом АААААС и т. д. Он решит, что выбранный пользователем пароль может оказаться настоящим словом, а потому должен быть в каком-нибудь словаре. Он возьмет список часто употребляемых слов (в Сети таких очень много) и перепробует их одно за другим: ААRON, АВACUS, АСТIVE и т. д. В среднем рабочий словарь человека насчитывает около 5000 слов, если не считать моих собственных изобретений вроде `idoit` (глава 1) и `marketingbozo` (глава 6). Можете сами посчитать, сколько потребуется времени при переборе с той же скоростью 1000 комбинаций в секунду. Неудачный выбор пароля сокращает время взлома с 31 000 лет до 5 секунд.¹ Даже если пароль образован из двух слов, состыкованных вместе (как первый пароль, назначенный мне `CompuServe`), его вскрытие все равно займет менее семи часов.

Совершенно необходимо, чтобы пароли были полностью случайными, типа `k!&wLa` или `b8I2=o`. Генерация таких паролей не составляет труда — есть множество утилит и сайтов, которые умеют это делать. Системе для продажи иностранной валюты, действующей в банке, который ежедневно переводит миллиарды долларов с одних счетов на другие, следовало бы назначать пользователям случайные пароли. При этом она должна

¹ Иногда пользователям кажется, что они защитят свой пароль, составив его из слов иностранного языка. При скорости пять секунд на словарь из 5000 слов злоумышленник может проверить все 404 языка из списка International Standards Organization, от абхазского до зунийского, в течение получаса. На практике он проверит десяток-другой самых распространенных. Если, что маловероятно, это не поможет, он возьмется за взлом другого трейдера.

была бы позволять им в любой момент заменить свой пароль другим случайным же, но не произвольно выбранным им самим.

Очевидно, у этого метода есть недостатки. Случайный пароль очень тяжело запомнить, особенно если им пользуются не часто или приходится помнить несколько паролей. Поскольку работа с этим приложением — основная задача трейдера и он регистрируется в системе, по крайней мере, дважды в сутки (в начале рабочего дня и после обеда), он при сильном желании и достаточно редкой смене пароля — скажем, дважды в год — мог бы его запомнить. Пришлось бы побороться с естественным желанием трейдера записать пароль на бумажке, пригрозив ему увольнением с этой желанной и высокооплачиваемой работы. Потом пришлось бы для острастки иногда обшаривать рабочие столы трейдеров и выгонять пойманных с поличным. Поскольку трейдеры уже смирились со многими другими ограничениями, обусловленными обеспечением конфиденциальности информации, возможно, они проглотили бы и это тоже.

Такой высокий уровень защиты паролей было бы гораздо труднее обеспечить на большинстве сайтов, обслуживающих клиентов, таких как сетевые журналы для подписчиков или даже магазины розничной торговли, да это и не было бы так уж целесообразно. Обычный покупатель не станет терпеть даже доли тех неудобств, на которые согласен высокооплачиваемый служащий, а ущерб от возможного взлома гораздо меньше. Обычные пользователи не в состоянии запомнить много случайных паролей — и один-то не все помнят. Если пытаться принуждать их, они станут записывать пароли на стикеры и приклеивать к мониторам либо перейдут к другому продавцу, у которого сайт устроен попроще. К счастью, какая-то единственная учетная запись на коммерческом сайте интересует злоумышленника гораздо меньше, чем система продажи иностранной валюты крупного банка, и компрометация защиты приводит к гораздо меньшему ущербу.

Бюджет неудобств

Полностью безопасен только такой компьютер, который не соединен ни с какими сетями, отключен от питания, замурован в бетон и закопан на три метра под землю. Но такие компьютеры не слишком пригодны для работы, поэтому всякое реальное использование компьютеров предполагает некий компромисс между полезностью и безопасностью. Пра-

вильный выбор соотношения между одним и другим при разработке приложения определяет успех или неудачу проекта. Если правила слишком мягкие, они не дают достаточной защиты, а причиняемые неудобства напрасны. Если же правила слишком строги, то пользователи либо откажутся от продукта, либо найдут способ обойти правила.

Для описания этой проблемы я сочинил термин *бюджет неудобств* (*hassle budget*). Под ним я понимаю объем неудобств, связанных с требованиями безопасности, который пользователь готов вытерпеть при работе с продуктом. Он может быть весьма различен от продукта к продукту и от пользователя к пользователю. Разработчики программ должны руководствоваться этим бюджетом в необходимом для решения задачи размере, но никогда не превышать его.

Простой пример дают часы в правом нижнем углу панели задач Windows. При двойном щелчке по часам появляется окно с часами и календарем, приведенное на рис. 3.5, которое показывает текущие дату и время и позволяет изменить их.

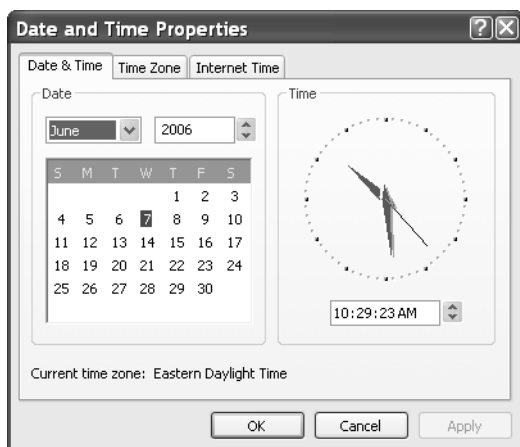


Рис. 3.5. Часы-календарь Windows XP

Этой функцией очень удобно пользоваться при обсуждении по телефону дат будущих встреч с клиентами. Мне надо быстро узнать день недели 23 ноября или дату через неделю после понедельника, и с помощью этого окна я получаю немедленный ответ, представленный в графической форме. Проблема, однако, в том, что только администратор системы может



Рис. 3.6. Окно сообщения, запрещающее пользователю менять время

увидеть это окно (рис. 3.5). Например, моя жена не является администратором нашей домашней машины (Бог миловал). Поэтому, если я хочу узнать дату, когда я сижу за ее компьютером, и я попытаюсь вызвать это окно двойным щелчком, появится раздражающее сообщение (не стоило называть его «сообщением об ошибке» — это не я сделал ошибку, а программист, его написавший), которое извещает о недостаточности моих прав для изменения системного времени (рис. 3.6). Поэтому чтобы увидеть календарь, я должен вернуться в начальное меню и выбрать Пуск → Программы → Стандартные → Идиотизм → Неудобства → Тупицы... (смысл ясен).

Может быть, это и правильно, чтобы только администратор имел право устанавливать системные часы. Правильное функционирование многих программ зависит от корректности внутренних даты и времени компьютера, и к их изменению и правда можно допускать только тех, кто понимает, что делает. Но разве разумно запрещать обычному пользователю даже увидеть часы с календарем? Это не настолько секретные данные: какой смысл запрещать пользователю узнать, что за число будет в следующий четверг? Обычному пользователю нужно разрешить видеть окно, но как-то предупредить его, что оно доступно только для чтения — может быть, затенить кнопку ОК, или вывести текстовую строку, или показать другое окно, предназначенное только для отображения. Разработчик неразумно потратил бюджет неудобств пользователя.

Когда у пользователя возникает ощущение, что его бюджет неудобств превышен, он ищет пути обхода правил, установленных программистом, не всегда отличая разумные от неразумных. Он не знает, как это сделать, и не очень-то хочет знать — это вообще задача разработчика приложения. В данном примере часов-календаря пользователи станут назначать себя администраторами, чтобы видеть это окно. С точки зрения пользователя это разумно, хотя на практике нарушает правило защиты, именуемое «принципом наименьших прав». Оно утверждает, что у пользователя дол-

жен быть наинизший уровень привилегий, позволяющий ему выполнить свою работу, благодаря чему при захвате его идентификационных данных злоумышленником потенциальный ущерб окажется минимальным. Если злоумышленник установит контроль над системой как пользователь с правами администратора, он сможет уничтожить любой файл на диске, тогда как если у него будут права пользователя с более низкими привилегиями, он, вероятно, сможет удалить только те файлы, которые принадлежат данному пользователю. Особенно это важно в домашних условиях, когда несколько членов семьи работают на одном компьютере. Обычному пользователю не обязательно знать этот принцип, и соображения удобства не должны стимулировать его нарушение.

Писавший этот код программист, несомненно, подумал: «Пользователю не следует этого делать: это диалоговое окно не для произвольного просмотра календаря». Эту исходную посылку и необходимо изменить. Это программист должен подстраиваться под пользователя, а не наоборот. Пользователь считал данную функцию полезной и удобной. Задача программиста — сделать ее безопасной.

Вот еще один пример, более жестокий. Много лет назад я работал в компании, которая делала ложные макеты кораблей и судов для обнаружения их радарными. Мы брали связки алитированного стекловолокна метровой длины и нарезали до половины длины волны радара, который надлежало обмануть (обычно несколько сантиметров) с помощью электрифицированного пресса с лезвием-гильотиной. У этого устройства было два предохранителя, расположенных так далеко один от другого, чтобы оператор не мог одновременно достать до них одной рукой, и оно не работало, если не нажать одновременно обе кнопки. Проектировщик думал, что таким образом он помешает оператору подсунуть руки под работающее лезвие. Однако был один оператор, любивший курить во время работы машины. (В те времена работникам разрешалось курить на рабочем месте.) Он сделал себе палочку с двумя выступами на нужном расстоянии. Положив ее так, чтобы выступы приходились против кнопок, нажимал на нее одной рукой и курил, действуя другой. Он вполне преуспел в искусстве одной рукой доставать, поджигать и подносить ко рту гвоздь для крышки своего гроба. Он почувствовал, что его бюджет неудобств превышен, и придумал выход. Надеюсь, он нашел протез руки с держателем сигарет.

Пользователь ленив

Если существует одно качество, которым отличаются человеческие существа, особенно из числа тех, кто пользуется компьютерами, то это лень. Пожалуй, я даже склонен утверждать, что лень — это третий по важности фактор, определяющий человеческое поведение, после голода и похоти.¹ Как я отмечал в главе 1, большинство программистов ездит на машинах с ручной коробкой передач, в отличие от подавляющего большинства их пользователей. Пользователям нужно только решить конкретную задачу: прочесть новости, чтобы не покупать газет, заказать продукты, чтобы не выходить в магазин, послушать или украсть музыку и фильмы, чтобы не покупать их, загрузить похабные картинки, потому что им не с кем провести выходные. Компьютер как таковой не вызывает у них ни малейшего интереса, так же как у них ни малейшего интереса не вызывает автомобиль сам по себе. Им еще больше понравилось бы, если бы достаточно было щелкнуть пальцами, чтобы решить их проблемы, не прибегая ни к первому, ни ко второму. Как они не меняют масло в двигателе и не проверяют давление в шинах, что должны были бы делать, и не проходили бы никакого технического осмотра, если бы не существовало законов на этот счет, так они не станут делать ничего для защиты своих компьютеров, если не будут к тому принуждены. При правильном проектировании программ эту особенность принимают во внимание и автоматически защищают все, что только можно. Понимание и укрощение человеческой лени успешно применяются в книжных клубах и, надо полагать, могут быть поставлены на службу разработчикам программного обеспечения. Приведем пример, показывающий, как не надо это делать, а потом внесем коррективы.

В прежние времена компании-производители ПО выпускали новые версии своих приложений примерно раз в год. В каждой версии появлялись новые функции и исправлялись ошибки, появившиеся благодаря функциям, добавленным в предыдущих версиях (и исправлялись ошибки, появившиеся благодаря исправлениям ошибок в предыдущих версиях,

¹ Сначала я поместил лень на второе место, вслед за голодом. Затем просматривавший рукопись коллега спросил у меня: «Ты бы постриг днем газон, если бы за это тебе был обещан секс вечером?» Когда в следующий раз какой-нибудь самодовольный сноб от математики заявит вам, что нельзя сравнивать яблоки с апельсинами, задайте ему этот же вопрос.

и т. д.). С тех пор как мы стали все чаще работать в онлайн и подвергаться атакам через Интернет, такая скорость обновлений перестала удовлетворять. Как только обнаруживается брешь в защите — предпочтительно, добропорядочным пользователем, — изготовитель ПО должен как можно быстрее залатать ее, — прежде чем ею воспользуются злоумышленники. Как правило, это делается с помощью *заплат (патчей)* — небольших программ, которые надо загрузить с веб-сайта производителя и установить на своем компьютере. Обычно это не новая версия программы целиком, а небольшой заменяющий фрагмент, в котором устранена уязвимость.

Проблема в том, что патчи, подобно ремням безопасности и средствам контроля рождаемости, действуют только тогда, когда их применяют, чего почти никто не делает по собственному почину. Поставщики размещают патчи на своих сайтах, иногда информируя о них зарегистрированных пользователей с помощью электронной почты. Не только значительная доля пользователей, сменивших свой почтовый адрес или никогда не регистрировавших свой продукт, но и большинство получателей уведомлений не считают нужным зайти на сайт, найти необходимый патч среди прочего хлама (особенно трудно найти что-нибудь у Hewlett-Packard, хотя, конечно, им приходится поддерживать чрезвычайно много продуктов), загрузить его и установить, рискуя при этом испортить что-то другое.

Возложение на пользователей обязанности следить за своевременной установкой патчей может привести к тяжелым последствиям. Например, червь Slammer, поразивший серверы баз данных 25 января 2003 года, выведший многие из них из строя и едва не остановивший Интернет полностью, не оказал бы такого разрушительного действия при правильном управлении патчами. Microsoft выпустила необходимый патч за 3 месяца до события — качественный патч, устранявший уязвимость и не имевший побочных эффектов. На пораженных компьютерах, в том числе нескольких, принадлежащих самой Microsoft, патч не был установлен, несмотря на то, что у подвергшейся атаке программы обычно есть специальный администратор, в число обязанностей которого входит слежение за подобными вещами. Если даже профессионалы, работающие в Microsoft, не следят за выходом собственных патчей, то какие шансы остаются у всех прочих? Никаких. Ноль. Ничто.

После этого Microsoft прибегла к тому, чем давно уже занимались производители антивирусных программ, — автоматической установке обновлений. Сервис обновления Windows сначала вышел сам в виде обновле-

ния, вследствие чего мало кто потрудился найти его, установить и эксплуатировать. Однако теперь он встроен в Windows XP Service Pack 2, и чтобы отключить его, требуются особые усилия. Он периодически проверяет сайт Microsoft и, обнаружив обновления, загружает их и устанавливает. Со следующей версией инструментальных средств Microsoft для разработчиков у прикладных программистов появится недорогая и необременительная возможность включать в свои программы аналогичные средства обновления. Основная проблема связана с тем, что если при автоматической установке патчей нарушается функциональность программы, компанию захлестывает поток дорогостоящих телефонных обращений в службу поддержки. Такая перспектива должна стимулировать тщательное тестирование обновлений.

Когда пользователь решает, какие элементы Windows должны быть включены по умолчанию, а какие потребуют от него специальных действий, он также руководствуется в основном ленью. Рассчитывать, что пользователи предпримут активные действия для обеспечения защиты своих машин, — это ошибка. Например, очень многие программы поставлялись с готовой учетной записью администратора с паролем «admin», который, как предполагалось, будет заменен. Количество администраторов, которые так и не удосужились это сделать, достойно удивления. И я был таким, пока меня не поразил один веб-вирус, который отгадал мой пароль. Ричард Фейнман, Нобелевский лауреат по физике, писал в своей автобиографии о лени как о третьем главном обстоятельстве, позволявшем ему вскрывать сейфы, чем он занимался ради забавы. (Первое место принадлежало беспечности хозяина, который записывал код и оставлял его у себя на столе, второе — его наивности, под влиянием которой он составлял код из важных дат своей жизни.) Большинство сейфов поставляется с готовыми комбинациями типа 0 — 25 — 0 или 25 — 50 — 25. Предполагается, что покупатели изменят их, но они часто этого не делают. Гораздо безопаснее поставлять программу с отключением большинства функций, чтобы пользователи включали только то, что им требуется, а не включать все в надежде, что пользователи отключат ненужное. Здесь тот же недостаток: пользователи станут звонить в службу поддержки, не обнаружив поначалу той функции, которая, как им известно, явно должна иметься в программе. Отвечать им каждый раз по телефону, что, дескать, «надо щелкнуть там-то, ввести пароль, и все заработает» — достаточно накладно. С технической точки зрения недостаточно по умолчанию отключить

какую-то функцию. Производитель ПО должен хорошо продумать, как помочь пользователям активировать нужные им части программы — безопасно и просто.

Работая с бюджетами неудобств и ленивыми пользователями, я сформулировал Третий фундаментальный закон Платта, гласящий, что «лень побеждает все». Если что-то легко сделать, это — надо или не надо — будут делать часто, а если что-то сделать трудно, то это будут делать редко — надо или не надо. Поэтому умный разработчик программы постарается, чтобы хорошие, правильные, безопасные вещи было делать легко, а плохие, глупые и опасные — трудно. Создателю диалогового окна «часы/календарь» полезно было бы это учесть.

Психологическая атака

Профессор физики моего колледжа однажды поставил классу такую задачу: придумать возможно больше способов измерить высоту здания с помощью барометра. Помимо очевидного измерения разницы давлений воздуха вверху и внизу наша мозговая атака, подкрепленная некими химическими стимуляторами, произвела на свет и другие способы: скинуть барометр с крыши и замерить время его падения, измерить изменение его веса при удалении от центра массы Земли и т. д. Но самый простой и точный способ из всех нами придуманных был следующим: пойти к владельцу здания и предложить ему назвать его высоту в обмен на исправный новый барометр.

Часто этот способ взлома системы оказывается простейшим — просто мило поговорить с пользователем. В то же время это самая большая опасность, с которой сталкиваются производители программного обеспечения, потому что все их ухищрения оказываются напрасными. Искусство взлома системы защиты путем деликатных разговоров называется психологической атакой, или *социальной инженерией* (*social engineering*). Просто удивительно, как часто этот способ приводит к успеху.

Суперхакер Кевин Митник (Kevin Mitnick), приговоренный к тюремному заключению за взлом компьютерных систем, свидетельствовал перед Конгрессом США в 2000 году: «Психологические методы были настолько успешны, что мне редко приходилось прибегать к техническим... Можно впустую потратить миллионы долларов на технические средства защиты, потому что можно просто позвонить кому-нибудь по телефону

и уговорить сделать на компьютере что-то такое, благодаря чему ослабится его защита или вскрыется информация, которую мы разыскиваем». В своей книге «The Art of Deception: Controlling the Human Element of Security» (Искусство обмана), Wiley, 2002 он описывает ряд своих излюбленных приемов.

Допустим, к примеру, что у взломщика в руках оказались телефонный справочник и организационная структура вашей компании — скажем, в результате просеивания ваших мусорных баков (это занятие называется «dumpster diving»). После этого вам звонит некая женщина (женщинам всегда верят больше, чем мужчинам) и говорит: «Привет, Боб, это Сара (замените на имя вашего администратора, взятое из организационной схемы). Я работаю у мистера Франклина (начальник начальника вашего начальника, опять взято из схемы). Кажется, нас знакомили на последней новогодней вечеринке. (Не было этого, но разве вспомнишь, да и кто станет возражать, особенно если у звонящего такой приятный голос?) Мы в офисе того нового клиента (в результате предшествующего звонка выяснилось, что этой шишки нет на месте), и нам нужна твоя помощь. Мне отсюда никак не войти в систему компании, а нам очень нужен файл по (крупный и важный проект). Ты не мог бы послать его электронной почтой на — (голос в сторону) Мистер Франклин, не повторите этот адрес? (снова в трубку) — *thisguy@seemsplausible.com*. Спасибо, ты нам очень помог. Обязательно сообщу мистеру Франклину. Пока».

Ну кто сможет устоять? Очевидно, звонившая знает компанию и чем она занимается. Она, очевидно, допущена к делам. Можно пометчать о грядущем повышении и результатах, последовавших бы за отказом сотрудничать. Это и есть психологическая атака. Скажете, что это не имеет никакого отношения к компьютерной безопасности? Но весь ваш тяжкий труд по защите своей информации оказался бессмысленным из-за этого милого короткого разговора. Вот оно — самое слабое звено в цепи, и пока вы его не укрепите, вся ваша замечательная технология бесполезна.

Вы потрясены? Напрасно. Не стоит недооценивать пределы человеческой глупости. Какими-то аналогичными методами можно украсть ID пользователя и его пароль. «Послушайте, Платт! — вскричите вы. — Я никогда этого не сделаю. Что я — идиот?» Но нет ничего сверхъестественного в том, чтобы передать кому-то свой ID пользователя и пароль. Вы же делаете это при каждой регистрации. Все, что требуется от злоумыш-

ленника, это создать такую ситуацию, чтобы вы думали, будто это законный запрос ваших данных авторизации, хотя на самом деле это не так.

На рис. 3.7 приведено сообщение, которое несколько лет назад часто можно было получить по электронной почте. Оно якобы исходит от PayPal.com, службы переводов электронных денег, но в действительности отправлено жуликом. Как видите, в поле From: указан адрес PayPal.com. Большинство пользователей не сознает, что этот адрес не указывает на реального отправителя письма. В поле From: можно поместить все, что угодно, так же как написать любой адрес отправителя на конверте с обычным письмом. Обычно вы пишете там верный адрес, чтобы заблудившиеся письма возвращались обратно, но здесь иной случай. Действительный путь к отправителю, так сказать почтовый штемпель, таится в почтовых заголовках, которые эта программа не показывает. Формат письма соответствует обычному внешнему виду PayPal, поскольку злоумышленник просто скопировал его с сайта PayPal.com. Все ссылки, за исключением «click here», указывают на настоящие страницы PayPal.com. Если же вы щелкнете по этой ссылке, то попадете на сайт с именем, похожим на PayPal (ebay-paypal.com, если мне не изменяет память), но принадлежащим

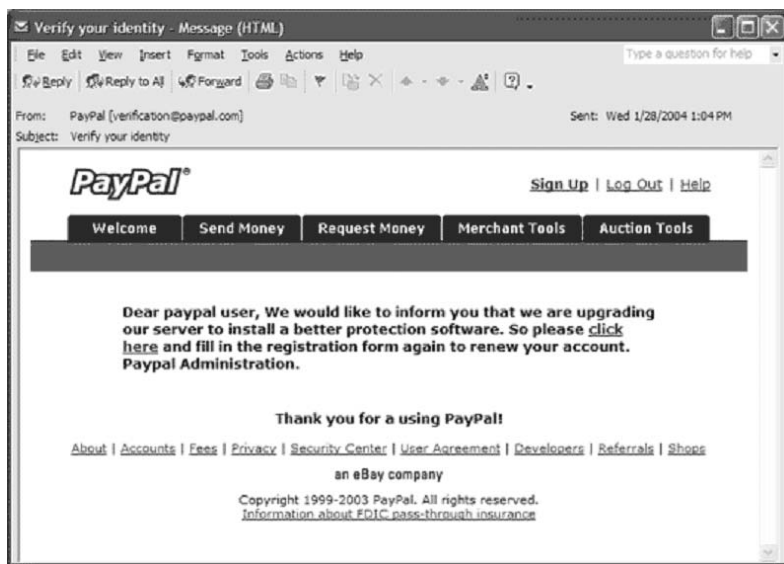


Рис. 3.7. Атака типа «фишинг»

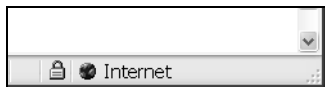


Рис. 3.8. Замок в панели статуса браузера, указывающий на защищенное соединение



Рис. 3.9. Замок на самой странице вместо панели статуса браузера, имитирующий защищенное соединение

злоумышленнику. Единственное явное отличие состоит в том, что замочек в правом нижнем углу, который должен обозначать защищенный сайт, находится на самой странице (рис. 3.8), а не в панели статуса браузера (рис. 3.9). Казалось бы, это очевидная ошибка, но в оживленной среде обычной веб-страницы (рис. 3.10) ее трудно заметить. За исключением грамматической ошибки («Thank you for a using PayPal!»), страница выглядит, как настоящая. Подобные психологические атаки объединены названием **фишинг** (phishing) — кража идентификационных данных.

Изготовитель программного обеспечения почти беззащитен перед такими приемами. Пользователи должны знать, что нельзя доверять ссылкам в случайных письмах. Но и тогда они могут выполнить инструкцию, предлагающую ввести строку «ebay-paypal.com» в браузере и попасть на сайт злоумышленника. Даже могущественный Citibank (подлинный адрес *citibank.com*) уязвим для веб-адресов, которые на первый взгляд кажутся похожими, таких как *cittibank.biz*, существовавший по состоянию на 1 марта 2006 года.

Чтобы украсть пароль, злоумышленнику может и не потребоваться маскарад. Можно потратить сотню долларов на покупку какого-нибудь контента — музыки, видео, порно и т. п. — или просто украсть его. Затем организуется веб-сайт, который кидает что-нибудь пользователям на закуску, а потом извещает их, что для доступа к остальному контенту необходимо открыть бесплатный счет. При этом пользователю дается (лживое) обещание, что его личные данные нигде не будут использованы, а для придания пушпей убедительности может даже добавляться контрольный

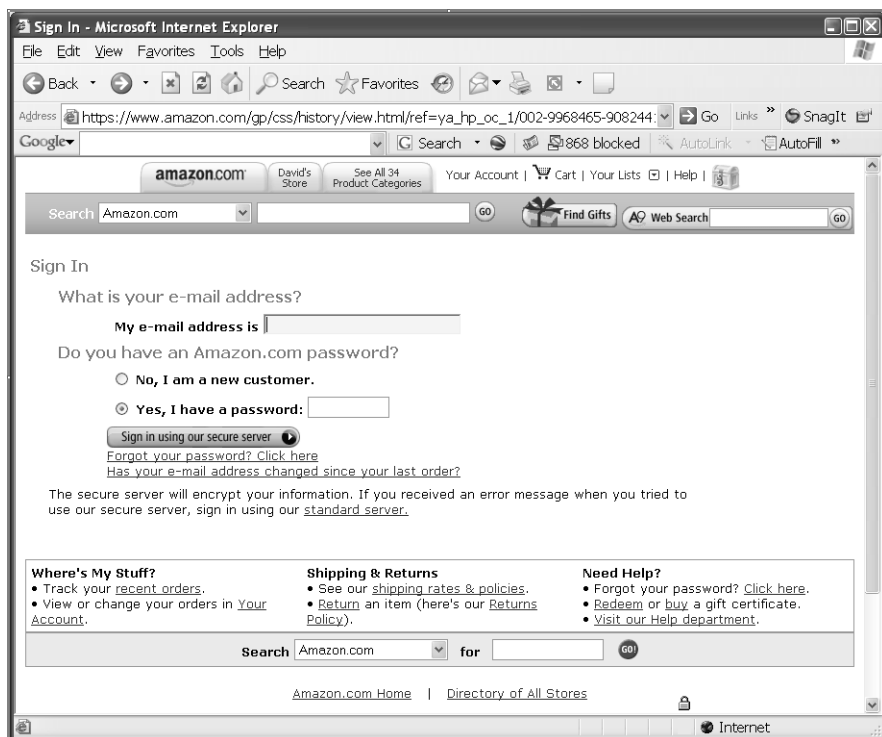


Рис. 3.10. *Замок на самой странице, который трудно различить на странице, где много материала*

вопрос на случай потери пароля. Выбранные пользователем ID и пароль проверяются злоумышленником на других сайтах, скажем Amazon или eBay. Или на сайтах инвестиционных фондов. Если пользователь указал те же ID и пароль, которые установлены им на других сайтах, а это происходит очень часто, то вся его защищенность утрачена.

Поскольку так легко украсть пароль, вы поймете, почему я столь неодобрительно отношусь к биометрической аутентификации (о ней рассказывается в главе 4), хотя устройства чтения отпечатков пальцев начинают встраивать в клавиатуры обычных пользователей. Если ваш отпечаток пальца будет украден, вы не сможете заменить его новым. Злоумышленнику даже не нужно добывать стакан, из которого вы пили, или проявлять отпечатки на вашей дверной ручке. Ему достаточно попросить вас иден-

тифицировать себя с помощью устройства чтения отпечатков и послать ему файл с результатом сканирования. С психологическими атаками крайне трудно бороться с помощью технологии. Купите книгу Митника, и после ее прочтения несколько бессонных ночей вам гарантированы. Последней книгой, оказавшей на меня такое воздействие, была «The Coming Plague: Newly Emerging Diseases in a World Out of Balance» (Грядущая чума: новые болезни в нестабильном мире), Laurie Garrett (Лори Гаррет), Penguin, 1995. Похоже, что взлом защиты компьютеров и геморрагическая лихорадка имеют между собой больше общего, чем мне представлялось ранее.

Заключительное слово о безопасности

Хуже всего, когда возникает ложное чувство безопасности. Расскажу вам об одном моем клиенте, имени которого не назову по причинам, которые вскоре станут понятными. В первый день моей работы у этого клиента я вошел в здание вместе с группой служащих, махнув охране карточкой на десять чашек кофе в Dunkin' Donuts, как если бы это был такой же пропуск, как у всех остальных. На следующий день я решил вести себя прилично и зарегистрироваться на входе. Наемная охранница-полицейский (между прочим, с нашивками капитана) не попросила у меня никаких документов. Она не стала звонить тому человеку, к которому я якобы шел, чтобы узнать, ждет ли он меня. Она не обратила внимания на имя, под которым я зарегистрировался в книге, — Тед Качински. Она помахала мне рукой и пожелала удачи. Моему клиенту лучше было бы не платить за такую «безопасность», а сжечь эти деньги на глазах у всех. Менее защищенным он в результате не стал бы, но ему хотя бы перестало казаться, что он защищен.

В то утро я минут 15 разглагольствовал перед собравшимися о полном отсутствии безопасности в их компании и советовал им быть осторожнее, потому что те, кому они платят деньги за охрану, не справляются со своим делом. В тот момент, когда я закончил свою речь, один из слушателей сказал: «Послушайте, возможно, это розыгрыш, но я только что получил e-mail с сообщением, что какой-то самолет врезался во Всемирный торговый центр». — «Там не стоит дата 1 апреля?» — спросил я, вспомнив о закрьвшемся теперь британском юмористическом журнале «Punch». Согласно легенде на обложке номера 1 апреля 1919 года было напечатано:

«Эрцгерцог Фердинанд оказался жив, война началась по ошибке».¹ Но это было 11 сентября. Теперь мы все знаем эту дату. Вот к чему приводит ложное чувство безопасности («Ножи для разрезания картона? Пожалуйста. Запрещены только опасные вещи, например горячий кофе с молоком из Starbucks»). Так что постарайтесь, чтобы у вас его — ложного чувства безопасности — не было. А кофе с молоком — пожалуйста, если вам нравится, хотя я никогда не видел в нем смысла, как в кофе без кофеина и пиве без алкоголя.

Что мы можем сделать

Защита имеет сегодня большое значение. Пока работа ограничивалась Солитером и Блокнотом, все было не так, но времена изменились. Если вы тоскуете по тем беззаботным денькам, то я разделяю ваши чувства, но таков неизбежный ход событий.

Кое-что о компьютерной безопасности вы сейчас узнали. Слишком мало, чтобы проектировать системы, но достаточно, чтобы знать, когда проявлять беспокойство (часто), и чтобы распознать мошенничество, если столкнетесь с ним. Если вам кажется, что нечто должно вызывать у вас беспокойство, то так оно, видимо, и есть.

Основные производители программного обеспечения, а не только разработчики систем защиты, начинают уделять внимание проблемам безопасности. В частности, на них, по-видимому, обратила внимание Microsoft. Билл Гейтс 15 января 2002 года разослал внутри компании циркуляр, назначавший высший приоритет тому, что он назвал «доверительными вычислениями» (trustworthy computing). «...доверительные вычисления — самая важная область нашей работы. Если мы не решим эту задачу, люди просто не захотят — или не смогут — воспользоваться всеми остальными нашими замечательными достижениями». Я годами повторял, что Microsoft должна сменить приоритеты, что тратить целый человеко-день — да что там, человеко-минуту — на создание моргающей скрепки, когда не

¹ Причины Первой мировой войны (1914—1918) многочисленны и сложны. Но спичкой, от которой взорвалась пороховая бочка, стало убийство эрцгерцога Франца-Фердинанда, наследника австрийского трона, совершенное 28 июня 1914 года в Сараево. И хотя журнал «Punch» известен в основном как юмористический, он первым 8 декабря 1915 года опубликовал отнюдь не юмористическое стихотворение Джона Макрэя «На полях Фландрии».

сделано все возможное ради устранения уязвимостей в защите и предотвращения аварийных сбоев, это означает поклоняться ложным ценностям.

В прессе сообщалось, что все разработчики компании прошли обучение по проблемам безопасности и затем строка за строкой проанализировали весь код с точки зрения его уязвимости. Кроме того, согласно имеющимся у меня сведениям, произошел существенный сдвиг в культуре компании: безопасность стала обязательным элементом каждой проектной спецификации, составляемой всеми разработчиками, а не дополнительным качеством, над которым работают, если остается время (как будто где-нибудь когда-нибудь «остается» время: время для важной работы должно быть выделено, иначе она никогда не будет сделана); продвижение по службе получают те менеджеры, которые создают более защищенные, а не более красивые продукты. Как я уже писал, среди программистов уважение со стороны коллег ценится выше, чем в других известных мне группах профессионалов, включая летчиков и врачей. И это уважение, — когда тот, чей ум вы признаете, смотрит на вашу работу и говорит, что она сделана толково, — теперь переходит от тех, кто придумывает анимационные интерфейсы, к людям, которые умеют находить и исправлять ошибки защиты. Но Microsoft — большая компания с огромным объемом кода, который некому сопровождать. При самом большом старании потребуетсЯ некоторое время, чтобы начали сказываться результаты этих усилий.

Несомненно, компаниям, выпускающим программные продукты, необходимо многое сделать для обеспечения более высокой защиты, в частности наладить автоматическую рассылку исправлений. А пользователи должны требовать от них этого, так же как они требуют улучшений в интерфейсе. Но, в отличие от уродливости интерфейса, о которой я уже писал, безопасность зависит не только от создателя программ. Она требует участия пользователя. Например, выпускаемые Microsoft новые продукты Windows XP Service Pack 2 и Windows Server 2003 менее уязвимы, чем ее предыдущие уроды Windows 95 и Windows 98. И злоумышленник постарается атаковать именно эти последние, а пользователь, который работает с ними, фактически заявляет: «Меня не волнует безопасность; я знаю, что есть значительно лучше защищенные системы, но я решил не пользоваться ими — делайте со мной, что хотите».

В классической книге Дейла Карнеги «Как завоевывать друзей и оказывать влияние на людей» (впервые опубликованной в 1937 году) правило номер 7 для тех, кто хочет сделать свою семейную жизнь счастливой,

гласит: «Прочтите хорошую книгу о сексуальной стороне брака». Здесь есть сходство с безопасностью, особенно если представить себе, что эти нехорошие ребята хотели бы сделать с вашим компьютером и с вами самим. В мире компьютеров слишком быстро происходят изменения, чтобы книги могли оказать большую пользу (исключая данную, разумеется). Но нужно раз в год читать хорошую статью по безопасности для пользователей и выполнять все ее рекомендации. Обновите операционную систему, установив самую надежную. Установите брандмауэр и антивирусное ПО. Активируйте автоматическое обновление, даже если в результате иногда будут зависать программы. Потому что, хоть я и ругал софтверные компании за то, что они пишут дрянные программы, здесь вы должны сами пойти им навстречу.

ГЛАВА ЧЕТВЕРТАЯ

Ты кто такой?

Майк, валютный трейдер в банке, который я когда-то консультировал, никак не мог понять, зачем нужны пароли. Верный выпускник Университета штата Висконсин, Майк упорно задавал себе пароль «Badger» — «барсук», талисман штата. После того как я сбрасывал его пароль, он вещал всем присутствовавшим: «Слушайте, ребята, Платт снова сделал меня барсуком!»

С чего это началось

Представляю, как вы ухмыляетесь, слушая эту историю, и хлопаете себя по лбу, как Гомер Симпсон, произносящий свое «Доу!» Конечно, вам тут все известно. Пароли служат для аутентификации — чтобы убедить компьютер в том, что вы тот, за кого себя выдаете. Рабочий день большинства пользователей начинается с ввода ID и пароля. Та же процедура повторяется каждый раз, когда вы проводите какие-то операции на разных веб-сайтах, даже когда просто проверяете почту. Пароли распространены повсеместно, пользоваться ими легко, и все связанное с ними должно быть хорошо понятно. Но способ, которым они применяются в настоящее время, — отдельные ID и пароль пользователя для каждой системы, с которой вы взаимодействуете, — уже не годится в современной вычислительной среде.

Почему с отстоем до сих пор не покончено

О тчасти проблема обусловлена непонятливостью пользователей. Обычно с этим можно справиться, и, что главное, обычно страдают сами незадачливые пользователи. Майка мне удалось убедить в ошибочности его поведения, когда другой трейдер, воспользовавшись его паролем, устроил беспорядок на его диске, напугав его, но не нанеся большого ущерба. Он испугался, я все исправил, он раскаялся и вступил на путь истинный. (Не в каждом офисе можно вот так схватить за руку.) Но в большей мере проблема связана с разработчиками и архитекторами программ, не понимающими требования и ограничения систем защиты, которые они встраивают (или пытаются встроить) в свои продукты. Это уже гораздо опаснее, потому что оказывает воздействие на каждого, кто пользуется их продуктом. Компьютерная безопасность — весьма специфическое искусство. Обычный прикладной программист практически ничего в ней не понимает, точно так же, как обычный терапевт очень слабо разбирается в нейрохирургии. И все же едва ли терапевт начнет сверлить дырки в черепе пациента, если тот пожалуется ему на головную боль, тогда как прикладной программист, если ему поручат проект системы защиты, в которой он ничего не понимает, вполне может заявить: «Отлично! Где руководство?» Я и сам так действовал, но увидев результаты, в конце концов сделал выводы.

Правильное проектирование системы защиты требует глубокого знания и тщательного анализа поведения пользователей, и программисты отличаются невежеством именно в этой области. Недавно я участвовал в одном обсуждении серьезных технических проблем безопасности, где фигурировали интересные математические задачи. Но я прекратил этот сеанс вопросов/ответов словами «Поправьте меня, если я ошибаюсь, но какими бы замечательными ни были, по вашему мнению, ваши сотрудники, какая от них польза, если они пишут свои пароли на бумажках, которые приклеивают к мониторам?» И мой оппонент признал, что в такой ситуации компьютер становится дорогим пресс-папье, и добавил, что это уже не его проблемы. Мне захотелось его ударить. Человеческий фактор — вот главная проблема, которую должна решать любая система защиты, а они обычно и не пытаются это сделать.

Помнить пароли труднее, чем может показаться. Когда у вас была единственная учетная запись на компьютере (на работе или в колледже),

проблем не было. Сегодня же учетные записи и идентификаторы с паролями, которые для них нужны, растут, как грибы после дождя. Я решил посчитать все учетные записи, которые завел на разных сайтах, и получил ошеломляющее количество — 27. И это только те, которые я сразу нашел, а где-то по углам моего диска затаилось, наверное, еще столько же. Когда в следующий раз сядете за компьютер, посчитайте, сколько их у вас. Почта. Чат. Один музыкальный сайт, другой. Авиабилеты. Агентство путешествий. Банк и инвестиционный фонд, все кредитные карточки и телефонная компания. Порносайт, о котором не должна знать жена. Ручаюсь, у вас наберется больше дюжины. А это означает, что невозможно хранить их так, чтобы это было и надежно, и удобно.

Несовместимые требования

Высокооплачиваемые гуру в области безопасности проповедуют, что пароль должен быть случайным, чтобы потенциальный вор не смог угадать его по каким-либо дополнительным данным о вас — чтобы он не был именем жены, ребенка, кличкой собаки и т. д. Это разумно. Мне случалось вскрывать такие очевидные пароли. Поэтому для выбора пароля я всегда пользуюсь генератором случайных чисел. Еще говорят, что пароль нельзя записывать. И здесь логика понятна. Нобелевский лауреат по физике Ричард Фейнман, который работал над проектом атомной бомбы во время Второй мировой войны, отыскивал плохо спрятанные бумажки с записанными на них шифрами от сейфов с секретными материалами. Он оставлял там дурацкие записки, сводившие с ума тех, кто отвечал за секретность (см. его автобиографию 1985 года «Вы, конечно, шутите, мистер Фейнман»). Далее эти гуру скажут вам, что для каждой учетной записи нужно иметь свой пароль. Как и два предыдущих совета, это имеет свой смысл. Если злоумышленник узнает ваш пароль, взломав не слишком хорошо защищенный сайт отряда бойскаутов, то он не сможет воспользоваться этим паролем на сайте вашего инвестиционного фонда, который охраняется гораздо лучше (по крайней мере, должен). И наконец, вам скажут, что пароль необходимо периодически менять, чтобы в случае его хищения им не смогли пользоваться долго.

Все эти идеи по отдельности совершенно разумны. К несчастью, человек устроен так, что он физически не в состоянии выполнить эти четыре требования одновременно. *Наши мозги устроены иначе. Именно по-*

этому мы изобрели компьютер. В лучшем случае можно одновременно выполнять какие-то два требования из перечисленных. Требовать, чтобы пользователи выполняли их все, и *знать*, что это невозможно, вместо того чтобы пытаться создать максимально надежную защиту, задействовав реальные возможности (а иногда и желания) пользователей, значит бессовестно обманывать людей. С таким же успехом священник может читать проповеди только тем, кто не грешит; это очень просто, но совершенно бессмысленно. По сути, такой подход означает отказ от реального изучения и решения проблемы.

Выбор пароля представляет собой попытку удовлетворить противоречивым требованиям. Можно выбрать случайную строку в качестве пароля и не записывать ее, и даже время от времени менять при условии, что запомнить придется только один пароль (и что вас заставят его менять, потому что сами вы не станете этого делать). Можно выбрать альтернативный вариант, а именно произвольное число случайных строк с частой заменой, если разрешить записывать их в предположительно надежном месте.¹ Либо это будет несколько разных строк, и вы не будете их записывать и даже станете периодически менять, но тогда все они должны следовать какой-то схеме, с которой ваш мозг может справиться, например в первые три месяца года это будут названия рыб, в следующие три — названия птиц или что-то подобное. Наименее вероятен взлом при первом варианте — один (ну, два или три в крайнем случае) случайный нигде не записываемый пароль для всех учетных записей, хотя ущерб при его раскрытии окажется наивысшим. Так и поступает большинство моих знакомых, и, вероятно, это самый безопасный способ. Он не идеален, но и остальные не лучше.

Несовершенство данного подхода в том, что на многих сайтах установлены несовместимые требования к паролям. На одних считается, что символы «А» и «а» различны (учитывается **регистр** знаков), а на других это один и тот же символ. На одних разрешаются не только буквы и цифры, но и символы пунктуации, на других они запрещены, на третьих хотя бы один такой знак обязательно должен присутствовать в пароле, чтобы за-

¹ В «Wall Street Journal» сообщалось о человеке, записавшем все свои 30 с лишним паролей на листок, который он носил в своем бумажнике. Однако этот список все увеличивался, и вскоре разобраться в нем стало трудно. «Подумываю, не воспользоваться ли мне татуировками,» — сказал он. Можно, конечно, но при смене паролей возникнут некоторые трудности.

труднить его взлом. У меня есть одна учетная запись, где символ пунктуации должен не просто присутствовать в пароле, но еще и находиться не на первом и не на последнем месте. А на кафедре в Гарвардском университете требуется пароль из шести цифр. Буквы не принимаются (думаю, это такой способ обойти проблему верхнего и нижнего регистра), знаки пунктуации тоже и цифр должно быть не больше и не меньше, чем шесть. Если бы в Гарварде так не оригинальничали, я держал бы в секрете их пароль, выбрав какой-нибудь из тех, что я помню, но у меня в мозгу нет больше места, чтобы запомнить еще один пароль только потому, что они не хотят быть похожими на других. В результате я пришил конверт с паролем, который они мне прислали, к доске объявлений в моем кабинете — на виду у всех.

Мне также сообщили, что в одной компании администраторы безопасности стали требовать, чтобы пользователи ежемесячно меняли пароли, и при этом в пароле была хоть одна цифра и хоть одна буква. Кажется логичным, но человеческий фактор явно не учтен. Человек не может каждый месяц запоминать новую случайную строку, и никто его не заставит это делать. Что же получилось в этой компании на практике? Мой источник сообщает, что пользователи выбрали себе схему, которую можно запомнить: название месяца и год, например Jan2003, Feb2003 и т. д. Если вор угадает или украдет хотя бы один пароль, ему станет понятна схема, которой, вероятно, руководствуются многие остальные сотрудники, и при смене пароля он сможет определить новый. Это не просто недостаток, это ужасно, и тот администратор, который решил таким способом обеспечить хоть какую-то защиту — полный идиот. Предостережение. Иногда встречаются грамотные администраторы безопасности, которые делают глупость в угоду своему начальнику, считающему себя специалистом в защите. Есть мнение, что как раз по такой причине Германия проиграла Вторую мировую войну. Один высокопоставленный идиот приказал, чтобы скрэмблер в шифровальной машине Энигма менял свое положение каждый день, думая, что тогда его будет сложнее угадать. Напротив, в результате этого обстоятельства количество комбинаций, которые приходилось опробовать Алану Тьюрингу со своими коллегами, сократилось на две трети, благодаря чему им удавалось иногда вовремя завершить перебор. Если бы не это указание, значительная часть информации, добываемой британскими криптографами, поступала бы слишком поздно, чтобы представлять тактический интерес.

Примерно так же, хотя и не столь остро, обстоит дело с идентификаторами пользователей. Открывая в компьютере новую учетную запись, нужно обычно выбрать себе уникальное в данной системе имя, по которому она будет вас опознавать. На одном сайте я зарегистрирован как `dplatt`, но на другом это имя уже кто-то занял, поэтому я зарегистрировался там как `daveplatt`, но на третьем сайте было занято и это имя, поэтому там я стал `daveplatt1`. Я никогда не помню, где у меня какое имя. Это затруднение, в отличие от затруднения с паролями, можно преодолеть эффективно и без труда, но разработчики сайтов этого не делают: отличный ID получается из адреса электронной почты. Он уникален, его легко запомнить, и всюду он одинаков. В некоторых передовых компаниях, например на `Amazon.com`, такой метод применяется. Но есть сайты, где в ID пользователя не допускаются символы (`@`) и точка (`.`), а на других такая строка оказывается слишком длинной. Объяснить такое недомыслие разработчиков невозможно. Им проще было бы написать программу, принимающую любые символы в ID пользователя, чем проверять наличие запрещенных. Какой-то руководитель сказал: «Давайте потратим свое время и деньги на то, чтобы пользователю жилось хуже, да и нам было не легче — просто ради неудобства». Может быть, он и не так сформулировал свое указание, но результат получен именно такой, а у него в мозгу по крайней мере один нейрон лишился синапса. Объем памяти, необходимой для хранения, не может служить здесь препятствием — во всяком случае для того, кто способен думать. Разница между стандартным ID и почтовым адресом из 64 символов может показаться большой, но в современном мире она не существенна. При нынешних ценах на диски для хранения 60 миллионов почтовых адресов достаточно места, которое стоит 1 доллар, причем удельная стоимость памяти ежегодно снижается на половину. Для того чтобы хранить почтовые адреса всех 5 миллиардов жителей нашей планеты, достаточно истратить примерно 100 долларов (опять-таки по сегодняшним ценам, которые стремительно падают).

Некоторые говорят, что такая схема небезопасна, поскольку многим известен ваш почтовый адрес. Говорящие так не понимают проблему. Идентификатор пользователя защищать не надо. Если бы это было не так, то его нельзя было бы показывать открыто на экране при наборе пользователем, и его маскировали бы звездочками, как пароль. В старых системах ID пользователей применялись в качестве адресов в программах для связи или обмена электронной почтой в рамках этих систем (требовалось ввести

команду "Phone PLATT"). На практике их часто показывают на веб-страницах, сообщая пользователю, что он зарегистрирован (рис. 4.1). Если бы Майк сделал Badger своим ID, а не паролем, все было бы правильно.



Рис. 4.1. Имя регистрации пользователя часто используется как элемент персонификации страницы

ID пользователя уникален, но не защищен. Пароль защищен, но не уникален. (Правда, один читатель сообщил мне, что нашел сайт, где ему не позволили задать выбранный им пароль, потому что этот пароль уже выбрал другой пользователь. Еще один пример бессмысленно потраченных разработчиками средств с единственной целью досадить пользователю. На самом деле все пользователи такой системы оказываются менее защищенными, поскольку она может раскрыть пароли.) Такая пара делает вас уникальным и защищенным (по крайней мере, должна). Никто меня так не злит, как тупицы, пытающиеся скрыть свою глупость под прикрытием разговоров о безопасности. Безопасность — это налог на законопослушных граждан. Неоправданное повышение или бесцельная трата этого налога должны рассматриваться как преступление.

Раздача призов тупоголовым: ConsumerReports.com показывал открытым текстом пароль, который вводил пользователь, и это продолжалось больше года — даже после того, как я к ним обратился. Сейчас это безобразия прекратилось, но ведь оно считалось смертным грехом еще тогда, когда я только начал заниматься компьютерами, и, вероятно, гораздо раньше. Даже когда я был первокурсником в колледже и данные вводились с телетайпов, печатающих на бумаге, машина перебивала пароли, так чтобы посторонние не могли их увидеть. Ни одного разработчика или администратора, которые не знают этого элементарного требования, нельзя и близко подпускать к вопросам безопасности в крупных сетях, но именно это и случилось. К счастью, никаких моих ценных данных у них не было. Но такие случаи заставляют задуматься, те ли люди отвечают за защиту

в многочисленных организациях, с которыми приходится иметь дела через Интернет.

На некоторых сайтах в качестве идентификатора пользователя выступает номер его учетной записи, поскольку он участвует во внутренней логике их работы, например это может быть номер активного клиента сайта заказа авиабилетов. Такая тактика в определенной мере оправдана, если есть идентификатор, который постоянно используется в контексте длительных и тесных отношений — например, если это цифровой идентификатор служащего или учащегося на значке, который вы обязаны постоянно носить и часто указывать в деловых документах. Но я вовсе не желаю устанавливать такие близкие отношения с авиакомпанией или универсальным магазином. Заставлять пользователя подчиняться способу внутренней реализации, выбранному программистом, вместо того чтобы разработать интерфейс, который подчиняется способу мышления пользователя, значит проявлять колоссальное неуважение к пользователю, что, к сожалению, слишком часто встречается на практике. Программист, разрабатывающий сайт, не желает облегчить вам жизнь, хотя мог бы это сделать. Один совет разработчикам: установите связь. Возьмите ID пользователя и определите по нему номер учетной записи. Если вы не знаете, как это сделать, или вам лень, лучше уйдите из этого бизнеса, пока не причинили никому неприятностей или пока кто-то не причинил неприятностей вам. Хороший пример и здесь показывает Amazon.com, а для других сайтов, не желающих выполнять данное требование, никаких оправданий нет.

Эти примеры иллюстрируют смысл, который я вкладываю в свои слова о том, что безопасность — это узко специальная область, в которой обычные программисты, как правило, ничего не понимают. Единственный способ для меня и любого другого человека запомнить все эти отдельные ID и пароли для веб-сайтов, с которыми приходится работать, так чтобы сохранить хоть какое-то удобство, это записать их и держать рядом с компьютером. Обычно люди прилепляют такие записи к мониторам или — в условиях повышенной секретности — кладут под клавиатуру. Я храню все ID и пароли для сайтов, связанных с путешествиями, в одном файле на своей машине, потому что иначе я не смог бы пользоваться этими сайтами. Все предельно просто.

Программы для управления личными финансами, такие как Quicken и Microsoft Money, тоже прибегают к такому способу для хранения ID и паролей ко всем банковским и кредитным счетам, с которыми вы рабо-

таете, и если бы не эта возможность, они были бы гораздо менее удобны. С той же целью разработаны некоторые программы управления паролями, такие как Aha (рис. 4.2) и RoboForms. Все ваши пароли они хранят в одном списке в зашифрованном виде, доступ к которому осуществляется через единственный (в идеале случайный и выученный наизусть) пароль, но тут есть свои осложнения и компромиссы, поэтому стандартным средством такие программы пока не стали. Слишком часто я сталкиваюсь с тем, что в больших корпорациях они запрещены: нет, говорят, ты должен запомнить все свои пароли. Я бы с радостью, но больше пары штук мне не запомнить, и никому другому этого не сделать, включая того болвана, который дал это указание. Даже если разрешено пользоваться этими программами, они могут работать только на вашем личном компьютере

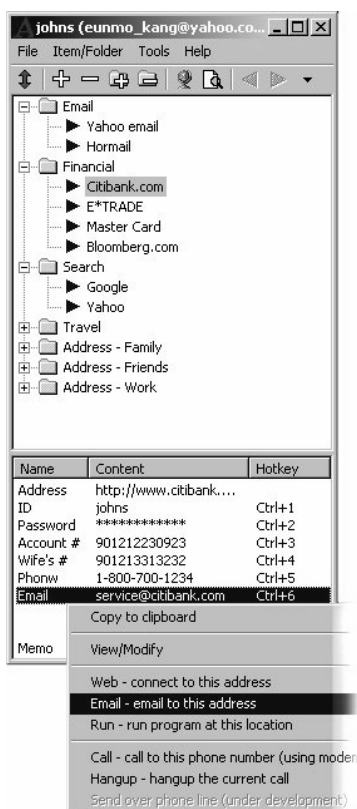


Рис. 4.2. Программа Aha для хранения паролей

и проблему доступа к вашим учетным записям из произвольного места не решают. Возможно, нет необходимости в доступе к пенсионному счету из любой точки, но возможность заказать авиабилет, находясь в произвольном месте, была бы полезна.

Как в древнегреческих трагедиях — чем упорнее мы боремся, тем глубже увязаем. Необходимость защитить каждый отдельный сайт своим особым образом приводит к тому, что никакие сайты не защищены вообще. Враг номер один компьютерной безопасности не пакетный анализатор (устройство, позволяющее как добропорядочным, так и злонамеренным гражданам просматривать трафик в компьютерной сети), о котором я рассказывал в главе 3, а липкая наклейка. Как же все-таки прекратить это безобразие с программами?

Ладно, делать-то что?

Нельзя ли заменить пароли удостоверением личности какого-то иного типа? В случае парольного доступа для установления личности нужно что-то знать — секрет, известный вам и системе. В альтернативном варианте нужно что-то иметь — это может быть карточка-ключ или какая-то часть вас самого, такая как отпечаток пальца. При этом система становится более защищенной, но сложной в работе, поэтому я сомневаюсь, что такая схема доступа может получить широкое распространение в обозримом будущем, и объясню причину.

Можно воспользоваться системой аутентификации на базе электронного ключа или смарт-карты, по существу представляющих собой микросхему, содержащую серийный номер. Зачастую такой ключ изготавливается в форме кредитной карточки, которую нужно вставлять в устройство для чтения (рис. 4.3), хотя начали появляться ключи в форме устройств USB (рис. 4.4). Стоят они около 20 долларов за штуку, и для каждой конкретной модели необходимо установить специальное ПО. Вместо запроса пароля или в дополнение к нему сайт, к которому вы хотите подключиться, попросит ваш компьютер прочесть карточку и послать находящийся на ней серийный номер, и, если он совпадет с ожидаемым, вы будете допущены. (Криптографические подробности защиты от преступного перехвата или фальсификации вашего диалога с сервером я опускаю.) Проблема, очевидно, в том, что пользователь должен физически обладать этим устройством, а сервер должен знать, какое устройство какому поль-



Рис. 4.3. Смарт-карта и устройство для ее чтения

зователю соответствует, чтобы определить его личность. Не так уж плохо для долгосрочных и близких отношений, но для случайной, определяющейся минутными настроениями торговли это смертельно. Вы можете себе представить, чтобы Amazon.com предварительно высылал вам смарт-карту и устройство для ее чтения, чтобы вы смогли купить у него книгу? А как быть с другими сайтами, которые наверняка станут применять другие типы карточек и читающих устройств? У вас на брелке и в бумажнике уже нет места из-за карточек супермаркета, заправочной станции и видеомаркета, так вы хотите еще добавить ключи от каждого сайта, с которым работаете? А как вы сможете работать на компьютере, у которого нет



Рис. 4.4. Смарт-карта SecuriKey в виде USB-ключа

устройств чтения карточек, например на общедоступном компьютере в библиотеке? Находясь там, можно обойтись без доступа к банковскому счету, но доступ к учетной записи активного клиента авиакомпании не помешал бы. Случайная и временная природа большинства коммерческих операций означает, что подобные устройства плохо для них подходят.

В данное время смарт-карты полезны в основном в совокупности с паролями при длительных, требующих повышенной безопасности отношениях. Например, мне нужно через Интернет подключить свой рабочий компьютер к внутренней сети одного клиента, для которого я выполняю некоторую работу. Помимо ID пользователя и пароля (я выбираю его случайным, и они требуют от меня заменять его каждые несколько месяцев) у меня есть смарт-карты и считывающее устройство, которые мне прислала эта компания. Взломщику потребовалось бы не только украсть зарегистрированную смарт-карту, но и узнать ID и пароль пользователя, которому принадлежит эта карта (найдутся и такие, кто запишет ID и пароль на смарт-карте...). Очевидно, хлопот больше, чем если ограничиться просто паролем ID/пароль. Но сеть моего заказчика достаточно секретная, и они настаивают на такой форме доступа, а мое вознаграждение достаточно соблазнительно, чтобы я смирился с неудобствами. (Замечание: такая система не защищает их от мошенничества с моей стороны. Их защищает мой страх перед тем, что они со мной могут сделать за это.)

Некоторые считают панацеей **биометрическую аутентификацию**. Так называют аутентификацию, основанную на неких неизменных узорах, присущих человеческому телу (узоре сетчатки глаза или отпечатков пальцев). Часто это выдают за окончательное решение, поскольку у каждого отпечатки пальцев уже есть и забыть их не так просто. Первые проверки в супермаркетах (на сегодняшний день достигшие наибольшего масштаба в Piggly Wiggly) показывают, что идея нравится большинству покупателей. Им достаточно зарегистрировать свою кредитную карту в магазине, а затем на кассе приложить свой палец к считывающему устройству, иногда для верности еще введя номер своего домашнего телефона. Пожилым гражданам особенно нравится то, что можно не бояться потери или кражи наличных или карточек.

Но помимо проблемы с инвалидами, лишившимися обеих рук (отпечатки пальцев ног? отпечатки носов? протезы отпечатков?), аутентификация по отпечаткам пальцев опасна тем, что их легко украсть и нельзя в случае кражи заменить новыми. Вы же не станете выгирать в ресторане

каждый стакан, из которого пили? Или каждую дверную ручку, которую поворачивали? Нет, конечно. Одно из достоинств паролей в том, что изготовление новых обходится очень дешево, практически бесплатно, поэтому можно выкинуть пароль, когда вы заподозрите, что он скомпрометирован. По этим причинам я ожидаю возникновения отрицательной реакции на устройства чтения отпечатков, как только ими займутся преступники. В крайне редких ситуациях, требующих очень высокой степени защиты, например ракет Трайдент, можно оправдать применение сканеров сетчатки глаза. Но как немыслимо, чтобы Amazon.com потребовал наличия устройства чтения смарт-карт при покупке книги, так еще невероятнее, чтобы им потребовался сканер сетчатки. Замены паролям здесь тоже не видно.

Несмотря на все недостатки паролей, мы не расстанемся с ними в ближайшее время. Какие требования должны предъявить пользователи разработчикам в этой области? Главных два. Во-первых, потребовать по возможности обходиться без них. Веб-сайт должен позволять задержаться на нем на десяток минут, не требуя при этом обещаний вечной верности. (Замечание: не всегда. Например, счет в инвестиционном фонде по своей природе связан с длительными личными отношениями, требующими защищенности везде и всегда. Поэтому данный параграф не относится к сайтам Fidelity и Vanguard для управления счетами.) Например, я только что купил через Интернет пару пижам для своей трехлетней дочки. Магазин (я не буду делать ему рекламу, потому что они мне досадили) потребовал у меня открыть счет с ID пользователя и паролем, иначе они отказывали мне в возможности сделать покупку. Нельзя было просто ввести номер кредитной карточки и адрес доставки — требовалось установить постоянные отношения. Можете вообразить себе реальный, построенный из кирпича магазин, в котором вам отказались бы что-либо продавать без предъявления документов? А этот сайт раздражал еще сильнее, чем такие сайты раздражают обычно. Мне пришлось четыре раза выбирать себе ID, который их удовлетворил бы, каждый раз заново вводя в форму свое имя и адрес, потому что программа сбрасывала эти данные. Их правила для паролей тоже не позволили мне выбрать тот, который я хотел. Если бы я не покупал любимую вещь для дочки («но папа, ты же обещал!»), то я послал бы их подальше после первой же неудачи, а может быть и раньше. Сайт, где покупателю предоставлены обе возможности, показан на рис. 4.5.

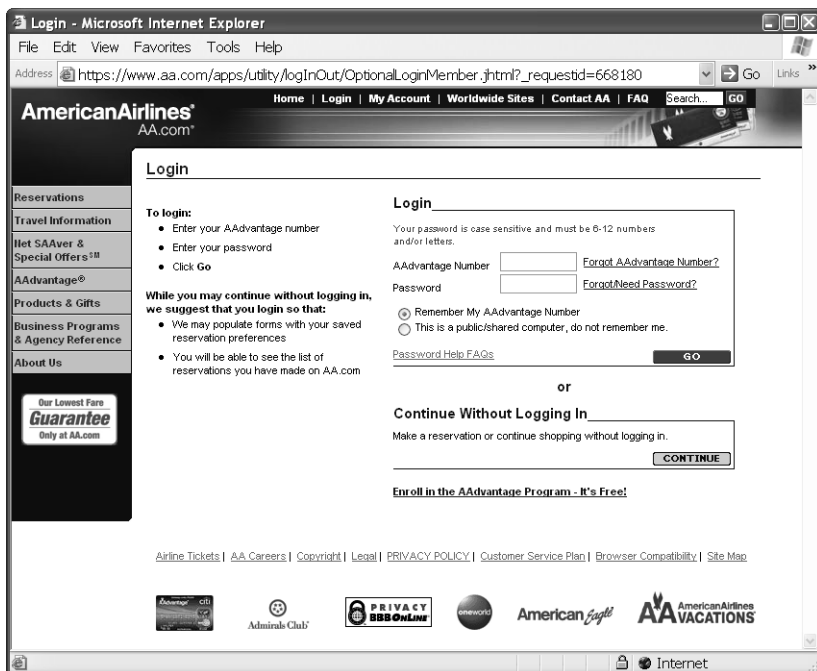


Рис. 4.5. Веб-сайт, который дает пользователю возможность зарегистрироваться, если он этого хочет, либо сделать покупку без регистрации

Больше всего меня раздражает аутентификация, с которой я сталкиваюсь сегодня в сетевых версиях газет, таких как «New York Times» и «Orlando Sentinel». Доступ к этим сайтам бесплатный,¹ но они настаивают на открытии счетов с ID и паролями и требуют их для доступа. Им хочется изучать демографические данные своих читателей и знать, какого рода люди какими статьями интересуются. Разумеется, они не принимают почтовый адрес в качестве ID и специальные символы в составе пароля.

¹ Недавно «New York Times» открыла службу под названием Times Select, которая бесплатно предоставляет большинство статей, но для чтения колонок основных журналистов требуется платная подписка. Первые результаты показывают, что клиенты дружно уходят, и я не сомневаюсь, что к тому времени, как вы будете это читать, данная служба лопнет. Сами журналисты раздосадованы тем, что теряют свои широкие аудитории, обеспечивающие им влияние и значимость.

У них была возможность следить за отдельными пользователями с помощью простых cookie и не причинять никому неудобств, как это делают «Slate Magazine» и «Union Leader» из Манчестера в Нью-Гемпшире. Вместо этого они предпочли досаждать вам и досаждать мне, способствуя неконтролируемому размножению паролей и тепловой смерти вселенной, и ничего при этом не защищают. Естественно, я выдумал ответы на все вопросы. «Times» считает, что мне 113 лет (максимальное допустимое для ввода значение), я женщина и живу в Албании (первая страна в списке для выбора). Тот, кто разрабатывал для них сайт, очевидно не знаком с Нулевым законом информатики, который гласит «мусор на входе — мусор на выходе». Надеюсь, вы тоже им что-нибудь насочиняете. Или обратитесь к услугам сайта BugMeNot.com, распространяющим данные для регистрации, которыми поделились читатели.

Во-вторых, разработчики должны разрешать пользователям свободно выбирать себе ID и пароль. Это значит, что пользователь должен иметь возможность при желании назначить на эту роль свой адрес электронной почты, и отказ его принять должен наказываться публичной поркой. Разработчики существуют для того, чтобы жизнь пользователей стала проще, а не наоборот. И кроме случаев, когда в игре участвуют большие суммы денег, например при управлении банковским счетом, пользователям должно быть разрешено выбирать себе любой понравившийся пароль. Не ограничивайте выбор символов и не требуйте надежный пароль. Пользователь просто запишет этот пароль на бумаге, и надежность этого пароля станет равна надежности бумажной наклейки.

LandsEnd.com правильно делает, что придерживается обоих этих принципов. Рассчитываясь за покупки на их сайте, вы, как обычно, сообщаете адрес доставки и платежные данные, а также свой адрес электронной почты для связи. После этого сайт предлагает запомнить введенные вами данные и использовать ваш почтовый адрес в качестве ID пользователя, для чего вам достаточно ввести пароль. Вполне благоразумно, что в пароле допустимы любые символы (но нет обязательных), поэтому он может быть таким же, как у других торговых площадок Интернета, где вы зарегистрированы. Суть сделанного вам предложения совершенно понятна, и принятие или отклонение его требует минимальных дополнительных действий.

Это лучшее, на что мы способны в настоящее время, но у аутентификации в нынешних ее вариантах остаются два минуса. Во-первых, не все столь удачно ее реализуют — скорее это редкость, а все остальные портят

общую картину. Во-вторых, Сеть все еще находится на ранней стадии своего развития. Это самая быстро меняющаяся область человеческой деятельности из всех когда-либо существовавших. Современные механизмы аутентификации достигли предела своих возможностей. Представьте себе, например, как трудно ввести ID пользователя и пароль на смартфоне по технологии ТТТ.¹ Если мы стремимся к росту и развитию Сети, то необходимо придумать более совершенный механизм аутентификации, чем секретный пароль, известный пользователю и серверу.

Задача управления всеми этими парами ID/пароль иллюстрирует принцип проектирования программных систем, называемый «масштабируемостью». Смысл его в том, что архитектурное решение, пригодное для небольшого количества пользователей, часто оказывается непригодным при попытке применить его к большому количеству. Представьте себе, что вы проектируете телефонную сеть и решили соединить прямыми проводами каждую пару домов, потому что так проще всего. Для соединения двух домов, А и В, достаточно одного провода (рис. 4.6а). Для сети из трех домов требуются три провода (от А к В, от А к С и от В к С), и это тоже достаточно просто (рис. 4.6б). Сеть из пяти домов требует десяти проводов: уже возникает некоторая напряженность (рис. 4.6с). Для сети из десяти домов требуется 45 проводов — слишком много для прокладки и управления. В этом случае мы говорим, что архитектура сети, основанная на прокладывании провода из каждого дома к каждому другому дому, «не масштабируется» свыше четырех или пяти домов. Для большего количества нужна другая архитектура — возможно, с центральным коммутирующим устройством, соединенным одним проводом с каждым из домов (рис. 4.6d). Несмотря на стоимость коммутатора, такая схема окажется дешевле и проще, чем при лавинообразном росте количества проводов. (Для сети из 100 домов к каждому из них нужно проложить 99 проводов, а всего 4950 отдельных проводов. Можете сами посчитать результаты для города с 10 000 домов.)

¹ ТТТ (Triple-Tap Turing — «печать в три нажатия») служит для ввода букв с цифровой клавиатуры, например сотового телефона. Нажав клавишу «2» один раз, вы получите букву «А», нажав еще раз — букву «В» и т. д. Неудобство такого способа ввода имен служит причиной популярности телефонных аппаратов, способных подключаться к телефонной книжке настольного РС.

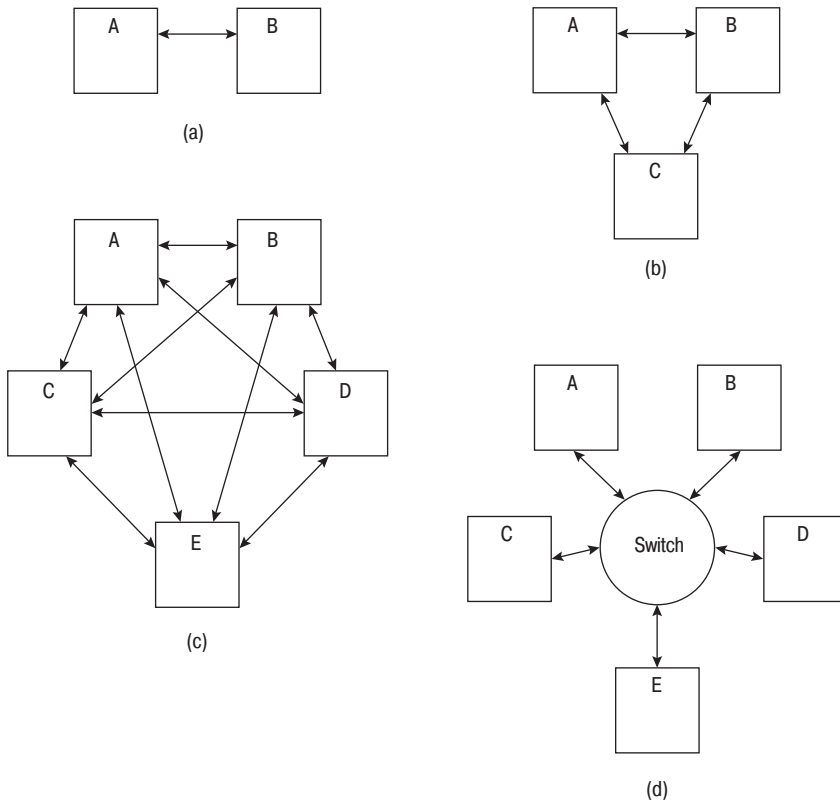


Рис. 4.6. Масштабируемость: а) два дома, б) три дома, с) пять домов с отдельными проводами и д) пять домов с центральным коммутатором

Сегодняшняя схема аутентификации, когда каждый сайт требует свои данные, удостоверяющие личность, и самостоятельно проводит аутентификацию, по описанным выше причинам плохо выдерживает масштабирование. Пользователю приходится заводить для каждого сайта отдельный набор удостоверяющих данных, что, как я объяснил, неудобно и небезопасно по своей сути. Каждый сайт, проводящий аутентификацию, должен самостоятельно вести учет данных регистрации пользователей и организовать его так, чтобы эти данные были доступны для чтения программе аутентификации, но злоумышленники не могли их украсть. В результате возникает колоссальное дублирование работы, которая очень дорого обходится и которую трудно осуществлять правильным образом.

Решение может быть найдено в некоторых других областях. Когда моя мама была молодой девушкой, потребительский кредит существовал исключительно в виде отдельных кредитных счетов у конкретных продавцов. Вы покупали мясо, и мясник прибавлял его стоимость к вашему счету, который надо было оплатить в конце месяца. Так же поступал булочник. Сегодня уже почти никто не ведет личных кредитных счетов. Вместо этого торговля передала всю процедуру оплаты операторам кредитных карт, таким как Visa и MasterCard, которые взяли на себя ведение счетов, платежи и сбор долгов. Коммерсанты могут сосредоточиться на своем основном виде деятельности, будь то продажа верхней одежды или автомобилей. Покупатели могут установить доверительные финансовые отношения с одним, двумя или тремя эмитентами кредитных карточек, и эти отношения будут иметь силу практически повсюду в мире. Ни одной из сторон не приходится устанавливать многочисленные двусторонние связи. Препятствий для совершения покупок стало катастрофически мало, как известно каждому, кто в ужасе смотрел на присланный ему счет по операциям с кредитной карточкой. Подобно тому как осуществление платежей было передано сторонним компаниям, веб-сайты могли бы — нет, просто обязаны — передать проведение аутентификации третьей стороне, где люди а) будут разбираться в том, что они делают, и потому б) будут это делать надлежащим образом.

Это не так безумно, как может показаться. Абстрагируемся от компьютеров и посмотрим, что происходит сегодня вокруг. Есть несколько долгосрочных отношений, которые вы поддерживаете практически ежедневно, например с вашим работодателем или учебным заведением, в которых вам приходится получить от своего партнера удостоверяющий личность документ с вашей фотографией. Может быть, у вас есть удостоверение для супермаркета, в котором вы бываете пару раз в неделю. Но вам же не требуется заводить отдельное удостоверение для каждой случайной отдельной коммерческой операции, где требуется удостоверить личность. Вам не нужно иметь одно удостоверение, чтобы получить посылку через UPS, другое — чтобы сесть в самолет (для каждой авиалинии свое, естественно), или еще новые — чтобы подтвердить свой возраст при покупке выпивки или выписке чека. Если бы вам пришлось устанавливать и сохранять отдельные отношения с каждым винным магазином, это серьезно затруднило бы различные воскресные мероприятия. Все происходит иначе: во всех этих случаях аутентификация (соответствие вашего имени указан-

ному адресату, факт вашего рождения до указанной даты) осуществляется с помощью выданного государством удостоверения, которое должно быть с вами при вождении автомобиля или пересечении границы. Продавцу гораздо проще принять выданное вам неким посредником удостоверение личности, чем требовать, чтобы вы установили с ним личные отношения. Мы уже передали третьей стороне процедуру аутентификации при большинстве деловых отношений. Дело за тем, чтобы перенести эту идею в Сеть и реализовать ее в электронном виде.

В компьютерной области эквивалентный подход называется **единой регистрацией** (single sign-on). Сторонняя организация, называемая **удостоверяющим центром** (trust provider), осуществляет платную аутентификацию пользователей. Когда вы регистрируетесь на веб-сайте, он не сам выполняет вашу аутентификацию, а передает ваши данные удостоверяющему центру, который их проверяет и сообщает сайту-автору запроса, что подтверждает вашу личность (рис. 4.7). В результате вы можете пользоваться одним и тем же идентификатором и паролем на разных сайтах, подобно тому как водительские права служат для удостоверения вашей личности в различных магазинах. Это очень мощная и хорошо масштабируемая система. Удостоверяющих центров, по-видимому, возникнет несколько: свободный рынок создаст выбор, а продавцы решат, чьими услу-

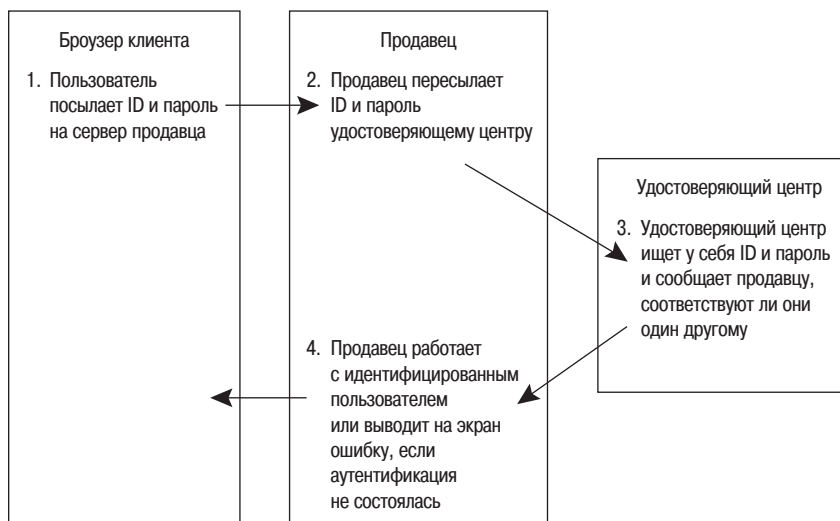


Рис. 4.7. Единая регистрация в реализации Microsoft Passport

гами пользоваться, подобно тому, как сейчас они решают, какие кредитные карточки принимать.

Однако пройдет какое-то время, прежде чем эта система станет массовой. Microsoft пыталась ее реализовать с помощью своей системы Passport еще году эдак в двухтысячном. Если вы пользовались такими сетевыми службами Microsoft, как MSN.com, Hotmail или Instant Messenger, то у вас есть Microsoft Passport. Он представляет собой идентификатор пользователя (которым служит ваш адрес e-mail) и пароль, хранящийся в веб-форме Microsoft. Изначально идея заключалась в том, что любой продавец в сети мог за деньги передать в Microsoft идентификацию своих покупателей. Microsoft рассчитывала, что многие веб-сайты воспользуются этой системой, и все будет прекрасно.

По крайней мере, так предполагалось. Проблема оказалась в том, что почти никто не стал пользоваться для аутентификации паспортами Microsoft, за исключением самой Microsoft. На сайте Passport было перечислено всего 84 сайта, принявших эту систему, из которых 22 принадлежали Microsoft (плюс Starbucks.com, которая тоже могла бы принадлежать Microsoft, если учесть, сколько кофе выпили там ее сотрудники). Это количество меньше прошлогодних 90 — и это в такое время, когда происходит экспоненциальный рост всего, что связано с Интернетом. Теперь даже перестали показывать список торговых площадок, где принимают этот паспорт. Судя по всему, Passport прекратил подключение новых предприятий, и eВау вышел из системы. Очевидно, рынок отверг данную реализацию. (Только пусть команда Passport не шлет мне сердитых писем. Passport — это удобный стандарт для единой регистрации среди многочисленных подразделений Microsoft, и в таком качестве весьма полезен. В качестве предполагаемого универсального веб-стандарта он провалился. Почему, не знаю.)

Главная проблема, сдерживающая продвижение паспортов Microsoft, по-видимому, заключается в том, что клиенты не хотят, чтобы Microsoft хранила их данные аутентификации. Я обсуждал систему Passport со слушателями занятий по .NET, которые я проводил в одной европейской стране в сентябре 2001 года. (До того.) Я поинтересовался, не собираются ли компании, в которых работали эти люди, принять Passport в качестве базовой системы аутентификации. Три человека, представляющие разные банки, сказали, что рассматривали возможность принятия этой системы на своих сайтах для работы с клиентами, но решили отклонить ее.

Слушатель, лучше других владевший английским, объяснил мне при согласном кивании остальных, что «как банки, они считают своим главным товаром доверие. Вы бы доверили своему брату хранить ваши деньги? Нет (видимо, он знаком с моим братом), но вы доверяете это нам (должно быть, он не знает мой банк, которому я доверяю лишь потому, что я должен ему гораздо больше по кредитам, чем он мне по депозитам). По результатам наших исследований и фокус-групп выяснилось, что неприемлемый (цифру он не назвал) процент наших клиентов сочли бы себя преданными нашим банком, если бы мы передали их удостоверяющие личность данные Microsoft, вместо того чтобы хранить самим. Они боятся, что эти данные украдут и ими воспользуются без их разрешения. Я не утверждаю, что они правы в своих опасениях, как не утверждаю, что мы лучше справимся с этой задачей сами, но, нравится это или не нравится, таковы ощущения наших клиентов, судя по их словам. Поэтому, если мы не хотим, чтобы они считали, будто мы сделали худшее, на что способен банк, а именно злоупотребили их доверием, то мы не можем воспользоваться этой системой, и больше говорить не о чем».

Не стану вас утомлять описанием технологий единой регистрации, которые существуют сегодня. Их внедрение — вопрос скорее организационный, а не технический. Например, кто может принять на себя обязанности удостоверяющего центра? Рынок уже сказал свое слово: только не Microsoft. Видимо, также не Sun и не IBM. Может быть, банки, потому что мы уже доверили им свои деньги, а может быть, страховые компании, потому что мы уже поверили их обещанию заплатить нам, если случится что-то неприятное. Может быть, эмитенты кредитных карточек, потому что мы в некоторой степени доверяем им, а они и так уже отвечают за неавторизованные покупки. Может быть, в какой-то форме будет участвовать государство, потому что оно уже занимается выдачей удостоверяющих документов, хотя не все хотят, чтобы оно вело за нами электронную слежку. Как и кто будет оплачивать работу удостоверяющих центров? Как решить проблему курицы и яйца для сайтов, которые не сотрудничают с удостоверяющими центрами, поскольку их покупатели не связаны с ними, а покупатели не связаны с удостоверяющими центрами, потому что нет сайтов, которые их поддерживают?

Со временем все это будет, и тогда регистрация в сети станет менее хлопотной, чем сейчас. Но пока это не случилось:

1. Защита, в особенности проверка данных, удостоверяющих личность, является узко специализированной областью, о которой обычные прикладные программисты имеют слабое представление.
2. Идентификаторы пользователя и пароли имеют тенденцию к неконтролируемому размножению.
3. Тупоголовые программисты (см. п.1) усугубляют проблему размножения (см. п.2), из-за того что часто не учитывают человеческий фактор и даже не понимают, что его надо учитывать (см. п.3).
4. Делегирование аутентификации пользователей третьим лицам — правильная идея, которая вполне осуществима с технической точки зрения, но практичной бизнес-модели для нее пока не предложено.

ГЛАВА ПЯТАЯ

Ну, что смотришь?

Безопасность личных данных очень беспокоит рядового пользователя Сети. Возможно, вы видели карикатуру Питера Штайнера (рис. 5.1). Она появилась 84 собачьих года назад, а по меркам Интернета гораздо раньше. Мы посмеиваемся и тоскливо вздыхаем, вспоминая свою наивность в те времена так же, как когда смотрим на своих детей, которые верят в Деда Мороза и Пасхального кролика. Как бы ни хотелось нам вернуть невинность, мы знаем, что это невозможно. Анонимность в Интернете мнима. Напротив, если только вы не постарались тщательно замести свои следы (а этого не делает почти никто, кроме злоумышленников), ваши собратья в Интернете не только выяснят, что вы собака, но и легко определят породу и возраст, есть ли у вас щенки, и в какие долги вы влезли из-за своей конуры.

Да, им известно, кто вы такой

Э то пугает и беспокоит очень многих. Работая над этой книгой, я постоянно сталкивался с тем, что в нетехнической прессе проблемам сохранения тайны личной жизни уделяется больше места, чем любым другим из области компьютеров — даже вопросам безопасности. Практически все статьи провозглашают, что «прайвеси» (privacy) — что бы ни означало это слово — это Хорошо, поэтому раскрытие какой бы то ни было информации кому бы то ни было — это Плохо, потому что Они (неважно кто) следят конкретно за вами и просто мечтают вывалить самые тайные



«Да в Интернете никто и не увидит, что ты собака»

Рис. 5.1. О, невинность юности! (Из коллекции журнала «New Yorker», 1993, cartoonbank.com, все права защищены)

ваши прегрешения (скажем, что вы ковыряетесь в зубах, когда смотрите «Черную гадюку») на первую страницу «USA Today».

Отчасти эти опасения оправданны. Вы злитесь, что кто-то знает о вас то, что, как вам кажется, ему знать ни к чему. Разве вы не пришли бы в ярость, если бы начальник отдела кадров прочел вашу медицинскую карту и после этого, встретив вас в кафетерии, приветствовал словами: «Здорово, Боб, как сегодня твоя проказа?» Или, как недавно писал наш обозреватель Адам Пененберг (Adam Penenberg) на Slate.com: «...иллюзия приватности и анонимности позволяет нашему подсознанию подавить наше сверх-я. Не будем делать вид, что, путешествуя по Интернету, мы никогда не были охвачены такими мыслями, которыми никогда не поделились бы с друзьями, коллегами, близкими или прокурором. Или скажем по-другому: если вы знаете, что кто-то подглядывает сзади во время вашего поиска в Google, повлияет ли это на предмет поиска?»

И все же тайна личной жизни, как почти все на свете, имеет обратную сторону. Почти такое же негодование возникает, если кто-то не знает о вас

того, что, по вашему мнению, он обязан знать. Например, когда вы звоните в банк или кредитную компанию, они часто просят вас ввести с клавиатуры номер вашего счета и только потом начинают занудно перечислять действия, которые вы можете выполнить, нажав на такие-то и такие кнопки.¹ Когда вас, наконец, соединяют с оператором, он первым делом спрашивает номер вашего счета. «Но я же только что ввел его с клавиатуры! — восклицаете вы. — Зачем он вам снова понадобился, вы, жертва лоботомии?» Или в авиакомпании не могут найти ваш электронный билет, или не могут провести оплату билета по вашему счету постоянного клиента. Или в приемном покое врач, осматривающий вашу дочку, не может получить доступ к записям педиатра, касающимся ее аллергий. Или... ну, вы понимаете.

Теоретически все выступают за личную тайну, пока не встретится хоть малейшее осложнение, ею вызванное, или пока им не предложат пару долларов за то, чтобы поступиться какой-то ее частью. Тогда они начинают выпрыгивать из себя, предлагая свои личные данные за членство в клубе друзей магазина или участие в программе постоянного покупателя. Извините меня, часто ли вы берете на себя труд принять меры, чтобы уборщику в вашем доме или офисе не была доступна ваша секретная информация? Уничтожаете ли вы банковские выписки, чтобы они не попали в руки к мусорщику? Почти никто этого не делает. Какие бы слова вы ни произносили, ваши действия показывают, что вы не считаете это нужным.

В данной главе проблемы сохранения личной тайны в Сети обсуждаются с точки зрения пользователя. Я имею в виду обычных нормальных пользователей, а не пещерных жителей, которые расплачиваются наличными с таксидермистом, чтобы никто их не выследил. (Ребята, да кому вы нужны?) Вопросами неприкосновенности личной жизни на рабочем месте я тоже не буду заниматься. Все знают, что ее там вообще нет, коль скоро ваше рабочее время оплачивается из кармана босса. Я разберу вопрос о том, что и кому известно о программных пакетах, с которыми вы

¹ Мой отец считает, что клиенту, ожидающему соединения с оператором, надо предоставить такие возможности: «Чтобы сказать «подите к черту», нажмите **1**. Чтобы сказать «если мой звонок для вас так важен, почему вы не наймете человека, который ответит на него, или вы считаете меня идиотом, способным поверить в ваше вранье?», нажмите **2**. Чтобы сказать «Я сейчас достану пушку и перестреляю вас всех, каналы», нажмите **3**». И т. д.

работаете, и о ваших привычках в Сети, насколько это должно вас беспокоить, и какими могут быть меры противодействия, если вы решите, что это для вас важно.

Отстой крепчает беспримерно

Повсеместное распространение Интернета и тот факт, что все интеллектуальные устройства на свете соединены между собой, оказывают влияние на все сферы человеческого поведения. Неудивительно, что картина сохранности личной тайны тоже претерпела изменения. По-моему, все происходит примерно так.

Во-первых, есть данные, которые всегда были общедоступными, но теперь их стало гораздо легче получить, а это значит, что гораздо большее число людей может изучить их и попытаться использовать в своих целях. Например, данные о сделках с недвижимостью — такой-то продал этот дом такому-то такого-то числа за такую-то сумму — в моем штате публикуются лет 300, но лишь последние лет пять или около того стало возможным заполучить их без особого труда. Раньше надо было подъехать к Бюро регистрации собственности в их приемные часы, найти место на парковке, подойти к зданию, подождать в очереди, пока получившие эту работу по знакомству канцеляристы допьют свой кофе, найти нужную книгу, если она на месте, и т. д. Трудоемкость такой процедуры означала, что никто не станет ей заниматься без особой надобности. Но сейчас все зарегистрированные сделки в Массачусетсе выкладываются в Интернет, поэтому любой желающий может, не выходя из своего офиса, запустить браузер и просмотреть эти данные.

Вроде бы все правильно? Открытые данные должны быть доступны публике по определению, так почему бы не упростить процедуру доступа к ним? Работая над этой главой, я полюбопытствовал, сколько мой товарищ по колледжу выложил за дом, который он купил в элитном местечке неподалеку. Я нашел эти данные на сайте бюро регистрации меньше чем за минуту. Но я также узнал, что первоначально его жена купила этот дом на свое имя за несколько месяцев до того, как они поженились, а его имя внесла в акт лишь несколько лет спустя, примерно в то время, когда у них появился второй ребенок. Это вполне логично, поскольку я хорошо знаю его кредитную историю (он мне должен деньги еще с 1977 года), но следует ли об этом знать посторонним? Скорее всего нет, и я чувствую себя

не совсем ловко из-за того, что сунул свой нос в чужие дела. Благодаря нынешней легкости получения открытой информации подобные вещи становятся известны гораздо большему числу людей. Поэтому приходится вести себя предусмотрительней, чем прежде, и стараться не влезть ненароком в чужие дела.

Вторая угроза личной тайне в современном компьютерном мире исходит от новых способов комбинирования и сопоставления имеющихся данных. Например, бакалейные магазины всегда регистрировали, сколько каких товаров они продавали в каждую неделю года, чтобы предвидеть будущей спрос и иметь достаточные запасы. Самые толковые учитывали связь суммарного спроса с внешними обстоятельствами, такими как погода, — если на улице жарче, чем обычно, надо заказать больше мороженого. Но сейчас, когда вы предъявляете им карточку постоянного покупателя, чтобы получить скидку, они регистрируют все, что вы купили на этой неделе, и сопоставляют с тем, что вы купили на прошлой и за предыдущий месяц; эту процедуру называют *извлечением информации из данных* (*data mining*). (Обычно можно попросить не делать этого, но такое требование должно быть выражено явно; оно называется *отказом от участия* (*opting out*).) В результате магазин может обнаружить, что покупатель корма для кошек часто покупают крупных индюшек за неделю до Дня благодарения, а к ним еще красного вина, тогда как покупатели корма для собак ждут до последнего момента, а потом покупают маленьких индюшек и запивают их белым вином. После этого можно печатать при расчете индивидуальные купоны и корректировать соответствующим образом запасы. («Ага, малый с кошачьим кормом: печатаем купон на большую индюшку и заказываем дополнительно красное вино».) Когда в большинстве супермаркетов маржа прибыли ничтожно мала, подобный интеллектуальный анализ может оказаться решающим для выживания. Его качество, возможно, является главным преимуществом Wal-Mart.

Некоторых тревожит, что торговые предприятия могут каким-то образом воспользоваться этой информацией в ущерб своим покупателям. Других пугают не столько магазины, сколько то, что к собранным данным получают доступ правительственные органы. Например, прокурор Кеннет Старр пытался заполучить записи о покупках Моники Левински в книжном магазине, предполагая, что они могут пролить свет на ее отношения с Биллом Клинтоном. Сегодня самым ярким примером такого рода служит Акт о патриотизме, который позволяет ФБР без ведома клиента чи-

тать записи о книгах, которые он берет в библиотеке (при наличии ордера на обыск, подписанного судьей, о чем часто забывают при обсуждении этого закона). Я предложил директору моей местной библиотеки стикер на бампер с текстом «бойтесь правительства, которое боится вашего читательского билета».

Третья сильная угроза личной тайне вызвана появлением новых, ранее не существовавших типов данных. Например, в течение примерно 15 последних лет можно было, находясь в любой точке планеты, легко узнать свои географические координаты с помощью приемника Спутниковой системы навигации (GPS) размером с книжку карманного издания и стоимостью несколько сотен долларов. Но в последнее время эти приемники стали настолько миниатюрнее и дешевле, что их часто встраивают в обычные сотовые телефоны, например Nextel i88s за \$80. Этот телефон может передать свои текущие координаты по радио в некую базу данных, к которой при наличии соответствующих прав доступа можно обратиться через Интернет. Теперь компании могут отслеживать местонахождение своего транспорта, родители — своих детей,¹ жены — своих мужей. Подобная информация лишь недавно стала доступной обычным гражданам. Кому разрешить доступ к ней, к какой части и при каких условиях?

Если вы вспомните, как называется эта книга, то решите, что я сейчас начну во всем обвинять разработчиков программ. Но данная глава составляет исключение. По большей части угроза личной тайне, исходящая от компьютеризации, не связана с бестолковостью программистов, которые пишут плохой код из-за того, что не понимают своих клиентов. Нет, она создается безобидными программами, сносно решающими свою задачу, но стоящими на службе людей, чьи ценности и интересы противоречат нашим. Подобно программистам, эти типы, свихнувшиеся на маркетинге (см. главу 6), не понимают, что вы непохожи на них самих, и потому не знают, когда сбор неких данных о вас вам безразличен, а когда приводит в ярость. Это все та же проблема («познай своего пользователя, ибо он не такой, как ты»), но относящаяся к другой группе людей.

Все же при разумном подходе и небольших модификациях правильно сделанное ПО может действительно защитить приватность пользователя. Для этого, однако, гикам придется сделать то, чем они раньше никогда не занимались, а именно попытаться понять своего пользователя и убедиться

¹ См., например, www.teenarrivealive.com.

в том, что он не такой, как они сами. (Вы замечаете некоторую систему, свойственную этой книге?) У программистов надежды и тревоги совсем не такие, как у пользователей их программ, и уступки, на которые они готовы в отношении тайны своих личных данных, также различны. Программа, которая просто дает пользователю возможность управлять мерой сохранения тайны его личных данных, окажется полезной только тем, кто имеет представление о том, какие данные важно защищать, а какие нет, кто готов разобраться в программе и сможет самостоятельно настроить ее приемлемым образом. То есть практически никому. Программа для обычного пользователя должна учитывать его неосведомленность в том, что опасно, а что нет, и предоставить разумные значения по умолчанию для разумных пользователей сразу после установки. Не верите? Привожу пример.

Пользователи не знают, в чем состоят риски

Сообщите мне номер вашего банковского счета — чекового счета, на который приходит и почти тут же уходит ваша зарплата.¹ Пойдите на сайт этой книги, заполните форму и отправьте мне электронной почтой. И не забудьте указать название и адрес банка, чтобы я точно знал, куда идти за вашими деньгами.

Что, не хотите? Не просто «не хочу», а «Ни за что, Платт! Я не верю вам ни на грош, тем более в таком деле, как эти важные данные. Никогда не раскрою их, черт бы вас побрал с такими вопросами».

Не могу вас винить за это. Я тоже не стал бы доверять мне, даже не из-за этой книги. Такой важной информацией, как номер вашего банковского счета, нельзя делиться ни с кем. В моей местной газете только что написали о появлении в нашем районе жуликов, которые звонили пожилым гражданам и, выдавая себя за инспекторов банка, просили их назвать номера своих счетов, чтобы поймать преступников. Полиция рекомендует никогда не раскрывать такую информацию, и совершенно справедливо. Я даже советую вам не вводить на моем сайте номер своей кредитной карточки, чтобы купить эту книгу. Действовать следует действительно на-

¹ Считается, что деньги умеют говорить, и это правда. Я сам слышал однажды, как они сказали: «Пока, Платт! Приятно было греть твой бумажник целых две микросекунды».

дежным способом, по-старинке, даже если это займет больше времени: распечатать бланк на бумаге и послать мне по почте чек. Вот так будет безопасно, верно? Ну, не совсем. Что находится внизу каждого чека в вашей чековой книжке? Верно, номер счета и название банка. Их даже предусмотрительно напечатали магнитными чернилами, чтобы можно было автоматически сканировать. Можно даже не бояться сделать ошибку при заполнении. Ох!

А что обычно напечатано сверху чека? Адрес. Правда, можно заказать чеки без адреса, но, если вы часто их выписываете, то обычно удобнее иметь адрес напечатанным, потому что его требуют продавцы, которые принимают чеки. Иногда на чеке печатают даже номер водительского удостоверения (опять эти торговцы), а для многих покупателей это также их номер социального страхования. Ох-ох!

Ну а теперь — чтобы наверняка получить награду за тупость — что там такое в правом нижнем углу чека? Верно, способ аутентификации пользователя, существующий столько же, сколько письменность: ваша подпись, сделанная вручную чернилами. Ее всегда было очень трудно подделать — требовались ловкость, доступная не всякому, и долгие годы тренировок. Сегодня простой сканер за 200 долларов заглотает ее, а потом выдаст обратно на поручении перевести деньги с вашего счета на мой. Угу.

Вы всегда думали, что чек — самый надежный способ перевода денег, но все изменилось, верно? Чеки не изменились, но изменился мир, в котором они обращаются. Живя в этом мире, вы, вероятно, заметили происшедшие изменения, но не подумали о том, какое отношение они имеют к обычной вашей чековой книжке. Сказанное должно было убедить вас, как трудно обычному пользователю определить, какая информация является критической, а какая нет. Еще труднее разобраться в способах разглашения или кражи критической информации. Да это и ни к чему. Пассажир не должен разбираться в том, при какой погоде безопасно летать, — это обязанность летчиков. Больные не должны знать, какие лекарства при каких болезнях можно принимать, как они взаимодействуют между собой и какие могут быть побочные эффекты, — это дело врачей. С какой стати людям кажется, что они знают, какую информацию не опасно разглашать? Они не знают, не могут и не должны знать. Программное обеспечение должно это делать вместо них, для того его и покупают.

Что становится известным прежде всего

Каждый компьютер в Интернете идентифицируется так называемым **адресом интернет-протокола (IP-адресом)**. Можете считать этот адрес телефонным номером компьютера в Интернете (хотя он не имеет ничего общего с тем номером, по которому вы звоните провайдеру, если вы все еще пользуетесь таким способом доступа в Интернет). У веб-сервера IP-адрес фиксирован, т. е. завтра он будет таким же, как сегодня, даже если сервер выключится, а потом снова включится. Например, у Microsoft.com постоянный IP 207.46.130.108, а у Amazon.com — 207.171.166.102. Введя эти цифры в окно адреса своего браузера, вы попадете на эти сайты. Когда вы вводите читаемый адрес типа «Microsoft.com», ваш браузер ищет на другом сервере (сервере имен) числовой адрес для этого имени и потом запрашивает сайт с этим номером. (В некоторых старых браузерах панель состояния действительно показывала, как происходит этот поиск.) Это примерно как если бы вы сказали в трубку телефона «Дэвид Платт, Ипсвич, Массачусетс», и телефон нашел мой номер, а потом набрал его.

Когда вы подключены к Интернету, у вашего компьютера тоже есть IP-адрес, назначенный вашим провайдером. Он называется «динамическим», потому что может меняться в разных сеансах подключения. Если у вас модемное соединение по коммутируемой линии, ваш провайдер присваивает вам IP-адрес, когда вы дозваниваетесь до него, и отбирает этот адрес, когда вы вешаете трубку. Этот адрес он может присвоить следующему дозвонившемуся пользователю. В случае кабельного или DSL-модема IP-адрес также присваивается при соединении с провайдером и теряется при отключении. Однако соединения такого типа часто поддерживаются постоянно, и один и тот же IP-адрес может сохраняться до очередного отказа электрического питания. Если ваш кабельный модем подключен к источнику бесперебойного питания, один и тот же IP-адрес может сохраняться месяцами и даже годами.

Когда вы запрашиваете веб-страницу, сервер сайта видит IP-адрес, с которого вы делаете запрос, так же как функция «caller ID» голосового телефона показывает номер звонящего. Серверу необходим этот адрес, чтобы знать, куда отправлять запрошенные данные. Можете узнать текущий адрес своего компьютера, зайдя на сайт этой книги, выбрав страницу главы 5 и перейдя по ссылке «демонстрация IP-адреса». Вы увидите

страницу с вашим текущим IP-адресом и некоторыми другими данными о вас, которые мне удастся выяснить.

Как только сайт узнал ваш IP, он может поинтересоваться в различных открытых базах данных, какой провайдер владеет этим адресом. Адреса присваиваются разным провайдерам особыми регулирующими органами, так же как телефонные номера присваиваются различным телефонным компаниям. Географическая точка, в которой ваш провайдер подключен к остальному Интернету, также обычно известна. Следовательно, сайт, к которому вы обратились, может относительно легко определить страну, из которой исходит запрос. Когда вы в следующий раз попадете на сайт, который заставит вас выбирать название вашей страны (UPS.com, Guinness.com), знайте, что они ленивы или глупы, потому что могли бы узнать ее по вашему IP-адресу, но вынудили вас делать свою работу. Если вы уже прочли главу 2, то вспомните, как я чехвостила UPS.com за то, что они пренебрегают этим приемом. Пошлите им почтовое сообщение с тем же упреком, хорошо? А после этого перейдите на обслуживание компанией, которая заинтересована в ваших удобствах (с UPS это легко, с Guinness — сложнее).

Интернет-провайдер обычно ведет журнал, в котором регистрируется выдача IP-адресов пользователям. Поэтому, когда вы входите в Интернет под своей учетной записью, сервер может без особых хлопот точно определить, кто вы такой. Хотя большинство провайдеров заявляет, что не раскрывает личных данных пользователей, обычно в их политике содержатся уловки, допускающие при наличии решения суда такое раскрытие и нарушение обычных правил. Если вы ведете себя прилично, вас вряд ли станут вычислять. Но когда при выяснении источника спама или другой атаки выйдут на IP-адрес у вашего провайдера, его свяжут с вашим компьютером.

Если вам не нравится, что сайт может вас вычислить, нетрудно помешать ему это сделать. Можно воспользоваться общедоступным компьютером, например в библиотеке или магазине Kinko's (не забудьте расплатиться наличными и не попадайте под камеры системы безопасности, либо измените внешность), который представляет собой интернет-разновидность таксофона. Либо воспользуйтесь службами анонимности, о которых я расскажу далее в этой главе, чтобы ваши запросы выглядели, как поступившие с другого IP-адреса. Однако мало кто утруждает себя этим. Большинство людей не возражает против того, чтобы их интернет-про-

вайдер контролировал область их интересов в Сети — если вообще задумывается о таких проблемах, что случается редко.

Донуть клиентов с помощью cookies?

Виновником многих нарушений тайны личных данных в Сети оказывается злоупотребление вещицей под названием **cookie**. В прессе по их поводу возникает много шума, но говорится мало толкового, и у многих возникло путаное представление о том, что это такое и как оно работает. «Это здорово!» — восклицает автор одной статьи. «Это плохо!» — возражает другой. — «Слежку за людьми устанавливают не cookie, а другие люди». — «Да, другие люди, с помощью cookie». Да будет известно всем: cookies не вредны по своей природе. На самом деле они необходимы для создания таких условий работы в Сети, которые требуются большинству пользователей. Но им можно найти и такие применения, от которых вы вряд ли будете в восторге. Смотрите.

Как говорилось в главе 2, на заре существования Сети все страницы были статическими. То есть их содержимое всегда было одинаковым для всех и не зависело ни от каких данных, полученных от пользователя. Автор веб-страницы, скажем, сел и печатал список всех фильмов, которые шли в местном кинотеатре, и этот список можно было прочесть, направив свой браузер на эту страницу. Если бы это было все, что требовалось от веб-страниц, никакой нужды в cookies и не возникло бы.

Проблемы возникают, когда мы хотим подогнать содержимое веб-страницы под вкусы и потребности конкретного пользователя. Допустим, к примеру, что вы вводите в окно адреса браузера *www.cnn.com*, чтобы увидеть главную страницу этого крупного информационного агентства. У CNN есть две редакции главной страницы, одна для США, другая для остального мира. (Можете убедиться в этом сами.) По вашему IP-адресу сервер CNN определяет (см. предыдущий раздел) страну, из которой, вероятно, был сделан запрос, и сначала показывает страницу, соответствующую этой стране. Однако определение по IP-адресу могло быть неправильным, вы могли уехать в другую страну, но соскучиться по родине, а может быть, просто хотите посмотреть на другую страницу. На этот случай CNN помещает вверху страницы ссылку, с помощью которой можно изменить страну (рис. 5.2 и 5.3).

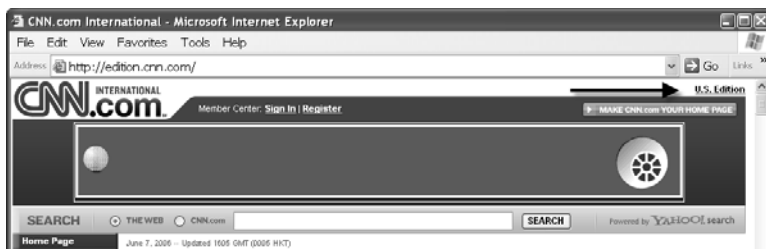


Рис. 5.2. CNN.com со ссылкой на международную редакцию

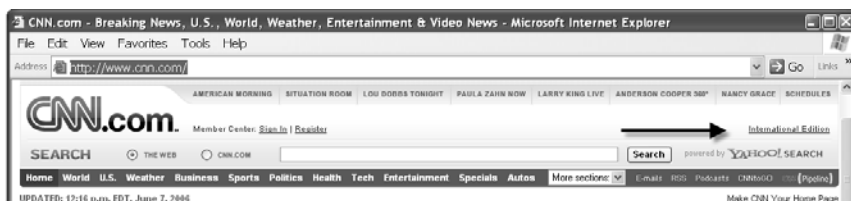


Рис. 5.3. CNN.com со ссылкой на национальную редакцию

Допустим теперь, что вы смотрите на главную страницу CNN.com и щелкаете по ссылке, чтобы перейти к некой статье. Откуда серверу знать, какую редакцию главной страницы вы разглядывали, а потому какое форматирование (международное или для США) применить к новой странице? В запрос веб-страницы эта информация не включается автоматически, потому что в те давние времена, когда разрабатывался протокол запросов, никто не думал о таких вещах. Этот протокол устроен так, что соединение между браузером и сервером теряется после каждого запроса страницы, — он, как говорят, **не хранит информацию о состоянии**. Иначе дело обстоит при телефонном звонке, когда соединение сохраняется, пока вы не повесите трубку. Сервер не знает, от какого пользователя поступил новый запрос. (Знать IP-адрес недостаточно, потому что в новом запросе владелец адреса мог успеть смениться, а при совместном использовании одного соединения несколькими компьютерами их запросы могут иметь один и тот же IP-адрес отправителя.)

Чтобы имитировать диалог в рамках соединения, веб-программисты пытаются сохранить информацию о первой запрошенной странице и передать ее серверу при запросе следующей страницы, каким бы временным интервалом ни были разделены оба запроса. Делается это с помощью

cookie, небольшого текстового файла, который располагается на жестком диске компьютера или иногда в памяти браузера. Когда вы выбираете ту или иную редакцию CNN.com, сервер записывает данные о вашем выборе в cookie для CNN.com (и при необходимости создает этот файл) на вашей машине. Когда вы в следующий раз запросите страницу у CNN.com, сервер узнает из этого cookie, какую редакцию вы выбрали, и соответствующим образом отформатирует новую страницу. В cookie можно сохранять множество признаков персонификации страницы, например язык, на котором ее нужно вывести.

Для каждого сайта на вашей машине есть свой собственный cookie. Сервер сайта может прочесть только те cookies, которые разместил сам. Браузер не позволит ему прочесть другие, и невыполнение этого ограничения есть ошибка браузера. Таким образом, Amazon.com не узнает, чем вы занимались на BarnesAndNoble.com, а Guinness.com не увидит ваши настройки для Budweiser.com. Cookie невелики по размеру: обычно не более нескольких сотен символов. Если поискать на диске слово *cookies*, то можно найти папку, где их будет целая куча (рис. 5.4). Однако нанести вред вашей машине они не могут. Это простые текстовые файлы, не про-

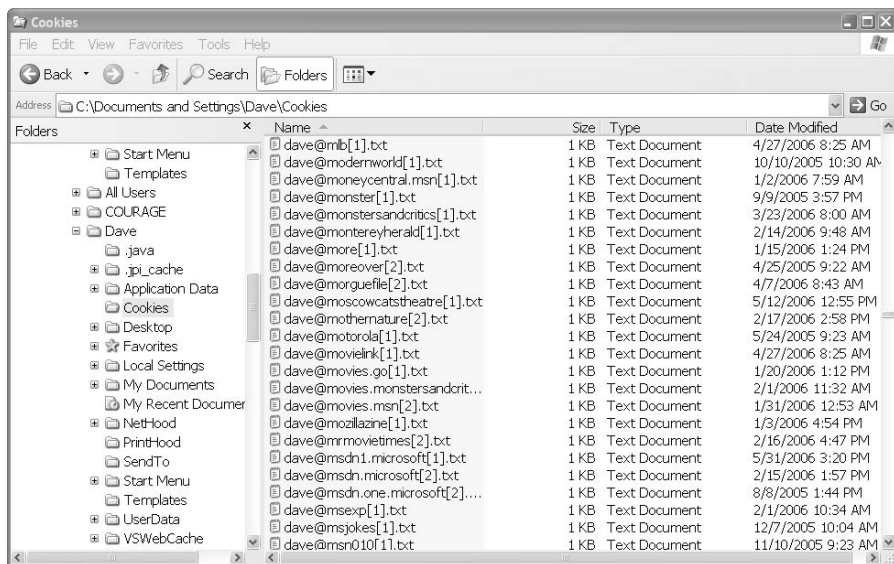


Рис. 5.4. Папка Cookies

граммы, и они не могут выполняться. Они нужны только для хранения некоторой информации, помещенной на ваш диск конкретным сервером.

Cookies часто хранят ID зарегистрировавшегося пользователя, что бывает удобно на сайтах, не требующих высокой степени безопасности. Например, на сайтах многих газет, скажем «New York Times», для чтения основного объема содержимого требуется ввести ID и пароль. Можно, однако, запомнить свои данные идентификации, чтобы не вводить их каждый раз. Веб-сервер поместит ваш ID в cookie. Ваш пароль туда записан не будет, но будет сохранено специальное значение, показывающее, что пароль проверялся и был признан верным. Сайты, которые запоминают большой объем информации о клиентах, например полную историю всех покупок на Amazon.com, не записывают все в один cookie. Эту информацию они записывают в свои внутренние базы данных, а в cookie хранят только ваш ID, по которому находят все ваши данные в своей базе.

Cookies удобны не только для пользователей, но и для разработчиков и администраторов сайтов. Например, разработчик может проследить маршрут движения пользователя по сайту, даже не зная, кем является этот пользователь в реальном мире: некто зашел с этой страницы, потом перешел на ту, потом ушел с нашего сайта на такой-то другой. Пятьдесят других пользователей проделали то же самое, но десять перешли со второй страницы на другую страницу нашего сайта. Возможно, стоит поместить на второй странице больше ссылок на полезный контент, чтобы пользователи не уходили с сайта, или поменять макет страницы, чтобы ссылки были лучше видны.

Cookies также применяются в рекламе. Они могут запоминать, какие рекламные объявления вы видели и по каким щелкали, что помогает точнее отбирать рекламу для показа. Например, если вы недавно щелкали по рекламе лодок, сайт может показать вам больше рекламы, касающейся лодок, и меньше — касающейся гольфа. Либо может выясниться, что интересующиеся лодками покупают больше выпивки, чем средний покупатель (для употребления во время плавания, надо полагать), поэтому, если вы щелкнете по рекламе лодок, вам станут показывать больше рекламы спиртного. Обычно сайтам платят за рекламу, только если по объявлению щелкнут, а не просто за показ. Для сайтов, живущих за счет рекламы, таких как CNN, стимулирование щелчков по рекламе решает вопросы жизни и смерти. Торговые площадки, которые адаптируют свои страницы в соответствии с предыдущими покупками клиента, например Amazon.com, по-

казывающий рекомендованные товары, тоже основывается на данной технологии.

Теперь вы сами видите, что сайтам нужны cookies, чтобы адаптировать страницы, чего требуют пользователи, и для решения жизненно необходимых коммерческих задач. Кажалось бы, подобный симбиоз должен удовлетворить обе стороны. Так откуда вся эта суета, и не малая, вокруг cookies? Во-первых, ряд пользователей принципиально не желает предоставлять никакому веб-сайту никакой информации. «Не вашего ума дело, молчите и не спрашивайте», — говорят они. Другие ненавидят рекламу и не хотят предоставлять никакой информации, которая может сделать рекламу более эффективной. (Эти пользователи часто покупают утилиты типа AdSubtract и GhostSurf, которые удаляют все рекламные объявления, оставляя вместо них на экране пустое пространство.) Некоторые пользователи считают, что их права нарушаются, когда некий подлый сервер записывает что-то без спросу на их машину. Увы, отсутствие памяти в протоколе веб-соединения делает технические альтернативы сложными и дорогими.

Проблемы нарушения личной тайны возникают, когда сайт соединяет регистрацию поведения пользователя в сети с конкретным физическим лицом. На большинстве сайтов газет можно указать фальшивые данные о себе, и в главе 4 я предлагал делать это назло. Но сайт, где вы совершаете реальные покупки, например Amazon.com или сайт авиакомпании, должен установить вашу подлинную личность, чтобы проверить сделанные вами платежи. Немало людей обеспокоено тем, что эта информация может когда-нибудь оказаться не там, где нужно. Обычно их тревога длится несколько секунд, а потом, однажды попробовав и убедившись, что ничего с ними не случается, они спокойно живут дальше. Но когда-нибудь в будущем могут возникнуть неприятные последствия. Например, вас выдвигают в состав кабинета, а некий репортер раскапывает запись вашего серфинга в сети: «В 2005 году вы 17 раз искали на Amazon.com книги о пытках, правда не купили ни одной. Что вы разыскивали?» В заявлении о сохранении конфиденциальности данных (см. следующий раздел этой главы) Amazon обязуется не раскрывать такой информации, но я сомневаюсь, что это обещание будет выполнено, когда сойдутся честолюбивый репортер, ищущий сенсации, и администратор базы данных, обуреваемый праведным гневом против ваших политических взглядов. Если вас беспокоят возможные проблемы такого сорта, воспользуйтесь одним из

продуктов, обещающих анонимность, о которых я расскажу далее в этой главе, либо поезжайте в магазин, стены которого возведены с помощью кирпича и цемента, и расплатитесь наличными, либо придумайте себе хорошее алиби. («Я искал материалы о том, чего *нельзя* делать!»)

Проблема становится особенно острой, когда разные сайты начинают объединять свои данные. Как раз этим занимался DoubleClick.com в 1999 и 2000 годах. Возможно, вы помните, какой поднялся шум — выпуски новостей редко уделяют столько внимания проблемам программного обеспечения.

Многие сайты сами не управляют размещением рекламы, поскольку это дорого и сложно. Вместо этого они доверяют агентствам типа DoubleClick.com продавать рекламные объявления, предоставлять их для показа, следить за тем, какая реклама на какого пользователя успешнее действует, а сами получают в конце месяца чек за щелчки, сделанные посетителями. Политика в области конфиденциальности личных данных таких различных сайтов, как PizzaHut.com, Travelocity.com и Shoeline.com, указывает в качестве их рекламного агента DoubleClick, а сам DoubleClick утверждает, что так же поступает более 11 000 других сайтов.

Проблема обозначилась, когда DoubleClick начал объединять данные со всех сайтов, которым он поставляет рекламные объявления. С помощью не очень достойного трюка он смог размещать на машине пользователя cookie, оказывавшийся доступным с любого сайта, который показывал рекламу DoubleClick.¹ В результате можно было при желании выяснить, что покупатель, изучающий обувь на Shoeline.com, только что купил у Travelocity тур на Гавайи, и потому ему лучше показать рекламу сандалий, а не зимних сапог. (DoubleClick утверждает, что пока они не дошли до такой степени детализации, а действуют на уровне сектора рынка — например, спорт, домашние животные, путешествия и т. д.)

¹ Веб-сайт должен был разместить на каждой своей странице крошечное изображение, часто состоящее из единственной точки. Это изображение, называемое также **веб-маячком** (web beacon), было помечено как поступающее с DoubleClick.com. Когда браузер собирал страницу, чтобы показать ее пользователю, он должен был запросить эту картинку с DoubleClick.com. В результате сервер DoubleClick мог видеть свой cookie, независимо от того, какую страницу запрашивал пользователь. Этот прием известен как **сторонние** (third-party) **cookies**.

Многим пользователям это не понравилось. Некоторые выдвинули против компании иски, которые она выиграла, т. к. суд решил, что не доказан ущерб, нанесенный покупателям в результате объединения данных. Опасения пользователей еще более усилились, поскольку DoubleClick приобрела крупную компанию маркетинговых данных Abacus, у которой была огромная база данных о клиентах, делавших покупки по почте. DoubleClick пригрозила — простите, пообещала — своим клиентам объединить эти данные с данными cookie и завести учет веб-серфинга с персональной идентификацией по всем сайтам, для которых она осуществляла рекламное обслуживание. Несмотря на победу DoubleClick в суде, скандал послужил падению цен на ее акции, и компания отступила. По крайней мере, временно.

Как и все в жизни, cookies имеют обратную сторону. Я трачу кучу денег на Amazon.com. Меня бы сильно рассердило, если бы работник магазина, в котором я потратил такие деньги, не запомнил меня или каких-то из моих предпочтений. С другой стороны, даже такой технофил, как я, видит проблему в том, любой может за 20 долларов получить доступ к операциям конкретного лица в Сети. Мне иногда бывает интересно, что в последнее время смотрел мой товарищ по колледжу — тот, чьи сделки с недвижимостью я разнюхал, и я готов заплатить \$20 за возможность поставить его в неловкое положение на очередной встрече выпускников.

Большинство браузеров допускает некоторую степень контроля над cookies. Но до недавнего времени этот контроль был весьма убогим. Можно было полностью отключить cookies, но тогда переставали работать многие удобные функции сайтов. Вводить каждый раз ID и пароль, чтобы прочесть «New York Times», «Orlando Sentinel», «Washington Post» и т. д., очень надоедает, когда знаешь, что достаточно выставить один флажок, чтобы избавиться от такой необходимости. Другие сайты, в частности популярные бесплатные почтовые службы Hotmail и Yahoo!, вообще не будут работать без cookies. Можно настроить браузер так, чтобы он спрашивал разрешения каждый раз, когда требуется cookie, но тогда перед просмотром каждой страницы придется три-четыре раза задумываться и нажимать кнопку OK или Cancel. Это надоедает через пять секунд. Не могу представить себе, чтобы кому-то показалось, будто такая функция может быть удобна.

Однако средства управления понемногу совершенствуются. Internet Explorer версии 6 предоставляет больше возможностей типа «решил и за-

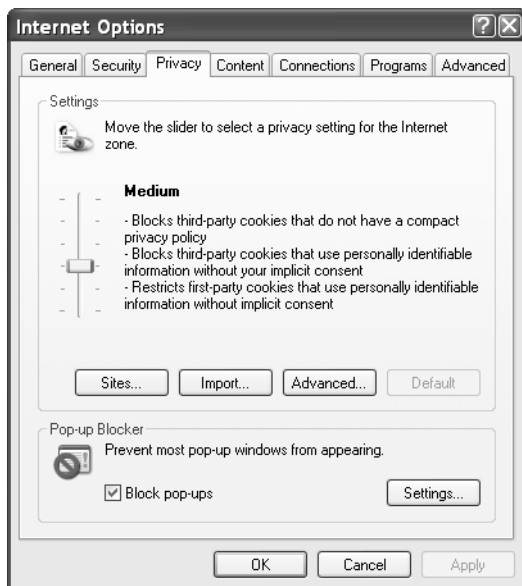


Рис. 5.5. *Настройки конфиденциальности по умолчанию в Internet Explorer 6. Кто-нибудь понимает, что означают эти настройки для cookies? Я — нет*

был», как на рис. 5.5. Одни типы cookies можно принимать, а другие блокировать, но я никогда не мог точно понять, что означает «third party cookies that use personally identifiable information without your implicit consent» (сторонние cookies, использующие личную информацию без неявного согласия пользователя). Браузер этого не объясняет, а тратить силы на поиски в другом месте у меня нет желания. Можно сместить ползунок вверх, чтобы установить более высокий уровень защиты, и вниз, чтобы понизить его. Можно также заблокировать все cookies, а потом добавить исключения — скажем, никаких cookies, кроме Google.com и Amazon.com. Я выбрал смешанную установку. Если щелкнуть по кнопке Advanced (Дополнительно), чего никогда не делают обычные пользователи, появится окно, показанное на рис. 5.6. Я выбираю принятие основных cookies (от сайта, который я фактически вижу) и блокировку сторонних cookies (фокус, применяемый DoubleClick). Пока меня удовлетворяет такая настройка.



Рис. 5.6. Мои пользовательские настройки для cookies в Internet Explorer 6

Чепуха заявлений о неразглашении информации

Каждый сайт, собирающий личные данные, публикует заявление о неразглашении информации (privacy policy). К несчастью, оно обычно содержит столько псевдоюридической трескотни, что найти там нечто полезное бывает почти невозможно. Этой дряни везде полно, значит, она кому-то нужна, хотя мне неизвестно, кому именно.

Работая над этой главой, я зашел на HarryPotter.com, принадлежащий Warner Bros. В их «Заявлении» более 2100 слов — на 400 больше, чем в предисловии к данной книге. И можете мне поверить, оно гораздо менее интересно. Там нет никакого юмора, если не считать слов «защита ваших личных данных очень важна для нас». Грамматический анализатор Microsoft Word оценивает читаемость этого текста как близкую к 12-му классу, т. е. примерно уровню «Wall Street Journal». (Большинство газетных текстов относится к восьмому классу, а данная глава — примерно к девятому.) Если учесть, что серия книг о Гарри Поттере рассчитана на пятиклассников, то двенадцатый класс сложности чтения несколько высоковат, когда нужно, чтобы ваши читатели все поняли.

Начинается с обычной корпоративной пошлости. Насколько я понимаю юридический жаргон, все сводится к следующему: «Мы делаем то, что нам в голову взбредет, а если вам это не нравится, тем хуже для вас». Законов, ограничивающих их права, очень мало — практически никаких в США

и немногим более в остальном мире.¹ Только свободный рынок может оказать сдерживающее влияние. В какой-то мере можно надеяться, что Warner Bros. не сделает с вашей информацией чего-либо ужасного. Скорее всего так и будет. Эта корпорация действует на массовом потребительском рынке. Ее опасение утратить свой тщательно выработанный белый и пушистый образ защищает вашу приватность эффективнее, чем любой закон.

Кроме того, к защите вашей личной информации их заявление о неразглашении не имеет никакого отношения, поскольку в конце этого универсального отказа от ответственности содержится предупреждение о возможности его изменения в любой момент: «Мы можем периодически обновлять это заявление о защите информации. Мы известим вас о существенных изменениях в характере обращения с персональной информацией, поместив сообщение на нашем сайте. Советуем периодически перечитывать данное извещение, чтобы знать, какую информацию мы собираем, как ее используем и кому предоставляем». Перевод: «Мы можем все поменять. Не нравится? Ваши проблемы». Коротко и понятно. Если, что маловероятно, вы дочитаете эту тягомотину до конца, решитесь ли вы когда-нибудь в жизни на это снова? А на десятках, если не сотнях, других сайтов, где вы бываете? Ни за что. Поэтому все может перемениться, а вы даже не узнаете об этом. Вот почему полезны группы наблюдения за охраной персональных данных, которые поднимают тревогу, когда некая компания переходит границы допустимого, и пытаются мобилизовать рынок на борьбу. Мне все же кажется, что им не стоит так часто кричать «волк!». Прочтите, что я пишу о службе Google Gmail ниже в этой главе.

В вопросе изменения политики в отношении личной информации Microsoft поступает гораздо лучше, чем Warner Bros. В заявлении службы аутентификации Passport (см. главу 4) сказано, что «.NET Passport будет запрашивать ваше согласие в отношении любых изменений данного заявления о конфиденциальности личных данных, которые существенно расширяют раскрытие или использование ваших личных данных по каналам, не указанным в данном заявлении на момент получения этих данных». Едва ли можно было рассчитывать на большее. Однако служба Passport проиграла на рынке и имеет весьма ограниченное распространение.

¹ Основное исключение составляет здравоохранение, где действуют такие нечеткие и драконовские законы, как HIPAA (Health Insurance Portability and Accountability Act). Поскольку это особый случай, я не буду его обсуждать.

Как ни странно, у злейшего преступника в мире cookies оказалось одно из лучших в Интернете и наиболее детальных заявлений о конфиденциальности личных данных. В заявлении DoubleClick.com не меньше 30 отдельных подстраниц. Там точно описано, как размещаются сторонние cookie, из чего состоят cookies, типы посещений сайтов, которые в них регистрируются, и как предотвратить сбор данных о вас. Там конкретно объясняется, для чего была сделана попытка соединить регистрацию действий пользователя в сети с базой данных имен и адресов, полученной от Abacus. Дается обещание не делать этого и приводится фрагмент из письма Федеральной комиссии по торговле, в котором говорится о ее доверии к заявлениям DoubleClick о том, что они не занимаются этим. Вам может показаться это недостаточным и вы можете не поверить их заявлениям, но мне не доводилось читать более детальное и точное изложение политики в области защиты личных данных.

А вот заявление о конфиденциальности личных данных для моего информационного бюллетеня по программированию: «Thunderclap не принимает рекламу. Кроме того, мы не продаем, не сдаем в аренду и не раскрываем наш список подписчиков. Мы приложим все усилия к сохранению в тайне имен наших подписчиков; однако не будем пытаться противодействовать судебному предписанию. Если вас это волнует, не подписывайтесь; можете читать бюллетень в режиме онлайн». По-моему, это подходит большинству сайтов. Можно даже изложить суть короче.

Заметая следы

Бывают случаи, когда действительно требуется анонимность в Сети. Классический пример — раскрытие незаконной деятельности правительства. Можно ли после всего нами виденного мечтать о том, чтобы остаться анонимным? Все зависит от того, чьих преследований вы стремитесь избежать и насколько энергично.

Довольно просто посылать анонимную электронную почту через **анонимный почтовый ретранслятор** (anonymous remailer). Он представляет собой сервер, которым обычно управляет какой-нибудь доброволец по идеологическим причинам, принимающий любую почту и пересылающий ее получателю с сокрытием отправителя. Получатель видит, что письмо пришло с анонимного ретранслятора, но не более того. Ретранслятор не ведет никакого учета источников полученных им сообщений (или во вся-

ком случае так утверждает). Очевидно, ответить на такое письмо несколько проблематично, но часто подобные сообщения и не требуют этого: «вот информация — опубликуйте ее». Местонахождение и режим работы анонимных ретрансляторов часто меняются, поэтому мне трудно указать вам хороший сервер. Да и как узнать, можно ли ему доверять? Попробуйте пойти путем общественной библиотеки или Kinko's. Либо запишите свои данные на CD и бросьте его в обычный почтовый ящик.

Если вы не хотите, чтобы веб-сайт знал ваш IP-адрес, спрятаться нелегко. Есть ряд платных продуктов, помогающих замести следы, таких как Anonymizer от Anonymizer Inc. и GhostSurf от Tenebril Inc. Благодаря им, когда вы вводите в окне адреса браузера, скажем CNN.com, ваш запрос посылается не на CNN.com, а на сервер Anonymizer, который запрашивает страницу у CNN.com и возвращает вам результат. Сервер CNN видит запрос не с вашего адреса, а с IP-адреса Anonymizer. Игра может не закончиться на этом, потому что некоторые сайты не принимают запросы от известных серверов анонимности, предполагая, что тот, кто стремится замести свои следы, не станет щелкать по рекламным объявлениям. Насколько надежна эта система? Все зависит от того, можно ли верить, что эта служба сохранит ваши данные в тайне. Против Агентства национальной безопасности? Вероятно, нет. Против вашего начальника на работе? Вероятно, да. (Если, конечно, вы не работаете в АНБ.)

Большинство пользователей не идет на такие хлопоты. Соединение через сервер анонимности обычно действует медленнее, чем напрямую, потому что вместо одного перехода в другую сеть требуются два. Иногда нарушается работа других приложений. Большинству пользователей безразлично, что сайт в состоянии вычислить, что они находятся, скажем, в районе Бостона. Очень часто им это даже на пользу, потому что поиск, скажем, зубного врача, даст того, который расположен поблизости. В Интернете проходит масса обыденной и скучной информации, которая никому не интересна. Найти что-то выдающееся на этом шумовом фоне очень трудно, и для большинства пользователей этого достаточно.

Загадка Google

Компромисс между конфиденциальностью и удобством хорошо иллюстрирует пример популярной почтовой службы Google Gmail (см. gmail.google.com). Мне кажется любопытным, что очень немногие заме-

чают существующую тут дилемму. У всех, с кем я об этом разговаривал, возникала мощная инстинктивная реакция: «Ну, круто! Я тоже хочу, как это сделать?» или «Какой ужас! Да их за это расстрелять мало!».

1 апреля 2004 года компания Google, предоставляющая услуги поиска в Сети, объявила об открытии новой бесплатной почтовой веб-службы под названием Gmail, аналогичной уже существовавшим Hotmail и Yahoo!. Google предложил почтовый ящик гораздо большего размера — 1000 мегабайт (равносильно примерно 700 гибким дискам), в противоположность двум мегабайтам у Hotmail и четырем у Yahoo!. Дата объявления настроила некоторых пользователей скептически, но все оказалось правдой.

Все эти три почтовые службы финансируются с помощью рекламы. Разница — и, если хотите, уловка — в том, что Gmail выбирает рекламные объявления, исходя из содержимого каждого почтового сообщения, которое вы читаете в данный момент. Компьютеры службы просматривают сообщение во время форматирования его для показа и выбирают рекламу по кодовым словам, обнаруженным в сообщении. Идея та же, что и при поиске, когда Google показывает рекламу, исходя из слов, присутствующих в поисковом запросе. Как и при поиске, реклама имеет текстовый вид и скромно располагается сбоку. Пример приведен на рис. 5.7. В письме говорится о сотовых телефонах, поэтому и реклама соответствующая. На рис. 5.8 показано то же сообщение в Yahoo!, где реклама выбирается в соответствии с демографическими данными (вашим возрастом, полом и т. д.) Она состоит из общей рекламы ресторанов в шапке страницы и рекламы кредитных карточек и гороскопов в левой части страницы. Реклама графическая и часто мельтешит перед глазами, что меня весьма раздражает.

Некоторых возмущает, что Gmail просматривает сообщения ради рекламы. Тридцать одна организация, занимающаяся проблемами конфиденциальности, через неделю после представления службы подписала письмо к Google с требованием отказаться от такой практики, утверждая, что «сканирование конфиденциальной почты нарушает подразумеваемое доверие к провайдеру услуг электронной почты». Один член законодательного собрания штата Калифорния (не буду называть ее имени, чтобы не делать рекламы таким занудным ригористам) даже попробовал запретить такую практику законодательным образом, но безуспешно. Статьи со спокойными и непретенциозными названиями вроде «Апокалипсис данных» возвестили о возникновении «нового режима контроля за информа-

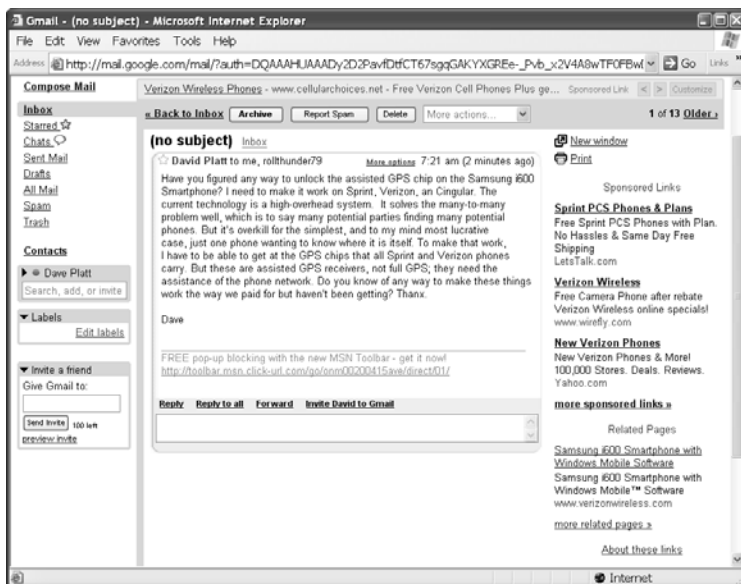


Рис. 5.7. Экран Gmail с контекстно-чувствительной рекламой



Рис. 5.8. Экран почты Yahoo! для того же сообщения

цией».¹ Споров было больше, чем в связи с фторированием воды в моем родном городке в угледобывающем районе Пенсильвании.

С другой стороны, большинство пользователей, не принадлежащих к числу фундаменталистов в области приватности, в беседах со мной не выразили ни малейшего беспокойства по поводу того, что Google читает их почту, чтобы определить, какую рекламу показать. Они даже довольны, что реклама может стать полезнее, поскольку тщательнее отбирается.² Если я пишу кому-то о боли, причиняемой геморроем, и при этом появляется реклама нового лекарства, можно только радоваться, а вот в Hotmail такого не случится. Эти пользователи рассуждают так, что их почту и без того читают фильтры спама, и это очень хорошо, а на работе почту большинства людей просматривает их начальство. Они и не рассчитывают на особую конфиденциальность своей почты, особенно у бесплатных веб-служб. Если когда-нибудь потребуются защитить какое-то конкретное письмо, его можно зашифровать с помощью Pretty Good Privacy или аналогичного средства, но это бывает очень редко. Кроме того, текстовая реклама Google воспринимается ими как менее навязчивая, чем раздражающая графика других служб. Самого себя я отношу к этой группе и заявляю фундаменталистам, которые хотят запретить совершеннолетнему человеку выбирать эту службу: «Послушайте, мы все взрослые люди. Если

¹ Аннали Ньюитц (Annalee Newitz), заметки на Metroactive.com с 5 по 11 мая 2004 года.

² На самом деле контекстная чувствительность Google просто поразительна, по крайней мере, для специалистов. Я знаю пару людей, которые поспорили на ящик хорошего пива, кто из них первый сможет послать через Gmail такое письмо, которое заставит появиться рекламу книги или фильма «Молчание ягнят». Чтобы сделать соревнование более спортивным, они договорились, что в письме нельзя упоминать название, имена главных действующих лиц, снявшихся в фильме актеров и автора книги. (Сами хотите произнести «я тебя умоляю!», или это сделать мне?) Имя победителя — если он определится — я опубликую на сайте этой книги. Я предполагаю, что этот вид спорта распространится быстрее, чем безумие с «домашними камушками» 1970-х, с возникновением сетевых соревнований, коммерческих спонсоров; в конце концов, состоится чемпионат мира, который, разумеется, будет спонсироваться Google. Я придумал для этого спорта название Googleplexing™ и зарегистрировал адрес Googleplexing.com как мировой центр сообщества всех заинтересованных.

нам потребуется, чтобы вы спасли нас от нас самих, мы обязательно к вам обратимся. Кроме того, Google гарантирует возврат денег в трехкратном размере, если вы почувствуете какую-то неудовлетворенность». (Напоминаю, что это бесплатная услуга.)

Конкуренты Google уже реагируют. На Yahoo! размер почтового ящика увеличен до 250 мегабайт. Hotmail продает регистрацию с 2000 мегабайт — вдвое против бесплатного размера Gmail — всего за 20 долларов в год, и без всякой рекламы. Клиентам предоставляется широкий выбор: смотреть специализированные текстовые объявления, случайные графические или заплатить и не смотреть никакой рекламы. Когда выбор большой, это очень хорошо. В ответ Google увеличил размер почтового ящика с 1000 мегабайт до 2500. Свободный рынок, как он всегда это делает, решит, будет ли успех у Gmail. Мне кажется, что будет, потому что пользователи в большинстве своем доверяют Google. Не думаю, что такой план прошел бы у Microsoft.

Решение

Конфиденциальность личных данных — штука забавная. Как видно из этой главы, каждый понимает ее по-своему. И не всегда причиной проблем с конфиденциальностью служит скверно написанное программное обеспечение.

Решение частично находится в области законодательства. Новые технологические возможности требуют новых типов правового регулирования. Закон, запрещающий ведение разговоров по сотовому телефону во время движения не требовался, пока не существовало сотовых телефонов. По мере роста мощи компьютеров и пропускной способности линий связи потребуется законодательство, ограничивающее возможности злоупотребления ими. Например, Закон о защите конфиденциальности в видеопрокате (Video Privacy Protection Act, 18 U.S.C. § 2710), объявляющий преступлением раскрытие информации о том, какие видеозаписи брал клиент, был принят в ответ на публикацию списка видеозаписей, взятых Робертом Борком, во время обсуждения его кандидатуры на пост члена Верховного суда (куда он не прошел). Неплохо для начала объявить преступлением просмотр налоговых или медицинских записей лица без его согласия. Однако нельзя полагаться на сознательные действия пользователей. Они не знают, что и где искать, и уже запутались в технологиях.

В Европейском союзе более строгие законы о конфиденциальности, чем в США, обычно требующие явного выражения согласия на сбор и использование данных, но флажки, указывающие на решение пользователя, часто выставлены по умолчанию и запрятаны в глубине лицензионных соглашений, где их никто не видит.

Частично проблема решается техническими средствами. Улучшить защиту своих данных можно с помощью серверов анонимности и блокираторов рекламы. Я уверен, что такие программы будут совершенствоваться. Они не дороги, но иногда возникают проблемы их совместимости с другими программами. Вот я сейчас пишу эти строки, а мой блокиратор рекламы сражается с антивирусной программой. Прекратите, или я накажу вас обоих! Благодаря этим инструментам у вас будет хоть немного больше конфиденциальности, чем у вашего соседа, потому что большинство людей палец о палец не ударят с этой целью.

В какой-то мере помогает простая бдительность. Как правило, нетрудно отказаться от сбора любой информации о вас. Большинство продавцов не станет с вами возиться, если вы против. Слежение за вами — это дополнительные расходы для них, и еще большие расходы, если вы создадите им проблемы.

Существующая в данное время картина в области защиты личных данных в сети отражает капиталистический свободный рынок, как это происходит почти везде. Есть компании, которые изучают рынки и тщательно собирают все данные в надежде, что какая-то неожиданная их комбинация позволит извлечь прибыль. Есть группы наблюдения, готовые поднять вопль при всяком возникновении действительной или мнимой угрозы конфиденциальности. И в середине между ними находятся широкие народные массы — «молчаливое большинство», по Ричарду Никсону, пытающиеся делать свою работу, воспитывать детей и иногда отдыхать, не тратя все свое свободное время и силы на поиск всяких гадостей в этих cookies. Если они вообще задумываются о таких проблемах, то ищут средство быстро решить их: купил программу, установил ее и больше ни о чем не волнуйся. Если они чувствуют серьезную угрозу, то поднимают большой шум, как это было в 1998 году, когда общенациональная сеть аптек CVS разослала своим клиентам рекламные письма, основанные на их медицинских записях. И компании, действующие на широких рынках, отступают при первых же признаках организованного сопротивления.

На мой взгляд, лучший способ ставить палки в колеса охотникам за чужими секретами, оказывается также простейшим и самым приятным. Разумеется, я имею в виду преднамеренное введение в заблуждение. Не отказывайтесь предоставить им данные. Но дайте им скверные данные. Пусть отношение сигнал/шум в собираемых ими данных упадет так низко, что из них нельзя будет извлечь никакой прибыли. Например, Babies «R» Us потребовала мой почтовый код, когда я на днях, будучи в Массачусетсе, покупал пеленки. Я не стал отказываться, а придумал код для Калифорнии. Махнитесь карточками покупателя со своим соседом, и они в магазине увидят, как человек покупает кошачий корм, рассаду люцерны и пеленки в один день, и корм для собак, сигареты и памперсы для взрослых — через неделю. Приятно помечтать, как опросы Гэллапа/Зогби на выходе избирательных участков для президентских выборов дадут 100-процентный ответ «уйди с глаз моих, пока я не раскроил тебе череп», но еще лучше направить их по ложному пути и посмотреть, как глупо они будут выглядеть. Когда мне звонят при проведении опросов на политические темы, я не вешаю трубку, а, не щадя своего времени, стараюсь их поморочить. Я вру намеренно и со злобным умыслом. И получаю при этом огромное удовольствие. Мне всегда интересно, не смущает ли маркетолога, записывающего мои ответы, то, что я фыркаю от смеха. Приглашаю вас присоединиться ко мне. Всего лишь несколько процентов, и вечером после очередных выборов нас будет ждать лучший пример политического идиотизма с тех пор, как Генри Киссинджер получил Нобелевскую премию мира.

Этими словами я собирался завершить данную главу. Но читатель, комментировавший один из ранних ее вариантов, написал мне следующее: «Меня мало интересуют всякие cookies и прочие штуки. Я не вижу, какой ущерб они могут мне нанести. Чего я действительно боюсь, так это того, что в один прекрасный день мои личные данные украдут и мой банковский счет окажется пуст». Прочтя эти слова, я понял, что у данного читателя приоритеты четко определены, в отличие от некоторых участников дебатов по поводу приватности. Поэтому теперь, когда вы знаете, в чем заключаются проблемы приватности, вы достаточно информированы, чтобы понять, насколько менее существенны они в сравнении с проблемами безопасности. Перечитайте еще раз главу о безопасности и сделайте то, что в ней рекомендовано. И только после этого вы можете позволить себе беспокойство по поводу того, что обувному магазину может стать известно о вашем недавнем посещении туристического агентства.

ГЛАВА ШЕСТАЯ

Десять тысяч гиков, помешавшихся на джолт-коле

Можно много узнать о животном, изучая его в его собственной среде обитания. Если вас интересуют повадки компьютерщиков, изготовивших программы, которыми вы пользуетесь (но которые вам не нравятся), читайте дальше. Я приведу вас на одно из их главных сборищ и покажу, как они себя ведут, когда думают, что их никто не видит.

Посмотрите на них в их естественной среде обитания

За год происходит великое множество компьютерных конференций по самым различным темам. Если бы я посещал их одну за другой и на каждой выступал, у меня не осталось бы времени ни на что другое (правда, я немного заработал бы при этом и вскоре скончался бы от истощения). Самые маленькие собирают человек по 20, но я возьму вас собой на самый крупный форум, на Tech Ed, проводимый под эгидой Microsoft. Там-то мы и увидим величайшее разнообразие поведения компьютерщиков и величайшее единодушие там, где нет различий.

Tech Ed — ежегодная серия конференций, обычно начинающаяся в конце мая или начале июня и продолжающаяся все лето. Я выступаю там с 1999 года, и эти мероприятия мне действительно нравятся. Одна конференция происходит в Соединенных Штатах, другая — в Европе и еще несколько — в азиатско-тихоокеанском районе. В североамериканской части набирается около 10 000 участников, в связи с чем это мероприятие случается только в городах, где есть очень крупные конференц-центры.

Даллас. Лас-Вегас. Новый Орлеан. Орlando. В близлежащем ко мне Бостоне только что построили новый центр для проведения конференций, и я мечтаю о том, чтобы спать в собственной постели, когда этот съезд состоится там.

Все эти гики

Поместить 10 000 компьютерщиков в одном здании и обслуживать их в течение недели так, чтобы все остались живы, — задача материально-технического обеспечения не из легких. Вы можете подумать, что самую большую проблему составят входящие и выходящие потоки, т. е. питание и туалеты, но в центрах для проведения конференций она обычно решена. В данном случае наибольшие трудности доставило подключение к Интернету. Интернет необходим для электронной почты, для мессенджера, для поиска ответов на возникающие вопросы. Компьютерщик без Интернета — что наркоман, не получивший привычной дозы. Он дергается, потеет, лезет на стену и бредит.

Проблема работы в сети решается на Tech Ed с помощью более чем 1000 компьютеров, устанавливаемых в зале, которыми могут пользоваться делегаты. Зрелище компьютеров, установленных рядами, как могильные плиты на мемориальном кладбище, при первом взгляде поражает (рис. 6.1). Участники устремляются сюда во время каждого перерыва, проносясь мимо столов, на которых возвышаются горы пончиков. Возможно, будучи компьютерщиками, они предполагают, что от гор такой высоты наверняка что-то останется к тому моменту, когда они разберутся со своей почтой, и обычно оказываются правы. Хоть это и противно здравому смыслу, но кучка поменьше могла бы заставить их остановиться.

Это компьютеры, подключенные по проводам, но эфир над конференцией трещит от беспроводных устройств связи. Все здание охвачено беспроводным Ethernet для ноутбуков — таким же, как в вашем местном Starbucks, только в тысячу раз больше. Гики притаскивают с собой свои игрушки, чтобы похвастаться ими перед другими гиками и для работы. Высокоскоростные сотовые модемы для ноутбуков на случай отсутствия точки доступа к беспроводному Internet. Наладонники со встроенными сотовыми телефонами. Сотовые телефоны со встроенными компьютерами. Сотовые телефоны и наладонные компьютеры с беспроводными наушниками, подключаемыми через Bluetooth. Я все жду, когда кто-нибудь



Рис. 6.1. Компьютерный зал Tech Ed PC, пустой

достанет телефон из ботинка, как агент Максвелл Сمارт из старого телесериала *Get Smart*.

У каждого на шее болтается дурацкая прищепка для таблички с именем и названием фирмы. Очень помогает начать разговор или драку: «Vorland? Кажется, была такая софтверная компания, или я что-то путаю?»¹ Я завязал беседу с одной из немногих женщин-делегатов, прочитав на ее значке, что она работает в исследовательском институте Аквариума залива Мон-

¹ Компания Vorland была крупным игроком в начале эры ПК, одной из восходящих звезд 80-х и начала 90-х. Ее дешевый (\$50) и отличного качества (по меркам того времени) Turbo Pascal привел в программирование многих новичков. 30-дневная гарантия возврата денег за товар и отсутствие защиты от копирования побудили других поставщиков предлагать такие же условия. К сожалению, ряд ошибок, описанием которых я не стану вас утомлять, привел компанию к утрате лидирующего положения, значительной части работников и рынка. Теперь они никого не интересуют в этой отрасли и, как многие очутившиеся в аналогичном положении, очень болезненно его воспринимают. Думаю, даже это короткое сообщение вызовет их критику. Однако настоящие гики, к числу которых я отношу и себя, сожалеют о прежней Vorland.

терей — любимого места посещения моей дочери. Я предложил прочесть у них лекцию, когда буду там в следующий раз. Тем самым я окуплю поездку, и, может быть, они покажут моим дочкам выдру.

В отличие от, скажем, съезда хирургов, на Tech Ed все ходят в футболках, и они тоже очень помогают возникновению контактов. Я ехал на лифте с малым в футболке от Live365.com — веб-сайта, на котором каждый может составить музыкальную программу и законным образом передать ее слушателям. Я поинтересовался, как это работает, поскольку мне понравились некоторые программы «пляжной музыки», и пока мы поднимались, он мне все объяснил. Со случайным прохожим на улице такого быть не может, но здесь, на конференции, это в порядке вещей. Как в церкви (а Tech Ed и есть своего рода Церковь Пресвятой Богородицы во имя Гикерства), для знакомства достаточно быть под одной крышей.

Помимо особенностей в одежде и манерах общения у гиков есть специфические пристрастия в еде и питье. Количество потребляемого ими кофеина уже вошло в легенду. Я замечаю, что захожу в режим конференции, если в гостинице передвигаю поближе к кровати кофеварку, чтобы утром запускать ее, не вылезая из постели. Перерывы между выступлениями — это жестянки с кофе, очереди к кофеваркам и лотки с любимыми напитками гиков «Горная роса» и «Джолт-кола» (с сахаром и двойным кофеином!). На Tech Ed я впервые увидел кофеинизированную воду. На столах громоздятся закуски, преимущественно соленого/сладкого/жирного типа. В Калифорнии подносы с морковкой и сельдереем остаются нетронутыми, зато делегаты (и докладчики тоже) набрасываются на пирожные. Один из выступающих предупредил: «Ребята, полегче с тяжелой едой. Конференция длинная, а сегодня еще только вторник». Он был прав: гипогликемия развилась у меня примерно к полудню среды. Компьютерные гики в большинстве своем молоды, поскольку это не такая древняя профессия, как, скажем, плотницкое дело. Как менялись с годами подарки, которыми мы обменивались на встречах выпускников колледжа — пивные кружки → стаканы → коньячные рюмки → кофейные чашки с 12 заповедями анонимных алкоголиков, — так и характер предлагаемой на Tech Ed еды будет меняться по мере старения аудитории. Когда они откажутся от мороженого Naagen-Dazs, я перестану туда ездить.

Кто говорит, когда и о чем

Самое интересное на Tech Ed — это, конечно, презентации: более 500 обсуждений в течение недели всех мыслимых тем, относящихся к разработке программного обеспечения в мире Microsoft. Как сделать это. Как избежать того. Конкретные примеры того, что произошло, когда люди пытались сделать это и одновременно избежать того. Новейшие усовершенствования для любых продуктов Microsoft. Новые стили программной архитектуры, обусловленные этими последними нововведениями. Обсуждения в комиссиях с участием крупных знатоков вроде меня. Игры на xbox с проекцией на экраны во всю стену.

В последнее время наибольшее внимание привлекают обсуждения вопросов безопасности, которые сейчас волнуют всех. Стив Райли (Steve Riley), чьи доклады всегда пользуются повышенным вниманием, рассказал о психологических методах атаки. В главе 3 я рассказывал о психологических атаках («социальной инженерии») как способах воздействия на людей с целью компрометации систем защиты, например позвонить в службу поддержки и узнать у них пароль, вместо того чтобы вскрывать его математическими методами. Стив — замечательный рассказчик, который всегда получает высшие оценки. Его выступление не оставило у слушателей сомнений в реальном существовании серьезной проблемы, но жаль, что он не остановился подробнее на компромиссах, возникающих при ее решении. Как определить такую политику безопасности, которая обеспечит достаточную степень защиты и в то же время не станет чрезмерно обременительной?

Конечно, такие попытки могут иметь негативный результат — слушатели перестанут вас понимать. Помню, однажды Эрик Олсон (Erik Olson), главный эксперт по безопасности веб-сервера Microsoft, провел отличную презентацию. Он подробнейшим образом рассказал, что надо сделать, чтобы защитить пользовательские идентификаторы и пароли на своем сервере, какие существуют компромиссы, и когда один оказывается предпочтительнее другого, и какие уязвимости остаются в любом случае. Я сидел рядом с разработчиками из некоего районного полицейского управления, озабоченно кивавшими головами, размышляя, как все это сделать, ничего не поломав, и какие еще остались проблемы, о которых Эрик не успел рассказать за отведенный ему час.

На Tech Ed всегда бывает несколько основных обсуждений с участием таких тяжеловесов, как Билл Гейтс и Стив Балмер. На них собирается много публики, но я практически перестал туда ходить. Слишком общо: недостаточно конкретных знаний. Например, в 2004 году в Сан-Диего Стив Балмер заявил, что Microsoft увидела ошибочность своего пути и прекращает добавлять в свои программы новые функции, пока не сделает их безопасными. Это замечательно, но пока я не увижу конкретно, как Microsoft повышает безопасность того, с чем я работаю, от этих замечательных устремлений нет никакого толка.

Tech Ed — это продукт Microsoft, поэтому неудивительно, что картина мира там демонстрируется такой, какой видит ее Microsoft. Если вы работаете преимущественно в мире, ориентированном на Microsoft, как многие из нас, то знать ее полезно. Но нужно помнить о существовании обратной стороны медали, которой вы не найдете на Tech Ed. Для этого придется поехать на другую конференцию, например на Java One, и там вы услышите дискуссии на темы типа «Microsoft: угроза или опасность?»

Докладчики борются за высокие оценки. Участники вводят их в режиме онлайн с почтовых компьютеров, которые вы уже видели на рис. 6.1. Итоги подводятся мгновенно и отображаются на мониторах, расположенных повсюду. Оценки указываются в баллах и могут снабжаться комментариями. «Слишком много на эту тему. Слишком мало на ту. Нужно больше примеров. Дежурная шутка была истеричной. Шутка — отстой. Ширинка была расстегнута. Купи Макинтош».

Докладчики постоянно следят за своими оценками и сравнивают их с оценками коллег. Моим лучшим результатом было четвертое место за неделю. На сайте показаны десять лучших оценок и, целесообразность чего спорна, десять худших. В Сан-Диего в нижний ряд попал человек, с которым я был знаком, хотя мы и не работали вместе. Он получил немногим более трех баллов из девяти — результат всяко не блестящий. Я не присутствовал на его выступлении, но комментарии слушателей были примерно такого содержания: «Он появился на полчаса позже, чем должен был, явно с похмелья». (В глазах выступающих это крупное преступление, тогда как способность хорошо выступить после перепоя, особенно заметного, ценится высоко.) «Его иллюстрации никуда не годились». Не знаю, пригласят ли его еще раз.

Что мне больше всего нравится на Tech Ed, так это возможность встретиться с коллегами-докладчиками, лидерами индустрии, «дигерати».

Обычно я работаю в одиночестве или с клиентами, уровень компетенции которых гораздо ниже, чем мой. (Я никоим образом не хочу их унижить: будь это не так, стали бы они мне платить?) Не то чтобы мне нечему у них научиться — я делаю это постоянно. Но на целую неделю погрузиться в море мыслей этих выступающих — вот моя главная услуга. Меня должны понять музыканты, которым на музыкальном фестивале случается выступить в джем-сейшене с другими хорошими музыкантами.

Мы говорим о том, чем мы занимаемся и что пытаемся сделать, над чем по-прежнему ломаем головы, вспоминаем ту или иную из прежних конференций. «Слушай, здорово мы им вломили в Барселоне прошлым летом? О чем твоя следующая книга? Какие отношения у тебя с издателем?» Я встречаюсь с живыми людьми, которых раньше знал только по их работе — так же как и они меня. Я люблю, когда цитируют какие-нибудь из моих книг: «Слушай, Дэйв, твой Второй фундаментальный закон мироздания гениален: «количество мерзости во вселенной постоянно; если в одном месте ее стало меньше, значит, в другом прибавилось, потому что избавиться от нее невозможно». К моему проекту это имеет прямое отношение».

На европейской Tech Ed вечером в день закрытия конференции докладчики собираются на ужин, для которого организаторы арендуют здание старой пивоварни Heineken в Амстердаме. Обсуждения закончены, участники разъехались по домам и можно снять маски. Можно расслабиться и насладиться общением, поскольку редко выдается возможность встретиться с таким количеством умных людей. Один из докладчиков, человек, которому я симпатизирую и которому нравятся мои книги, говорит, обращая ко мне: «Платт, это потрясающе». Размахивая своей кружкой и проливая пиво (я пришел позже него), он продолжает: «Посмотри на этих умниц. Вот [один толковый человек], а вот [еще один толковый человек], а там [еще один чрезвычайно умный человек]. И я заслужил свое место за одним столом с ними, так же как и ты. Какой кайф!» Полностью с ним согласен.

Я не хочу утверждать, что докладчики на Tech Ed безумствуют неистовее, чем участники прочих съездов, например страховые агенты, руководители похоронных бюро или продавцы оборудования для колостомии, — и уж явно далеко нам до Братьев Храма. Но очень интересно видеть, как все эти серьезные и рациональные люди ведут себя столь раскованно, хотя тут есть опасности. Если видеозапись, на которой я распеваю караоке или,

точнее, *пытаюсь* петь караоке («In the jungle, the mighty jungle, the lion sleeps tonight»), попадет в Интернет, моей репутации конец.¹

Замечательно наблюдать, как новый докладчик вступает в наше братство. Одна дама, которая провела свою первую лекцию днем в пятницу, все еще тряслась в лихорадочном возбуждении, когда нас в Барселоне в 3 часа утра выставляли из закрывающегося бара: «Ребята, это было здорово, так всегда бывает?» — «Алексис, — сказал я ей, — тебя укусил вервольф. Теперь ты — одна из нас и обречена вечно скитаться по воздушным дорогам мира, и поддерживать тебя будет только слабая надежда на переход в первый класс, и ты всегда будешь гнаться за «кайфом докладчика», который можно испытать только перед лицом аудитории. Добро пожаловать, и пусть судьба будет к тебе благосклонна. Кроме того, ты можешь поставить всем выпивку. Кажется, вон то заведение еще открыто.»

Конечно, такое могущество всегда обоюдоостро. Я помню, как однажды на конференции, аналогичной Tech Ed, собирался посетить лекцию, а ее отменили. Я поинтересовался причинами у организатора... «Эта лекция была бы второй для женщины, которая ее готовила, но оценки ее первой лекции оказались такими убийственными, что выходить второй раз она отказалась», — ответил мне он. Ее разнесли в пух и прах, причем целая группа участников обрушилась на нее со злобными личными нападками, для которых шкура новоиспеченного лектора слишком тонка. И это случилось в Канаде, где люди, по общему мнению (и моему опыту), весьма доброжелательны, т. е. она явно стала причиной скандала. Может быть, оно и к лучшему, что я пропустил эту лекцию. Как альпинистам щекочет нервы их вид спорта, потому что только благодаря собственному мастерству им удастся избежать падения, так и мы, лекторы, нервно ожидаем успеха лекции, потому что много раз видели, как с треском проваливаются наши коллеги, сами несколько раз попадали в такую ситуацию и знаем, что может произойти, когда беремся не за свое дело, или плохо готовимся, или относимся к аудитории с недостаточным уважением — не тем, которого заслуживает бочка живых гремучих змей.

¹ Когда на следующий день я возвращался с Tech Ed, то все еще насвистывал эту мелодию, делая пересадку в лондонском аэропорту. Кто-то, кого я не разглядел, крикнул мне с другого конца терминала: «Платт, черт тебя возьми, ты когда-нибудь замолчишь?» Не волнуйтесь: с работы меня пока не уволили.

Мы, лекторы, в чем-то, конечно, конкурируем друг с другом, но больше сотрудничаем. Один из таких лекторов познакомил меня с издателем данной книги. Мы зарабатываем себе на жизнь довольно редким способом, и мне кажется, что при наших встречах мы выказываем лучшее, что у нас есть. Несколько дней мы проводим бок о бок друг с другом, а потом наши пути снова расходятся до новых встреч где-нибудь в Редмонде, или Амстердаме, или Сингапуре.

Это ярмарка

Помимо всех лекций Tech Ed представляет собой крупную торговую ярмарку. Участники Tech Ed прибывают туда в расчете увидеть новейшие достижения в индустрии. На лекциях они учатся писать программы, а на коммерческих просмотрах видят продукты и услуги, реально существующие на сегодняшний день, в том числе инструменты, которыми они могут пользоваться для работы.

Выставочный зал по размерам не меньше футбольного поля, и наличие подходящего пространства служит, наверное, важнейшим фактором, определяющим выбор места проведения конференции. Участие в выставке стоит дорого — от 7000 долларов за самый маленький стенд, но разве могут поставщики этого рынка позволить себе упустить случай показать свой товар 10 тысячам гиков, собравшимся в одном месте? Тут есть все: программы, компьютеры, наладонные компьютеры и программируемые сотовые телефоны, книги и журналы, обучение (и через сеть, и лично). Здесь есть все, в чем есть биты.

Как написал в одном сетевом издании о ярмарке Tech Ed Эрик Синк (Eric Sink), основатель фирмы, производящей инструменты для программирования, «...это время, проводимое лицом к лицу с клиентом. Побывав на этом шоу, мы можем лично встретиться с потенциальными клиентами. Они выскажут нам свое мнение о нашем продукте. Они попросят включить в него функции, которых там нет. Нам представится редкая возможность пообщаться с участниками нашего сегмента рынка как с реальными людьми. Мы также сможем лично пообщаться со своими конкурентами и партнерами... и клиентами. Целую неделю пользователи наших продуктов будут останавливаться у стенда, чтобы поболтать. Одним будет просто интересно увидеть нас живьем. Другие расскажут, как они довольны

нашим продуктом. Третьи сообщают, что им не понравилось в нашем продукте». Как видите, у участников рынка будет напряженная неделя.

Коммерсанты всеми доступными средствами стараются заманить проходящую мимо публику на свои стенды. Вспомните своих знакомых гиков. Конечно, они все ненормальные, но, как вам известно, они безумно любят всякие игрушки и забавы. Поэтому витрина должна обязательно привлекать внимание. Народ останавливался у гоночной машины «Формулы 1», рекламировавшей программное обеспечение для сбора телеметрических данных во время гонки. На стенде графической компании был выставлен трехметровый «волшебный экран», к которому выстроилась очередь желающих порисовать. Большим успехом пользуются фокусники. Ни один гик не может справиться с искушением разобраться, на чем основан трюк, особенно если его выполняют у тебя на глазах. А из-за пристрастия гиков к кофеину у них имеют бешеную популярность тележки с бесплатным эспрессо. Если вы думаете, что в такой преимущественно мужской аудитории большое внимание обращают на красивых девушек, то ошибаетесь. Думаю, что дело здесь не в политкорректности, а в том, что с самого начала конференции мозговая активность гиков переключается на тот раздел, который заведует битами, и до самого конца ни о чем другом они уже думать не могут.¹ Один читатель сообщил мне, что некий крупный поставщик однажды пригласил танцовщиц из Dallas Cowboy Cheerleaders, но не смог вспомнить ни этой фирмы, ни названия продукта.

Продавцы также наперебой бесплатно раздают товары — обычно дешевый хлам — в обмен на контактную информацию. Делегаты не нищенствуют, зарабатывая не менее 50 000 в год, а часто вдвое больше. По каким-то непонятным причинам эти безделушки, «цацки», оказываются для них настолько привлекательны, что каждому участнику при регистрации выдают большой рюкзак, в котором можно увезти это добро. Брелки для ключей. Фонарики. Бейсбольные кепки. Йо-йо. Светящиеся авторучки. Резиновые мячики, которые можно стискивать на совещаниях, представляя себе, что это горло твоего начальника. Раньше были популярны

¹ Очень хорошей иллюстрацией служит старый анекдот о программисте, который решил завести любовницу, чтобы повысить эффективность своей работы. На вопрос о том, как это поможет ему писать код, он ответил: «Очень просто. Любовнице я скажу, что у жены, жене — что у любовницы, а сам тогда действительно смогу поработать!»

складные ножички (типично гиковская штучка), но теперь они вышли из моды, поскольку их запретили проносить в самолет. Компания, демонстрировавшая большой волшебный экран, раздавала в качестве призов маленькие экраны. Всегда популярны футболки: поскольку в большинстве фирм они приняты в качестве официальной рабочей одежды, можно целый год ничего не покупать, если стащить их в достаточном количестве.¹ Одна компания решила, что «к черту эти глупости с безделушками», и перешла сразу к делу, выдавая настоящие двухдолларовые банкноты всем, кто досидит до конца их рекламного показа. Сам я этого не видел, но говорят, что очередь была через весь зал.

Если вы решили задержаться у стенда, то выберите себе правильного собеседника. Первым всегда в глаза бросается какой-нибудь маркеттоид,² извергающий «умные» слова, которых не понимает, в расчете подцепить покупателя, который лучше купит продукт, чем признается, что тоже не понимает этих слов. Помню, как один такой малый восторгался неким программным продуктом, последняя версия которого только что была сделана «объектно-ориентированной». Не буду вас мучить определением этого «умного слова», скажу лишь, что оно описывает внутреннюю структуру программы, но не то, о чем следует беспокоиться пользователю, и не то, что он вообще может заметить. К тому времени маркеттоиды меня уже несколько утомили, и я прекратил его поток красноречия, сказав: «Простите мне мое невежество, но что такое объектно-ориентированное программное обеспечение, в чем его отличие от необъектно-ориентированного, и почему это отличие должно интересовать меня как покупателя?» Видеть, как он корчится, словно проткнутое булавкой насекомое, пока не появился технический специалист, который узнал меня и выручил его из беды, и мы оба весело не посмеялись, стало для меня самым ярким событием выставки в тот год.

¹ Я однажды обыскал весь выставочный зал в поисках рекламных носков, потому что у меня кончились чистые, но безуспешно. Кидаю рекламную идею — кто откажется от пары красивых удобных носков?

² Специально для этой цели я выдумал слово *marketingbozo*. Может быть, не очень выразительное, но полезное. Я употребляю его для той разновидности безмозглых болванов, которые тратят целое состояние на рекламу декофеинизированной диетической Джолт-колы и потом не могут понять, почему она плохо продается.

Важнейшим признаком хорошего экспоната служит интерактивность. Посетители должны иметь возможность попробовать товар в работе. Одно дело прочесть описание шлема с трехмерной интерактивной графикой. Совсем другое надеть его на себя и покрутиться, наблюдая за перемещениями трехмерных тел, синхронными с поворотами моей головы. Под впечатлением от увиденного я разговорился с производителем о технологиях, которые придется освоить программистам, чтобы писать приложения для такого устройства, и дал ему несколько советов по поводу того, как это сделать проще. Эти шлемы пока дороговаты — 5000 долларов за штуку, но при массовом производстве цена может упасть до 1000 долларов за пару лет (если компания доживет до того времени), и тогда они станут высокотехнологичным рождественским подарком. Производитель демонстрировал шлем вместе с фильмом, показывающим геометрические тела, и был явно нацелен на рынок игр. Но когда я поинтересовался у участников конференции, попробовавших этот шлем, какой контент им было бы интереснее всего увидеть в этом устройстве, ответ был однозначен: порно. На мой вопрос к производителю, планируют ли они выход на этот рынок, ответ был отрицательным. «Стало быть, если Лонг Донг Сильвер выпишет вам чек, вы у него денег не возьмете?» — «Не возьмем». Бьюсь об заклад, что венчурные капиталисты, которые вложились в него, дали бы другой ответ. Мне не хотелось говорить, что он лжет, но либо это так, либо он глуп и долго на этом месте не удержится.

Продавцы дарят брелки для ключей или пластиковые пружинки-слин-ки¹ всем, кто останавливается у стенда, но для отдельных друзей у них есть особое угощение. О вечеринках на Tech Ed ходят легенды, и количество полученных на них приглашений свидетельствует о вашем статусе, даже если вы не собираетесь присутствовать на всех. Закрытые ужины в избранных ресторанах. Водные путешествия с выпивкой. Барбекю на природе с механическим быком. Какая-то попойка в зоопарке Сан-Диего, на которую я не смог заполучить приглашение, но, судя по рассказам, потрясающая. Один из докладчиков Microsoft, в обязанности которого входит посещение выставок примерно раз в месяц, сказал, что на эти мероприятия нужно брать с собой запасную печень.

¹ Пластиковая пружинка из множества разноцветных витков.

Гики, поколение «Next» — передача эстафеты

Из того, чем мне приходилось заниматься на Tech Ed, самым интересным и, пожалуй, беспокойным делом было судейство в конкурсе программистов Imagine Cup летом 2003 года в Барселоне. Это соревнование команд студентов, которые изучают информатику, финансово поддерживается Microsoft. Сами студенты выбирают задачу, а потом предлагают ее программное решение с помощью новейших технологий Microsoft. Помимо развлекательной и познавательной сторон, 16 финалистов награждаются бесплатной поездкой на Tech Ed, а три лучшие команды делят также 40 000 долларов призовых денег. Все команды сохраняют право собственности на свои изобретения.

Мы, восемь судей, трое из США, а остальные из различных университетов со всего света, должны были оценить участников соревнований. Шестнадцать команд вышли в этот тур — две из США, остальные из Европы, Азии и Южной Америки. Все они продемонстрировали мастерство программирования и преодолели технические препятствия с изобретательностью и воображением. Но меня по сей день коробит от их неумения описать задачу, от того, как они выбирали проблемы, с которыми собирались справиться. Почти все они испытывали преклонение перед технологией как таковой, противником чего я, как вы, наверное, заметили, являюсь. Я думаю, что их академические наставники, допускающие это, наносят им большой вред. Страшно подумать, что с таким настроем они выйдут в мир.

Например, немецкая команда (единственная, участники которой надели блейзеры и галстуки) пыталась решить проблему неконтролируемого размножения паролей — необходимости иметь различные ID пользователей и пароли для многочисленных учетных записей в Сети, о чем я писал в главе 4. Да, это реальная и трудная задача, но их подход к ее решению показал, что они не поняли, в чем ее трудность. Для доступа к коммерческому сайту надо было не вводить пароль, а позвонить с сотового телефона на сервер этой команды, чтобы сайт аутентифицировал вас при помощи системы определения номера звонящего. Этот же подход с единой регистрацией я описал в главе 4, но задействующий аппаратное устройство, которое, как они рассчитывали, уже должно быть у покупателя.

Они не поставили себя на место своих клиентов — ни пользователей, ни сетевых торговцев. Допустим, что вы хотите купить книгу на Amazon.com.

Что вы предпочтете — ввести, как обычно, пароль или вспомнить, куда вы положили этот чертов телефон, добраться до него, зарядить аккумулятор (еще нужно найти зарядное устройство), включить, дождаться конца начальной загрузки, набрать 10-значный номер, потом ввести другой 12-значный номер заказа? Когда я спросил у них, почему пользователь захочет делать покупки на сайте, с которым такие хлопоты, они ответили: «Потому что он безопаснее». Они не понимают, что главная проблема любой системы защиты — это легкость использования. Если клиенты решат, что бронированная дверь открывается слишком тяжело, то оставят ее открытой и отключат датчик сигнализации, либо найдут обходной путь, где не заперто, либо вообще пойдут в другое место. Команда не подумала и о том, как решить парадокс курицы и яйца, характерный для их конструкции: многие ли сайты станут платить за службу аутентификации, если в ней не зарегистрировано достаточное количество пользователей, и многие ли клиенты станут регистрироваться, если нет полезных сайтов, действующих эту систему? Это классический пример, когда изобретатель не задумывается, будет ли его изобретение полезным. Чего еще можно было ожидать от страны, язык которой подарил нам такое гибридное слово, как *schadenfreude*.¹

Команда из Тайваня разработала систему, которая идентифицировала музыкальные мелодии или, по крайней мере, пыталась. Вам случалось проснуться утром, чтобы при этом в голове у вас звучала мелодия, и вы не могли вспомнить ее название? Мне тоже не случалось. Но эта команда написала такую программу, что можно набрать телефонный номер и напеть мелодию компьютеру на другом конце провода, после чего программа определит, что это за песня, сыграет ее вам и предложит загрузить в качестве ринг-тона на ваш телефон, если вы введете номер своей кредитной карточки. У участницы команды — милой девушки с приятным голосом — это получалось, а у меня — нет. Может быть, мое хрипение из-за расстроенных перелетом суточных ритмов сбило с толку устройство или в его базе данных не нашлось мелодии «Three Day Drunk...» (Пью третий день...). (Fish Head Music, 1999, www.jim-morris.com. Неизвестные мелодии трудно запоминаются по определению. Наверняка у них нет

¹ Оно означает «радость при виде чужого несчастья». Узнав, что О. Дж. Симпсон проиграл дело и должен отдать Голдменс все свои накопления, я сплясал *schadenfreude*-джигу. Такое замечательное слово; жаль, что не я его придумал.

и его хита 2005 года «Booze Is the Duct Tape of Life» (А без пьянки жизнь плохая).) И здесь тоже технология как таковая показалась мне любопытной, особенно алгоритмы обработки сигналов (я же все-таки гик). Но не стану врать — я не очень боюсь проснуться с неизвестной мне музыкой в ушах.

Хуже всех была, по-моему, команда из Сингапура. Они утверждали, что в сингапурском супермаркете можно больше 20 минут простоять в очереди в кассу. Не знаю, как такое может случиться в суперэффективном Сингапуре, но им виднее, и я согласен, что ликвидация 20-минутных очередей в кассу стала бы огромным благом для человечества. Они проделали огромную работу, наверно большую, чем другие команды. Они написали прекрасные программы и даже разработали вместе с инженерами-электронщиками специальную аппаратуру, которая им понадобилась. И создали уродливого монстра, каких нужно топить в момент рождения.

Они взяли идею самостоятельного расчета, которую я терпеть не могу и отказываюсь использовать в магазинах Home Depot или моем ближайшем супермаркете, и реализовали ее на колесном ходу, что мне еще более отвратительно. Портативный лазерный сканер они поставили на тележку. Сняв продукт с полки, вы подносите его к сканеру, прежде чем бросить в тележку. Ваша честность контролируется датчиком веса под корзинкой, а вес должен соответствовать данным сканирования. Для расчета следует провести кредитную карту через устройство чтения на тележке. Чтобы отоварить купон, надо подержать свой наладонник или сотовый телефон с инфракрасным портом перед экраном интеллектуального киоска, а потом под сканером.

Упаси нас Бог от этой ужасной самодостаточной технологии. Мало того что нам пришлось бы самим делать работу, которую сейчас делают за нас другие, но нам пришлось бы учиться и преодолевать мелкие огрехи сложной технической реализации (см. главу 1). Например, нельзя положить в тележку кошелек, или куртку, или пакеты с покупками из других магазинов, потому что они сбьют показания весов. Если вы решили вернуть просканированный и положенный в тележку товар, его снова придется сканировать, прежде чем положить обратно на полку, и не забыть нажать кнопку сторнирования. Несомненно, сканер очень заинтересует малышей («Смотри! Ой! Мой глаз!»), как и детей школьного возраста («Круто! Настоящая лучевая пушка! А ну-ка кокнуем ту старушку! Бах!»), не говоря о тинэйджерах. Чтобы купить простую плитку шоколада, при-

дется прочесть инструкцию, язык которой, конечно, местами будет очень похож на английский. Разумеется, в каждом магазине будут тележки особого типа, поэтому осваивать придется все. Наверное, большой ущерб розничной торговле может быть нанесен только в результате одновременного применения всех имеющихся средств ядерного поражения.

Служба доставки бакалейных товаров, такая как Peapod (*www.peapod.com*), предоставляющая возможность заказать продукты, которые будут потом доставлены вам на дом, решила бы эту задачу гораздо лучше. Она избавляет вас от траты времени не только на расчеты в кассе, но и на покупки и поездки, а это не меньше часа в неделю, попутно сокращая дорожное движение и общий расход энергии. В сельском районе Массачусетса, где я живу, эта служба приносит доход, и я думаю, что в густо населенном Сингапуре она должна оказаться еще более прибыльной. Если вы хотите сохранить привычную тележку для покупок, пометьте все пакеты бирками с радиочастотными идентификаторами, и тогда при расчете можно будет посчитать стоимость всей тележки одним нажатием кнопки, а не сканировать каждый предмет.¹ Но размещать на каждой тележке сканер, чтобы учитывать отдельные предметы, — это гигантский шаг назад.

Я поместил команду из Сингапура в середину, потому что они потратили больше сил, чем все другие. Остальные судьи — менее пресыщенные, чем я, или ценящие программирование больше, чем практичность, — дали им более высокие оценки, и в итоге они заняли третье место, заработав денежную премию. Если бы я знал, что они так близки к победе, я задвинул бы их еще дальше, чтобы им ничего не досталось. Можете не сомневаться — все эти свои соображения я высказал им по окончании соревнования.

¹ Они реально сегодня существуют и оказываются все чаще востребованными такими массовыми покупателями, как министерство обороны и Wal-Mart. При нынешней цене около 25 центов за штуку, они все еще слишком дороги для большинства отдельных упаковок. Сейчас их используют с отдельными дорогостоящими товарами типа телевизоров или наборов недорогих предметов. Но стоит цене упасть вдвое раза четыре подряд и достичь цента за штуку, как они появятся всюду. Предполагается, что это произойдет примерно в 2010 году. Подробнее см. *www.autoidlabs.org*. 15 ноября 2004 года FDA и крупные фармацевтические компании объявили, что вскоре эти идентификаторы появятся на аптечных пузырьках часто подделываемых лекарств, таких как виагра.

К счастью, на конкурс была представлена одна несомненная жемчужина. Двуязычная система ввода заказов официантами, разработанная Ту Нгуеном для вьетнамского ресторана, который держит его семья, привела меня в полный восторг и заслужила решение большинства судей присудить ему первую премию в 25 000 долларов. Повар, его отец, плохо говорит по-английски, а найти в Омахе, шт. Небраска, официантов, владеющих вьетнамским и английским, не так просто. Если Ту не собирался вечно исполнять роль посредника, ему нужно было придумать средство, благодаря которому ресторан смог бы функционировать без его участия. Он поместил меню ресторана в Pocket PC — карманный компьютер с беспроводным подключением к сети. Он достаточно дешев (\$200, и цены падают), поэтому можно обеспечить такими каждого официанта. Официанты вводят заказы на английском языке и посылают их по беспроводной сети в обычный PC, на котором работает серверная программа Ту. Эта программа с помощью таблицы переводила заказ на вьетнамский язык и распечатывала его на кухне.

Мало того, что перевод осуществлялся без участия человека, но еще и заказы поступали к повару быстрее, потому что официантам не надо было идти на кухню или стоять в очереди у единственного терминала. Отходы и ошибки из-за неправильно понятых заказов резко сократились, а о разгневанных клиентах, кричащих «разве я это заказывал, ты, [расовое оскорбление]?», было забыто. Заказы стали выполняться быстрее, уменьшилось время обслуживания и вырос объем продаж на единицу площади обслуживаемого пространства. Появилась возможность безошибочно вносить в блюда изменения в соответствии со вкусами посетителей, добавляя брокколи или убирая кинзу, а это понравилось клиентам, которые стали захаживать в этот ресторан и рассказывать о нем своим знакомым. Улучшились также отзывы о ресторане в местной прессе, поскольку процедура оценки профессиональными критиками всегда включает в себя исполнение особых заказов.

Ту не интересна технология ради нее самой, он сконцентрировался на бизнес-проблеме, которую хотел решить. Получившаяся у него маленькая, простая, прекрасно сконструированная система облегчила едва ли не каждый аспект управления этим рестораном. Если сменить таблицу переводов, можно работать с любыми другими языками — скажем, вводить заказы на испанском, а распечатывать на японском. Сейчас рассматривается возможность принятия этой системы для нового спортивного стадиона

на в Омахе, и я полагаю, что по окончании учебы у Ту будет выбор предложений от работодателей, если он не начнет собственного дела. Вот вам еще одна история успеха иммигранта в Стране Всеобщих Возможностей.

Я изо всех сил старался просветить все команды, у которых нет достаточного понимания проблем. Вычислительная техника перестала быть областью технологии, теперь это область человеческих отношений. Я старался повлиять на этих впечатлительных студентов, поощрял точную постановку задачи и карал поклонение технологии, и мне кажется — я надеюсь! — что из всего, чем я занимался на Tech Ed, это самый крупный мой вклад в данную отрасль — конечно, если не считать книгу, которую вы сейчас читаете.

ГЛАВА СЕДЬМАЯ

Так кто они такие, эти ненормальные?

Гик (сущ.): человек, испытывающий удовольствие и гордость от применения компьютеров или другой технологии, по мнению некоторых, иногда чрезмерные.

— надпись на футболке из магазина Microsoft Company Store

В предыдущей главе описывалось, как ведут себя гики, когда собираются большими группами. Пора взглянуть на отдельных представителей этого племени. Из того, что уже рассказано, должно быть понятно, что главная причина скверного качества современных программ — это непонимание их разработчиками характера пользователей, их потребностей и желаний, страхов и надежд. Из-за этого незнания (и незнания своего незнания) разработчики считают, что пользователи похожи на них самих. «Мне это нравится, — думают они, — и мне это очевидно. Следовательно, это должно понравиться всем и всем будет очевидно. Это же обычный здравый смысл, коллективный разум!»¹

¹ Ярчайший оксюморон. Нет ничего более индивидуального, чем разум. Всякий раз, сказав про себя «все так считают» или, что то же самое, «это коллективный разум», задумайтесь на секунду. Ведь глупо требовать, чтобы другие думали так же, как вы, когда вы потому и спорите с ними, что они думают иначе? Скажите лучше «я считаю». И вы поймете, что должны узнать, что действительно считает другой человек, а не приписывать и не навязывать ему свои представления. Но уже это проявление разума отнюдь не коллективного.

Homo Logicus

Неверно. Мироззрение гиков, которые создают программы, сильно отличается от мироззрения пользователей. И это не должно удивлять. Подумайте, насколько по-разному видят мир врачи и их пациенты: что для одних восхитительное, тающее во рту кремовое пирожное, запиваемое крепким кофе Starbucks (если их сайт снизойдет до того, чтобы показать вам, где его можно купить, — см. главу 2), то для других верный путь к инфаркту. Или вспомните, как по-разному видят мир католические архиепископы и их прихожане: «Стоит ли устраивать скандал из-за нескольких совращенных мальчиков у алтаря? Перевести священника в другой приход, и никто не заметит». Иногда кажется, что представители этих и некоторых других профессий (например, юристы) обитают в каком-то другом мире, и то же можно сказать о компьютерных гиках. И как человек становится врачом или священником, потому что ему лично кажется привлекательным представление людей этой профессии об окружающей действительности, так и профессиональным программистом человек становится потому, что мироззрение программистов оказывается привлекательным для того гика, который в нем живет.

Алан Купер, восхищение которым я уже выражал, обозначил тип личности, характерный для программистов, термином *homo logicus*. Я хочу показать некоторые отличия этого типа от других людей. Сначала на примере отдельных личностей, а затем покажу, как эти отличия проявляются в групповом поведении на одном из сборищ этих личностей. Не бойтесь, они не кусаются. По крайней мере, не на страницах этой книги.

Отравление тестостероном

Первое, на что обращаешь внимание на Tech Ed или любой другой компьютерной конференции, это что очереди во время перерывов выстраиваются не в женские, а в мужские туалеты. Гики — в подавляющем большинстве мужчины. Точные цифры найти трудно, но среди слушателей моего курса для опытных программистов 9% женщин¹, а в прош-

¹ Как-то я отчитал студентку, просившую перенести срок сдачи домашней работы. Она пожаловалась Розмари, моему старшему ассистенту: «Ведь правда, Дэйв плохо относится к женщинам?» И Розмари меня защитила: «Пол тут ни при чем. Дэйв со всеми обращается как с мерзавцами». Ну не прелесть ли она!

лом году их было 12. Другие преподаватели факультета указывают аналогичные цифры.

Да и собственно в индустрии ПО — чем более трудной считается некоторая область, тем больше в ней мужчин. Все эти десять лет, что я трудился в Microsoft над низкоуровневыми проблемами, мне встретились в этой сфере одна или две женщины, которые занимались разработкой. Популяризация, администрирование, продажи, поддержка — пожалуйста, но почти каждый из моих знакомых, кто изо дня в день копошится во внутренностях компьютеров, относится к лицам мужского пола с наличием хромосом X-Y. (Женщины, занимающиеся программированием на низком уровне, не пишите мне гневных писем. Я знаю, что вы есть, и не сомневаюсь, что вы умеете работать, — просто вас мало. И если вы выражаете желание, чтобы вас было больше, то моя жена выражает опасение, что на самом деле вы просто не хотите стоять в очередях в женские туалеты на конференциях медицинских сестер.)

Я так думаю, что дело здесь не в умственных способностях. За 12 лет преподавания самой толковой из моих студентов была Дафна, окончившая химический факультет МТИ. Она не только получала высшие оценки за все задания (и всегда сдавала их вовремя, что никому не удавалось ни до, ни после нее), но и выполняла все дополнительно предлагаемые мной проекты, а также другие, которые придумывала по собственной инициативе, одновременно воспитывая двух детей школьного возраста. К несчастью для вычислительной техники, она предпочла заняться ювелирными изделиями из бисера (www.daphnesdesigns.com), а не писать код вместе с прочими гиками.

Когда я спросил у Дафны ее мнение по поводу того, почему среди программистов так мало женщин, она ответила: «Часто люди выбирают, куда им пойти учиться, в старших классах школы, а там дети нередко скрывают свои способности, чтобы не выделяться из общей массы. И для девочки в общественном отношении быть гиком в десять раз хуже, чем для мальчика. У моей дочки, увлеченной физикой, гораздо больше проблем в общении с девочками, чем с мальчиками. Те прощают такие огрехи общественного поведения, как разговоры о квантовой механике, но девочки к ним нетерпимы. Кроме того, моя дочка терпеть не может разговоры о прическах, тряпках и мальчиках, а именно об этом говорят почти все девочки».

За последние 50 лет доля женщин-врачей выросла от совершенно незначительной до более чем половины выпускников медицинских коллед-

жей, и это сильно изменило медицину как профессию. Многие считают, что недавние скандалы в Католической церкви, связанные с развратными действиями в отношении детей и попытками их скрыть, отчасти обусловлены тем, что церковная иерархия полностью состоит из мужчин. Доминирование мужчин в компьютерной индустрии почти наверняка сильно влияет на разработку ПО. Может быть, это одна из причин, по которым мы наблюдаем появление дурацких, неэффективных, «крутых» функций типа плавающей панели меню, о которой я писал во введении, вместо полезных вещей, действительно облегчающих наш труд. Не исключено, что если бы среди программистов было больше матерей с детьми, мы увидели бы безжалостное изменение приоритетов, от которого выиграли бы как отрасль, так и общество в целом.

Пако Андерхилл (Paco Underhill) описал в книге «Why We Buy: The Science of Shopping» (Почему мы покупаем: наука шоппинга), Simon and Schuster, 1999 результаты своего тщательного гендерного исследования покупателей: «Мужчины влюблены в технологию как таковую, в возможность сказать: «Вот это да, черт побери!», похвастаться «крутизной» своей новой игрушки... Собравшись вокруг барбекю, они сравнивают размеры своих жестких дисков и скорости своих модемов. Чистое фатовство». У женщин «... совсем иной подход к высоким технологиям. Они превращают технологии в бытовую технику. Самую невероятную игрушку они лишают всякой таинственности, стараясь выяснить ее утилитарность. Сталкиваясь с технологией, женщины видят ее цель и смысл — чем она может быть им полезна. Технологии всегда обещают сделать нашу жизнь легче и эффективнее. Женщины как раз и требуют, чтобы технология сдержала свое обещание». И я хотел бы увидеть, как исполняются эти обещания, поэтому отчасти и написал эту книгу. Может быть, мои дочери помогут этому осуществиться.

Контроль и удовлетворенность

Гикам необходимо чувствовать, что они постоянно контролируют ситуацию. Возможно, эта потребность возникает у них в школе. Они не делали успехов в спорте, девушки не хотели с ними разговаривать, зато они могли в любой момент включить или выключить этот желтоватый ящик в углу. И если они досконально — с точностью до бита — инструктировали его, он выполнял их команды. Как я писал в главе 1, гики пред-

почитают машины с ручной коробкой передач из-за контроля, который она им дает. Они не возражают против лишней работы — как сначала, во время учебы, так и потом, когда приходится постоянно, изо дня в день, переключать передачи во время езды. Они считают это приемлемым компромиссом (так и должна работать машина) и гордятся, что у них достаточно ума, чтобы понять это и выполнять. Теоретически они знают, что есть люди, которые с ними не согласны, но считают, что такие чудачки составляют незначительную долю населения. Но им непонятен такой образ мыслей, и они очень удивляются, узнав, что «незначительная доля» — это почти каждый.

Страстно стремясь к контролю, гики постоянно стараются разобраться, как что работает и как заставить его работать лучше. Чтобы пояснить это своим ученикам, я часто задаю им вопрос: «Когда в последний раз вы испытывали удовлетворенность? Радость, печаль, голод, гнев, особенно по отношению ко мне, — это да, но я спрашиваю об удовлетворенности. Когда в последний раз вы ее чувствовали? Вообще когда-нибудь вы ее испытывали? Это не риторический вопрос. Назовите дату, время и место». Никто еще не ответил мне. Ни разу. Отчасти это обусловлено социальными причинами. Несмотря на все наше богатство, наша эпоха не эпоха удовлетворенности. Сотрудник Института Брукинса и комментатор Национальной футбольной лиги Грегг Истербрук (Gregg Easterbrook) написал в книге «The Progress Paradox: How Life Gets Better While People Feel Worse» (Парадокс прогресса: жить становится лучше, а люди чувствуют себя хуже), Random House, 2003: «Если когда-нибудь человек снова окажется в раю, то он начнет жаловаться на меню, в котором вечные молоко и мед, и на дружелюбных львов, которые слишком громко мурлычут». Но и в нашем современном обществе программирование оказывается профессией, которая по своей природе чужда удовлетворенности и влечет к себе людей, у которых неудовлетворенность — черта характера. Конечно, для подлинного гика стакан не является ни наполовину полным, ни наполовину пустым: он просто вдвое больше, чем ему следовало быть, хотя наличие разумного резерва может не помешать.

Программисты целыми днями трудятся над тем, чтобы сделать свои программы капельку лучше (в соответствии со своими представлениями о том, что такое «лучше», часто не совпадающими с представлениями пользователей), и еще немножко лучше, и еще, потому что они принадлежат к такой породе людей. Так устроена их психика, и они не могут вы-

ключить ее, выйдя за стены своей лаборатории. Они и во время завтрака думают над тем, нет ли такого лучшего способа, который человек с их интеллектом обязан увидеть. Привожу пример из романа Нила Стивенсона о гиках (Neal Stephenson «Cryptonomicon», Arrow, 2000):¹

«Правильное поедание каши — это сочетание тонких компромиссов. Доверху наполненная огромная миска с крупой, залитая молоком, выдает новичка. В идеале требуется, чтобы абсолютно сухие крупинцы и ледяное молоко попадали в рот, минимально контактируя, и вся реакция между ними происходила во рту. Рэнди [главный гик] разработал эскизы особой ложки для поедания каши, у которой вдоль ручки шла трубка, а молоко накачивалось маленьким насосом, благодаря чему можно черпать ложкой из миски сухую крупу, нажимать большим пальцем кнопку и впрыскивать в ложку молоко в тот момент, когда вы заносите ее в рот. Менее достойный способ — поглощать кашу маленькими порциями, кладя в миску немного хлопьев и съедая их, пока они не превратились в отвратительную слизь, на что в случае Cap'n Crunch требуется примерно тридцать секунд».

Сотворение моделей

Как писал Фредерик Брукс в классическом труде «Мифический человеко-месяц»², программисты работают «почти исключительно с чистой мыслью». В отличие от тех, кто выпускает, скажем, кофейные зерна, или ручные гранаты, или книги — вроде той, что вы держите в руках, — программист не производит ничего конкретного, осязаемого. Врач имеет дело с реальными костями, носами или внутренностями, а программист манипулирует лишь мысленными конструкциями, созданными им самим или другими программистами.

Гики, следовательно, представляют себе мир с помощью абстрактных моделей, в отличие от представителей других профессий. Например, стек — это такая структура памяти внутри программы, при которой последний

¹ Несмотря на объем (900+ страниц), действующих лиц-гиков и сюжет, основанный на криптографии и сложных проблемах теории чисел, роман занял двенадцатое место в списке бестселлеров «New York Times», и я весьма его рекомендую. Если вам нравится *Why Software Sucks*, то должен понравиться и *Cryptonomicon*.

² Фредерик Брукс «Мифический человеко-месяц или как создаются программные системы». — Пер. с англ. — СПб.: Символ-Плюс, 2000.

фрагмент информации, помещаемый в нее программистом, становится первым, из нее извлекаемым. Ситуация аналогична стопке бумаг, лежащей на столе. Очередь — это такая структура памяти внутри программы, при которой первый фрагмент информации, помещаемый в нее программистом, становится первым, из нее извлекаемым, что аналогично тому, как люди выстраиваются в очередь друг за другом. Очередь с коммутацией пакетов — это такая очередь, в которой объекты, ожидающие обслуживания, выстраиваются в очередь, а каждый сервер, закончив выполнение текущего задания и освободившись для нового, берет первый элемент с переднего конца этой очереди («Следующий, пройдите, пожалуйста к окошку 4»). Кассиры в банке или посты регистрации пассажиров обычно действуют в режиме очереди с коммутацией пакетов, тогда как в супермаркетах обычно выстраиваются отдельные очереди к каждой кассе. «Неужели вам непонятно, что в этой очереди должно быть переключение пакетов?» — возмущенно говорит гик менеджеру супермаркета, когда приходится торчать за дряхлой старушкой, которая ищет свою чековую книжку, чтобы расплатиться за конфетки.

Гики живут этими мысленными моделями и обожают их. Например, в прошлом году я три недели выступал с докладами вместе с таким же гиком, как я сам. Однажды, завтракая в ресторане, я обнаружил отсутствие вилки и стянул ее с соседнего места. Вскоре подошел мой коллега и сел на это место, и так как у него не было вилки, он стянул ее с соседнего стола. Вскоре туда сел еще один житель гостиницы и украл вилку с соседнего столика. Мы с коллегой целыми днями спорили по поводу того, двигалась ли украденная вилка по часовой стрелке по комнате или же «вилко-дырка», или «анти-вилка», двигалась в противоположном направлении. И что произойдет, когда она достигнет стены: отразится и двинется в обратном направлении или будет совершать колебательные движения между двумя последними столиками? Будет ли за обедом такая же картина, как за завтраком, и если нет, то почему? Мы дошли до того, что взяли вилку с соседнего стола, хотя она не была нам нужна, только чтобы посмотреть, что будет дальше. Когда официант заметил отсутствие вилки и положил новую (это было в «Ritz—Four Seasons» в Лиссабоне; там следят за такими вещами), мы сказали: «Что ж, вилка и анти-вилка аннигилировали друг друга. Черт побери, он нам все испортил!»

Наверно, вы думаете, что мы слишком далеко зашли, возможно, вы правы. Но мы не считаем себя сумасшедшими или людьми со странностями.

ми, хотя иногда с удовольствием разыгрываем эту роль перед публикой.¹ Нам кажется, что нас интересует окружающий мир и его устройство, что у нас достаточно интеллекта, чтобы обнаружить неясные закономерности, которым подчиняются никак не связанные на первый взгляд события, что мы можем быстро понять какое-то новое явление, используя эти открытые нами закономерности и связав его с уже понятными вещами. И нам жаль людей, у которых недостаточно интеллекта, чтобы видеть мир таким, каким видим его мы. Когда вы в следующий раз будете в ресторане и обнаружите, что у вас нет вилки, оглянитесь вокруг — наблюдаю ли я за вами, нахально положив украденную вилку перед собой.

Гики и качки

Homo logicus может показаться совсем незнакомым видом, но Алан Купер в своей книге «The Inmates Are Running the Asylum», SAMS, 1999² отмечает, что можно многое узнать о них, если сравнить с более знакомыми видами, например с *homo athleticus*, качком обыкновенным. На первый взгляд никакого сходства. Качки физически развиты, гики — нет. Качки в центре внимания и уважения в старших классах, гики — нет. Девочки выются вокруг футболистов, большинство которых (не все, Брайан) глупы, как бревна. Мачо неприлично быть круглым отличником (если бы я им был, стало бы прилично), и даже мой кубок за шахматные победы не мог состязаться с письмом из университета. И хотя я знал, что в конечном счете стану зарабатывать много больше, чем качки из старших классов (что, как отмечал мой отец, гораздо сильнее привлекает противоположный пол), все равно было обидно. Общество стало подозрительно к нам относиться с тех пор, как первый пещерный гик подобрал острый камушек и заявил: «Какой замечательный фрактальный узор! Нельзя ли масштабировать его до размера наконечника копья?» — «Пойди и возьми мамонта голыми руками, как положено неандертальцу!» — ревел в ответ

¹ Вот отличный способ: включите компьютер и громко озвучивайте все, что он выводит на консоль, на полсекунды опережая сам вывод. Если делать это, закрыв глаза и сжав кулаки, то достигается особый эффект.

² Алан Купер «Психбольница в руках пациентов или почему высокие технологии сводят нас с ума и как восстановить душевное равновесие». — Пер. с англ. — СПб.: Символ-Плюс, 2004.

пещерный качок. Подобно покойному Родни Дэйнджерфилду, мы, гики, всегда жаждем почета и уважения.

Однако эти две группы во многом схожи. В мои студенческие времена тех, кто хорошо разобрался в операционной системе, даже называли «компьютерными качками» (computer jocks). Те и другие полны высокомерия: одни по поводу своего атлетизма, другие — своего интеллекта.¹ Благодаря своему высокомерию те и другие склонны к соперничеству, особенно среди своих, хотя соперничество гиков не так заметно окружающим, потому что в нем участвуют мозги, а не мускулы. И обе группы безжалостно топчут менее способных, особенно посторонних, которых они считают недостойными внимания. Вспомните создателей компьютерных вирусов — не тех, кто стремится украсть деньги, такие просто мошенники, а о тех, кто взламывает системы из чистого удовольствия. Сравните их с известными накачанными хулиганами. Сходства больше, чем вам кажется.

Качкам после окончания школы пришлось начать вести себя по-другому, ибо босс, дававший им работу, или сержант-инструктор научили их так держать строй, как не снилось никаким учителям. От гиков ничего похожего не требовалось. Напротив, с тех пор как их искусство стало востребовано обществом в большей степени, чем умение кидать мячик, они стали еще противнее. Купер пишет: «Стало совершенно приемлемым такое поведение, которое причиняет умственные страдания непостижимостью программы или наносит эмоциональную травму человеку, который просто хочет получить деньги в банкомате».

Чтобы программы перестали быть противными, ситуация в мире гиков должна измениться. Говорят же родители своим буйствующим старшим детям: «Эй, поаккуратнее с маленькими». Гики больше не играют в игры с самими собой, как 20 лет назад. На самом деле они значительно уступают в численности невоющим участникам: своим пользователям, своим клиентам, тем, кто платит им зарплату. Программы плохие не потому, что так обязательно должно быть, а потому, что гики задурили этим головы народу. Если программа противная, так это потому, что ее такой написали плохие программисты, которым наплевать на пользователей. Мы не стали

¹ Отсюда я вывел еще один закон, т. н. платтовский парадокс программиста, гласящий, что ума у нас столько, сколько нужно, чтобы делать то, что нам хочется, а глупости столько, сколько нужно, чтобы выбрать то, чем мы занимаемся. Разбирайтесь сами.

бы терпеть, если бы врачи или пилоты авиалиний стали вести себя подобным образом. Так почему мы должны терпеть это от гиков? Не должны. В заключительной главе я еще скажу об этом.

Жаргон

Один из основных барьеров, которыми всякая замкнутая группа отгораживается от посторонних, — это жаргон — особый, только ей понятный язык. Священники говорят на латыни, дошкольники — на «поросячьей латыни», а у воров в «Оливере Твисте» Чарльза Диккенса было свое аргю, на котором Ловкий Плут, он же Джек Даукинс, называл чистку ботинок «полированием скакунов». Неудивительно, что гики тоже разговаривают на своем языке, не доступном для понимания всем остальным. Когда я начинаю болтать с какими-нибудь приятелями-программистами, моя жена пожимает плечами и говорит: «какая-то гиковщина». Я очень старался избегать гиковского жаргона в этой книге, чтобы не пугать читателя и покупателя. Если бы вы только знали, как трудно это было.

Гики любят не только новые слова, которые представляют абстрактные мысленные модели гиков, они любят придумывать акронимы, из начальных букв которых образуется новое слово. Так, известное слово «радар» появилось как акроним для radio detection and ranging (обнаружение и определение расстояния с помощью радио), а вездесущее слово «модем» произошло от «модулятор/демодулятор». Очень любят акронимы врачи. Важным источником акронимов является военная область. Возникшие в армии SNAFU и FUBAR стали настолько общеупотребительными, что в моем редакторе их уже автоматически распознает программа проверки орфографии. Гики пользуются такими акронимами, как DLL (Dynamic Link Library — библиотека динамической компоновки, их много в Windows) и SDK (Software Development Kit — инструментальный набор разработчика программ). Поскольку мы все же гики, то сделали еще один шаг и придумали акроним для такого типа акронимов. Их назвали TLA — сокращение от Three-Letter Acronym (трехбуквенный акроним). Разумеется, TLA и сам представляет собой TLA.

Будучи гиками, мы быстро поняли, что проблема TLA в том, что есть всего 17 576 уникальных TLA для алфавита из двадцати шести букв, и Microsoft их уже давно все израсходовала. Можно было бы увеличить количество, добавив специальные символы типа Р и Ю но они редко

встречаются на клавиатурах за пределами Исландии.¹ Вместо этого гики увеличили длину акронимов, создав такие, как WSDL (Web Service Descriptor Language) и SOAP (Simple Object Access Protocol).² Акронимы этого типа оставались без названия, пока в статье, опубликованной в 1998 году, я не назвал их FLAP. Это сокращение от Four-Letter Acronym Package (четырёхбуквенный акронимный пакет), и, разумеется, FLAP сам есть не что иное, как FLAP.

В итоге получается 456 976 разных комбинаций, но нынешним гикам понадобилось менее 20 лет, чтобы радость по поводу памяти размером в 64000 символов сменилась недовольством компьютерами, у которых ее в 64000 раз больше, т. е. порядка 4 миллионов символов. Поэтому я полагаю, что набор FLAP будет быстро исчерпан. Готовясь к тому моменту, я придумал акроним FLEAP (Five-Letter Extended Acronym Package; FLEAP — сам по себе FLEAP), который даст нам около 12 миллионов различных комбинаций. Гики уже изобрели OLEDB (не стану утомлять вас определением), и наши старые друзья SNAFU и FUBAR также представляют собой FLEAP. Когда и FLEAP закончатся, можно воспользоваться моим изобретением SLEAPE, произносящимся как «sleepy». Съездив в Канаду, я получил заряд вдохновения и придумал, что это сокращение означает Six-Letter Extended Acronym Package (шестибуквенный расширенный акронимный пакет). Для алфавита из 26 букв есть свыше 300 миллионов SLEAPE, и на какое-то время их должно хватить. Первые эксперименты с SLEAPE дали сладкозвучное слово MOOTWA (Military Operations Other Than War — военные операции, но не война), каковым обозначают процесс национального строительства в Ираке и помощь потерпевшим от урагана в Луизиане, хотя я не уверен в том, правильно ли считать это акронимом, поскольку последняя буква A не соответствует никакому слову. В качестве SLEAPE для гиков у нас есть PCMCIA,

¹ Эти буквы до сих пор употребляются в исландском языке и представляют собой пережиток древнеисландского. Первая — это буква «eth», которая произносится как звонкая th в though. Вторая называется «thorn» и произносится как глухая «th» в thought.

² Как Священная Римская империя не была ни священной, ни римской, ни империей, так и Simple Object Access Protocol (Простой протокол доступа к объектам) отнюдь не прост и не имеет никакого отношения к объектам (популярная технология программирования). Гики не могут удержаться, чтобы не воспользоваться красивым акронимом, даже если он не совсем точен.

что первоначально означало тип карты расширения для ноутбуко РС, но скоро стало расшифровываться как «People Can't Memorize Computer Industry Acronyms» (люди не могут запомнить акронимы компьютерной промышленности). Конечно, SLEAPE и сама представляет собой SLEAPE.

Мозги и препятствия

Гики особенно восхищаются блестящими решениями, нахождение которых усложняется ограничениями. Так качок гордится тем, что поколотил кого-нибудь одной рукой, и примерно так мы удивляемся, что огромный расписной хвост не помешал выжить павлину. «Если человек нашел это сногшибательное решение при таких ограничениях, — думает гик про себя, — то на что же он способен в их отсутствие?»

Одно из их любимых ограничений — это хайку или хокку (haiku), японская стихотворная форма из трех нерифмованных строк с пятью, семью и снова пятью слогами. В феврале 1998 года журнал «Salon» провел соревнование, в котором читателям предлагалось написать в виде хайку сообщение компьютера об ошибке. Питер Ротман (Peter Rothman) получил поощрительную премию за следующее:

Windows NT crashed.	(Windows NT обрушилась.
I am the blue screen of death.	Я синий экран смерти.
No one hears your screams.	Твой крик не слышен.)

Чарли Гиббс (Charlie Gibbs) описал отношение большинства программистов к неприятной и утомительной обязанности программы общаться со своими пользователями:

Errors have occurred.	(Случились ошибки.
We won't tell you where or why.	Не скажем — где, почему.
Lazy programmers.	Лень программистов.)

Гран-При получил Дэвид Диксон (David Dixon) за описание вселенских событий:

Three things are certain:	(Три вещи верны:
death, taxes, and data loss.	Смерть, налог, гибель данных.
Guess which has occurred.	Догадайся, что это.)

А вот что написал я после поездки в Торонто:

My cat pressed Reset.	(Нажала «Reset»
Now she's a tennis racket.	Кошка. Ее больше нет.
No more furballs, eh?	Жаль милых котят.)

Платино-иридиевый стандарт гиковских хайку озаглавлен «DeCSS Decryption». Группа европейцев дизассемблировала программу DVD-плеера и вскрыла схему шифрования, которая должна была препятствовать копированию DVD-фильмов. Программа для дешифрования была запрещена в США согласно закону Digital Millennium Copyright Act (закон об авторских правах в цифровую эпоху), потому что ее основным назначением было выполнение незаконных операций. Однако указания о том, как написать такую программу, значительно труднее запретить, чем саму программу, ввиду Первой поправки, защищающей свободу слова. Чтобы придать инструкциям по дешифрованию художественную ценность и благодаря этому еще сильнее защитить их Первой поправкой в качестве произведения искусства, один сумасшедший гик (говорю это в самом комплиментарном смысле, что он, я уверен, поймет, будучи настоящим гиком) по имени Сет Шон (Seth Schoen) записал указания в виде невероятного хайку, состоящего из 456 строф, с соблюдением всюду слогового размера пять-семь-пять. Все это произведение я не стану приводить целиком (есть соответствующая ссылка на сайте этой книги), но вот вам образец:

Now help me, Muse, for	(Помоги Музы
I wish to tell a piece of	прошу, чтоб поведать
controversial math,	о спорном коде,
for which the lawyers	что защитники
of DVD CCA	прав DVD CCA
don't forbear to sue:	преследуют судом.
that they alone should	якобы лишь у них
know or have the right to teach	есть право учить
these skills and these rules.	этому мастерству и правилам)

Я показывал это хайку многим гикам. Общая реакция — восторг, качание головой, часто с почтительным произнесением ругательств. Я серьезно подумывал о том, чтобы забросить писание собственных хайку, чувств-

вужа, что никогда не смогу приблизиться к шедевру Шона. Но прежде чем сделать это, я должен отдать ему дань — в форме хайку, разумеется:

Someone with enough Time on his hands to write this Must have no mortgage.	(Тот, у кого хватило времени, чтобы это написать, должно быть, не знает,
I am astounded, Awed, amazed, flabbergasted. Ye gods, what a geek.	что значит платить по закладной. Я потрясен, изумлен, ужасаюсь. Боги! Вот это гик!)

Семь привычек гиков

Юморист По Бронсон (Po Bronson), пародируя бестселлер Стивена Кови (Steven Covey) «The Seven Habits of Highly Effective People» (Семь навыков высокоэффективных людей)¹, сочинил «Семь навыков выдающихся инженеров». Вот они:

1. Они эгоистичны в своем благородстве и стараются выглядеть благородными в своем эгоизме.
2. Слепота улучшает их зрение.
3. Они укусят не только ту руку, которая их кормит, но и свою собственную.
4. Они изо всех сил делают вид, что их совершенно не волнует создаваемое ими впечатление.
5. Они будут чинить то, что работает, пока оно не сломается.
6. Не я дал неправильный ответ, а вы задали неправильный вопрос.
7. Отсутствие критики можете рассматривать в качестве комплимента.

Я очень смеялся, когда прочел это впервые, на что Бронсон и рассчитывал. Но эти принципы не давали мне покоя, пока я не изучил их внимательнее. По дальнейшему размышлению я пришел к выводу, что Бронсон высказал истину полнее, чем, возможно, представлял себе. Вот как принципы Бронсона преломляются в отношении гиков:

1. «Они эгоистичны в своем благородстве и стараются выглядеть благородными в своем эгоизме». В главе 1 мы видели, что программисты

¹ Стивен Кови «7 навыков высокоэффективных людей» — Пер. с англ. — М.: Альпина Бизнес Букс, 2007.

просто отображают внутренний механизм работы программы, чтобы избавить себя от необходимости разбираться, что собой представляют их пользователи и каковы их реальные потребности, а затем проектируют программу так, чтобы им проще было ее сделать. Классический пример — постоянный запрос подтверждений у пользователя: «А вы уверены, что хотите сделать это? Точно уверены? Точно-точно-точно?» Вместо того чтобы дать возможность отменять действия (что удобно пользователям, но трудно программируется), они перекладывают ответственность на пользователя. «Слушай, приятель, если тебе это нужно, можешь его сохранить, а нет — так выброси. Если ты такой тупой, что не знаешь, чего хочешь, так нечего садиться за компьютер». У программистов это называется «мощь». Неправильно. Это называется плохим программированием. Безалаберным. И даже идиотическим (худшее оскорбление). И с таким программированием необходимо покончить.

2. «Слепота улучшает их зрение». Мы тоже это видели в главе 1. Помните, как плохая программа спрашивает «Вы хотите сохранить изменения?», прежде чем записать файл обратно на диск? Разработчик интерфейса слишком хорошо знал внутреннее устройство программы (возможно, потому, что и то и другое писал один человек) и предполагал, что пользователю оно тоже известно. Знай он поменьше о внутреннем устройстве программы, оно его не отвлекало бы. Он бы сосредоточился исключительно на пользователе и его потребностях. Он сделал бы интерфейс лучше, если бы меньше знал о программе, подобно тому как из одних и тех же звуков слепой получает больше информации, чем зрячий.

3. «Они готовы укусить не только ту руку, которая их кормит, но и свою собственную». В главе 2 мы видели, как UPS кусает руку, которая ее кормит, — клиентов, платящих за доставку пакетов. Их веб-сайт требовал, чтобы каждый пользователь вручную ввел название своей страны, и только потом соглашался с ним разговаривать, в отличие от Google, определяющего страну автоматически. Они не понимают, что кусают и самих себя, хотя это так. Сайт, с которым легко работать, привлекает больше пользователей. Тот, с которым работать сложнее, отпугивает их. В один прекрасный день UPS обнаружит, что конкуренты написали сайт, который лучше, и увели у них часть клиентов. Не украли клиентов, а (в старомодном стиле ведения бизнеса) заслужили успех, выяснив, чего хотят клиенты (даже если сами клиенты думают, что хотят чего-то другого), и предоставив им это.

4. «Они изо всех сил делают вид, что их совершенно не волнует создаваемое ими впечатление». В этой книге я подчеркиваю, что культура гиков ценит интеллект превыше всяких других качеств. А чье имя служит синонимом высочайшего уровня человеческого интеллекта? Разумеется, Эйнштейна (рис. 7.1). Трудно переоценить его влияние на целые поколения гиков в отношении стиля одежды и, как отметил Стивен Ландсберг (Stephen Landsburg), внешнего вида вообще. Гики презирают книги типа «Одежда, приносящая успех», потому что в их мире успех измеряется интеллектуальным одобрением коллег. Словом «пиджак» презрительно обзывают менеджеров, которые не умеют держать паяльник. Для *homo logicus* важно показать окружающим, что они такие умные, что не обязаны играть по общим правилам. Это наглядное проявление собственного образа, создаваемого гиком. Однако гиковский код пренебрежения одеждой не менее строг, чем тот, который они отвергают. Я знаю не один случай, когда гик, чтобы показать, что его не волнует создаваемое им впечатление, отказывался от места, где ему платили бы на 10 000 больше, только потому, что там ему пришлось бы ходить на работу в галстуке.

5. «Они будут чинить то, что работает, пока оно не сломается». Или, как говорил персонаж одной канадской комедии, «если оно не сломано, значит, ты плохо старался». Я говорил об этой тенденции в главе 1. Гики стремятся делать возможными сложные вещи, вместо того чтобы де-

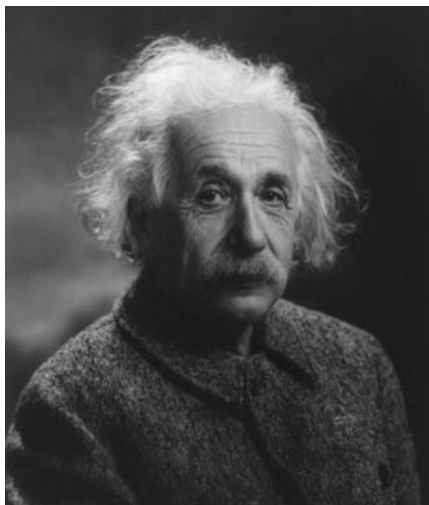


Рис. 7.1. Фирменный взъерошенный вид Эйнштейна

лать простые вещи простыми. Телефонная справка AT&T не была сломана: она работала прекрасно, когда просто сообщала вам нужный номер, повторяла его, чтобы вы успели записать, и вешала трубку. Затем умельцы из AT&T добавили возможность автоматического набора номера, для чего требовалось нажать кнопку на телефоне после того, как голос сообщил вам номер. И это работало хорошо, поэтому понадобилось поменять что-то еще. Тогда гики придумали, что пользователь должен принять или отклонить автоматический набор номера до того, как услышит номер, что требовало дополнительно семь отдельных физических движений при каждом обращении в справочную службу, даже если вы не собирались пользоваться автоматическим набором. Это было названо «усовершенствованием». Если их целью было уменьшить активность обращений к справочной службе, они ее блестяще достигли. AT&T, «Ma Bell» (Bell Telephone System), изобретатель транзистора и лазера, почил в бозе. История. Гост.¹ Совпадение? Как знать...

6. «Не я дал неправильный ответ, а вы задали неправильный вопрос». Более яркий пример такого поведения, чем веб-страница для отыскания магазинов Starbucks, которую я описал в главе 2, трудно придумать. Мне нужен был ближайший магазин Starbucks для моего почтового ZIP-кода, и страница сказала, что не может такой найти. Даже если бы во всей вселенной существовал всего один магазин Starbucks, то это был

¹ Компания, известная сейчас как AT&T, раньше называлась SBC (была создана при распаде первой AT&T в 1983 г.). Останки AT&T были куплены в 2005 г. за жалкие \$16 миллиардов — примерный размер прибыли Microsoft в том году. То, что было сделано потом, несомненно, войдет в анналы как одно из самых безмозглых бизнес-решений в истории: они решили все переименовать в AT&T, даже потратили уйму денег на переименование успешной компании беспроводной связи Cingular, появившейся ранее в результате покупки AT&T Wireless и раскрутки ее под именем Cingular, на которую было потрачено много миллионов. Нет сомнений, что они это делали потому, что их высшее руководство продвигалось по службе в те времена, когда в телекоммуникационной отрасли к названию AT&T относились с божественным почтением. Но это имя почти ничего не говорит клиенту, которому нет 30 лет, потому что эта компания перестала кого-либо интересовать как раз в то время, когда эти клиенты стали замечать подобные вещи. Обращаюсь к руководству SBC, как и ко многим другим компаниям: перечитайте мою первую главу и зарубите себе на носу, что ваш пользователь — это не вы.

бы магазин, ближайший к моему дому. Он может быть за углом, или на другом конце страны, или на Марсе, но один магазин должен быть ближайшим. Веб-сайт сообщил мне, что такого не существует. Только после долгого блуждания я обнаружил кнопку, которая ограничивала поиск радиусом в пять миль, а ближайший магазин оказался в восьми. «Вы меня спросили, есть ли магазин ближе пяти миль отсюда, и я вам ответил», — таков ответ сайта. Осел, это не я задал неправильный вопрос. Это ты задал его вместо меня.

И наконец

7. «Отсутствие критики можете рассматривать в качестве комплимента». Гики считают отсутствие критики не просто комплиментом, но высшим комплиментом, а в действительности единственным комплиментом, который можно сделать или принять. Подумайте, случилось ли вам быть разбуженным в половине пятого утра в выходной день звонком с работы, который сообщал: «Все работает отлично, никаких проблем, желаем вам приятного отдыха, можете снова лечь в постель»? Если стоит тишина, значит, все идет хорошо. Тишина, которой вы, любезный читатель, позволяете им наслаждаться, вызывает у них ощущение полного порядка. Когда я объясняю ребятам из UPS.com, как бестолково у них организована процедура выбора языка, они говорят в свою защиту: «А никто не жаловался».

Частично дело в том, что в разделе обратной связи их сайта работать так же неудобно, как и в остальных местах, о чем я писал в главе 2. Нужно пройти пять экранов с непонятными заголовками, чтобы добраться до бланка претензий, потом заполнить семь разделов и только тогда его можно отправить. (Я опубликую на сайте этой книги правильную ссылку, чтобы облегчить процедуру и дать вам возможность завалить их жалобами.) Но отчасти дело в том, что вы не подозревали, что все можно и нужно сделать лучше, что ваше программное обеспечение не должно быть противным. Теперь вы знаете. Прочтите заключительную главу, и я объясню вам, как.

ГЛАВА ВОСЬМАЯ

Microsoft: ни с ней жить нельзя, ни без нее¹

Скажем прямо. Microsoft правит балом в мире ПО, как бы кто-то ни пытался доказать обратное. Если мы хотим, чтобы программы перестали быть отстоем, то в первую очередь должны давить на Microsoft. Когда Microsoft поднимает уровень своих продуктов, то же самое вынуждены делать поставщики веб-серверов на машинах Sun или IBM, то есть даже в тех областях, которые ей не подчиняются. Точно так же как возможность купить хороший кофе в сотнях магазинов Starbucks (см., например, заголовок в колонке интернет-юмора: «Новый Starbucks откроется в мужском туалете действующего Starbucks») вынудила других продавцов кофе улучшить качество своего.

Они правят миром

Ни отдельные люди, ни даже организованные группы не могут заметно влиять на Microsoft: она для этого слишком велика. Но если разобраться, какой образ мышления в ней господствует, то можно понять, какие способы борьбы с Microsoft более эффективны, а какие менее. О том, какими себя видят гики, мы говорили в главе 7, а в этой я постараюсь продемонстрировать, какой видит себя Microsoft. Вы увидите, что быть Microsoft труднее, чем может показаться.

¹ Или, как говорит мой парикмахер, «и жить с ними нельзя, и пристрелить тоже».

Отношения между Microsoft и остальным миром проявляются в анекдотах, которые о ней рассказывают. Время от времени я буду приводить их в этой главе.¹ Как всякое иносказание, они содержат зерно если не истины, то мысли — так ложка сахара нужна, чтобы выпить горькое лекарство. Например, следующая распространенная шутка свидетельствует о всеобщем убеждении, что Microsoft правит миром программного обеспечения жесткой рукой и не всегда стремится к нововведениям:

В: Сколько потребуется программистов Microsoft, чтобы заменить электрическую лампочку?

О: Нисколько. Билл Гейтс просто объявит новый стандарт Microsoft® Темнота™.

Я и они

Прежде чем пускаться в обсуждение Microsoft, я должен полностью раскрыть суть моих отношений с ней. Я не являюсь в настоящее время, никогда не был и вряд ли стану — после того, как Билл Гейтс прочтет эту книгу, — сотрудником Microsoft. Иногда я консультирую их на временной основе либо преподаю у них или от их имени, и занимаюсь этим в той или иной мере с 1994 года. В урожайные годы мои доходы от этих занятий описывались пятизначным числом, в иные — нулем. Предыдущие два года были средними, таким же, видимо, окажется и этот год.

У меня есть 1300 акций Microsoft, текущая цена которых составляет около \$31 000, и я купил их за полную цену на открытом рынке в период 2000 и 2001 годов. За пять лет они выросли в цене на 11% — примерно такой же доход за этот срок мог дать банковский сберегательный счет. Акции одних технологических компаний были прибыльней (Amazon.com вырос за это время на 60%), других — нет (Sun Microsystems упали на 94%). Было бы нечестно отрицать, что я рассчитывал на большее, но не я один. Они стоят больше, чем обошлись мне, поэтому особенно жаловаться не на что.

Я писал и продолжаю писать книги для Microsoft Press и статьи для MSDN Magazine, их журнала для разработчиков. Как и моя преподава-

¹ Не знаю автора ни единого из них, но охотно укажу имена в переизданиях этой книги, если кто-либо представит мне доказательства (а не просто претензии) авторства.

тельская деятельность в Гарварде, эта работа не приносит мне особого дохода, но дает мне известность. При этом то обстоятельство, что я не являюсь служащим Microsoft, обеспечивает мне особый тип доверия. Клиенты часто привлекают меня к работе потому, что чувствуют мою достаточную близость к Microsoft, дающую мне знание действительного положения вещей, и в то же время известную независимость, позволяющую мне говорить им правду, а не вести официальную пропаганду, как делают все крупные корпорации. Я рассчитываю, что по мере того как Microsoft становится, как и должна, все более мощной корпорацией, такая линия консультирования будет развиваться.

В самой Microsoft у меня репутация надоедливого человека. Поскольку я не являюсь непосредственно сотрудником, то могу говорить то, что думаю, могу заявлять, что король голый, тогда как служащие, стремящиеся к карьерному росту, этого делать не могут. И не всегда ко мне прислушиваются. Вот несколько примеров из моих книг, изданных Microsoft Press:

«Вспомните, Windows 1 и 2 не были никому нужны лет пять. Microsoft чуть ли не вкладывала их в коробки с попкорном Cracker Jack, и все равно Windows никого особо не интересовала, потому что в ней не было полезных, удобных и экономящих время функций... Windows 3.x [была] неплохой многозадачной системой для MS-DOS, и к тому же в ней был Solitaire, ну а дальнейшее всем известно». (David S.Platt «Understanding COM+», 1999, р. 3.)

«Я хочу, чтобы в Оксфордский словарь английского языка включили слово, которое я придумал. Это MINFU, напоминающее акронимы SNAFU и FUBAR, которые переключались из военной лексики в обычный язык. В приличном обществе MINFU означает Microsoft Nomenclature Foul-Ur, т. е. терминологическая неразбериха Microsoft (в Microsoft предостаточно неразберихи). Например, активацию по месту внедренного объекта называли «визуальным редактированием» (видимо, чтобы отличить от тактильного редактирования или обонятельного редактирования), и это MINFU. ... Хорошо бы мне удалось ввести его в руководство по стилю Microsoft Press». (David S.Platt «Introducing Microsoft .NET». Third Edition, 2003, р. 127.¹)

¹ Дэвид С. Платт «Знакомство с Microsoft .NET». — Пер. с англ. — Русская Редакция, 2001.

«Проект [этой программы] был разработан в конце 1990-х годов, в эйфорические времена золотой лихорадки Сети, когда индекс NASDAQ перевалил отметку 3000 и двигался к 5000. Взбесившиеся от собственного крика рекламщики стояли на головах и вопили: «УРА! ВАШ СОТОВЫЙ ТЕЛЕФОН БУДЕТ ПОЛУЧАТЬ РЕЗУЛЬТАТЫ СПОРТИВНЫХ МАТЧЕЙ И СВЕЖИЕ БИРЖЕВЫЕ КУРСЫ!» И никто не задал вопрос: а зачем это нужно, кроме как для удовлетворения недолгого любопытства нескольких сексуально разочарованных гиков, которым нечем больше скрасить субботнее одиночество? И никто не поинтересовался, как можно заработать на этом. В представлении большинства веб-контент должен был быть бесплатным — наивная идеология, которую сейчас презрительно называют «дот-коммунизмом»... Гигантской кросс-платформенной оргии не случилось несмотря на чувствительную смазку тоннами змеинового масла, предлагаемого Microsoft, а NASDAQ сейчас упал до 2000». («The Microsoft Platform Ahead», 2004, pp. 67–68.¹)

Теперь вы знаете, в каких отношениях я с Microsoft, и можете придать им тот вес, который сочтете нужным, когда прочитаете все, что я написал о Microsoft в этой главе и в оставшейся части книги. Я мягче к ним, чем иные критики, но вы уже видели, как я обрушился на них за конкретные технические ошибки; вспомните введение, где я обвинил их в пустой трате 27 человеческих жизней в течение каждого дня в силу всего одной отстойной функции. Не сомневаюсь, что одни читатели назовут меня тайным агентом Microsoft, а другие — злобным мстительным ублюдком, но это участь любого писателя, который называет вещи своими именами, всегда и везде. Как повар, который готовит чили для целой кучи людей, я рассчитываю, что примерно половина сочтет мое отношение к Microsoft слишком острым и столько же — слишком мягким.

В: Ты уже попробовал фирменные презервативы Microsoft?

О: Да, но сначала пришлось загрузить заплатку.

¹ Дэвид С. Платт «Платформа Microsoft 2005». — Пер. с англ. — Русская Редакция, 2004.

Что было раньше

Вспоминаю, как в 1992 году я летел на очередную конференцию по программированию. В то время Windows 3.1 была новинкой, я энергично готовился к своему дебюту преподавания этой системы в Гарварде, и только-только началось обсуждение Windows NT (первая попытка представить операционную систему индустриального уровня, которая планировалась к поставке следующим летом). На мне была футболка Microsoft с предыдущей конференции, что соединяло в себе замечательные качества а) удобства и б) свободы. «Мои поздравления, Microsoft, — сказал мне сосед. — Ваша рыночная капитализация только что превысила General Motors». — «Я не работаю на Microsoft, — возразил я, — но разрешаю вам купить мне пиво за их счет». — «С удовольствием! Здесь, в первом классе, оно все равно бесплатное. Согласен даже на два пива».

Это судьбоносное событие (рыночная капитализация, а не бесплатное пиво) было отражено в газетах несколькими днями ранее. Народ был изумлен, но не потому, что компания новой экономики превзошла компанию старой экономики, а потому что компания новой экономики не владела активами в том смысле, в каком они тогда понимались. Microsoft, несомненно, приносила прибыль: она не была той дутой интернет-компанией, с которыми мы столкнемся через десять лет, оцениваемыми в миллиарды и тратящими деньги без счета.¹ Но компания, хоть и приносила прибыль, не производила осязаемых продуктов, фактически всего лишь какое-то магнитное подобие единичек и ноликов, которые простому человеку трудно отделить от случайного шума. Но если вставить дискету с этими единичками и ноликами в ту светло-коричневую коробку, которую можно купить в компьютерном магазине, нолики и единички образуют на экране определенный узор, и эту возможность трудно переоценить. Фактически общее решение инвесторов Уолл-Стрит свидетельствовало о том, что они оценивали эту возможность управлять движением электронов выше, чем обладание сборочными цехами, заполненными металлообрабатываю-

¹ В качестве классического примера приведу компанию Blue Mountain, которая рассылала по электронной почте заказные поздравительные открытки. Компания была куплена за 780 миллионов долларов в 1999 году и продана за 35 миллионов двумя годами позже.

щими станками, которые на памяти живущих поколений позволили выиграть две мировые войны.

В ту пору в компьютерном мире доброжелательно относились к Microsoft. До появления недорогих персональных компьютеров выполнять вычислительные работы было тяжело. Вычислительное устройство было размером, по меньшей мере, с холодильник, стоило более миллиона долларов и дорого обходилось в эксплуатации, обычно требовало помещения с кондиционированием воздуха, специального электропитания и штата администраторов. Современных электронных таблиц, которые сейчас есть всюду, тогда не существовало и не могло существовать. Так же как компьютерных игр и сетевой порнографии.

Это был совершенно особый мир, который не может помнить половина человечества, не достигшая лет 35. Персональные компьютеры и операционные системы для них сделали вычисления открытыми для широких масс. Больше не нужно было идти на поклон к верховному жрецу, восседавшему в стеклянном замке, чтобы напечатать документ, и можно было играть в игры, не выслушивая претензий за растрату драгоценного компьютерного времени. А если программы время от времени аварийно завершались и не располагали немислимым количеством функций, которыми я все равно не пользовался, то все равно это было неизмеримо больше, чем за год или два до того. Все было, как при появлении форда модели «Т», который открыл транспортные возможности обычным людям, и они стали путешествовать куда хотят и когда хотят, не связанные больше железнодорожными путями, расписаниями и платой за проезд.

Программист #1: *«Я слышал, что если диск Windows 95 играть в обратном направлении, получишь послание дьявола!»*

Программист #2: *«Это что! Вот если проиграть его в прямом направлении, то установится Windows 95!»*

Почему отстой продолжается

С тех пор отношение людей к Microsoft переменялось. Ее считают монополией, империей зла. Билла Гейтса изображают на футболках (полагаю, без его ведома) в виде беспощадного Борга (рис. 8.1) из *Star Trek: The New Generation* («сопротивление бесполезно! вы будете ассимилированы!»). Одну такую купили моему приятелю, когда он поступил на работу в Microsoft и переехал в Сиэтл. Менее чем за десятилетие милая от-



Рис. 8.1. Билл Гейтс в виде Борга

важная собачонка превратилась в страшную трехсоткилограммовую гориллу. Неофициальный девиз Google «Don't be evil» (Не причини зла) — это в первую очередь пощечина Microsoft.¹

Это проблема общественного восприятия, имеющая сходство с феноменом Израиля, но в ускоренном временном масштабе. Как и Израиль, Microsoft обнаружила, что быть знаменитым не столь приятно, как это кажется, когда находишься в самом низу. Ни тот, ни другая не захотели бы и не смогли бы вернуться в свое исходное состояние, но и нынешнее их положение не то, которое когда-то представлялось в розовом свете.

Это и неудивительно. Вспомните все истории взросления, которые вы видели или хотя бы слышали. Вспомните свои детские мечты о том, как здорово быть взрослым — можно ложиться спать, когда захочешь, вставить карточку в банкомат, и оттуда посыплются деньги, на которые можно купить столько сладостей, сколько пожелаешь, и весь день не есть ничего другого. А потом, когда вы становитесь (по крайней мере, номинально) взрослым, обнаруживается, что на самом деле спать необходимо, ну хоть немного, и что сначала нужно заработать деньги, а уж потом получать их из банкомата, и что за шоколадки приходится расплачиваться не

¹ Когда эта книга пошла в набор, стало известно, что Google согласился ввести цензуру поиска для пользователей из Китая. Президент Google Эрик Шмидт сообщил: «Мы все взвесили и выбрали меньшее из зол», на что последовал саркастический совет заменить девиз компании на «сотвори меньше зла».

только в магазине, но и в тренажерном зале и у зубного врача. Да, вы не хотите и не можете вернуться назад, к тому же у вас есть определенные утешения, вроде спортивных машин, секса и пива, но действительность оказалась не такой радужной, как вам раньше казалось. То же происходит сегодня с Microsoft.

Сегодня Microsoft не пользуется прежней любовью. Это их озадачивает, поскольку памятливы времена, когда они были любимцами медиа и всего мира, подбодряемые народным мнением, что компания делает лучше мир и нашу жизнь.¹ Но эти перемены неизбежны, когда кто-то выходит из детского возраста. Моя восьмимесячная дочь казалась невысказанно сообразительной, когда размазывала батат по всему лицу, чтобы хоть немного попало ей в рот (рис. 8.2), и мы радовались за нее, потому что это было ее новым достижением. Она стала выглядеть гораздо менее сообразительной, когда делала то же самое в возрасте трех лет и хотела разделить похвалу со своей младшей сестрой, которую та заслужила за то, что почти сама ела протертую морковку. (Она убьет меня, если я опубликую ту фотографию: у меня достаточно неприятностей с первым фото.) Мы терпе-



Рис. 8.2. Девочка с бататом. Мило в восемь месяцев и не очень в 36

¹ Сегодня это место занято Google, хотя возникает вопрос, не изменится ли положение после покупки двумя основателями компании широкофюзеляжного Boeing 767 для личного передвижения.

ливо сносили все странности и ошибки первых программ проверки орфографии, потому что прежде у нас такой функции не было. Но довольно скоро странности перестали быть милыми и начали раздражать, старые ошибки не исправлялись, и мы говорили: «Послушай, ты уже большая девочка. Неужели ты не можешь положить себе в рот эту картошку? Пора повзрослеть!» Microsoft не сразу это осознала, но сейчас понимание приходит.

Как и все, кто стремится вверх, Microsoft испытывает параноидальный страх перед возможной потерей своего лидирующего положения. Вы скажете, она слишком велика, чтобы это случилось? Два слова: General Motors. Бывший лидер автомобильной промышленности сейчас, по мнению многих, находится в штопоре. Еще пример? Digital. Имеется в виду Digital Equipment Corporation, старая добрая DEC. Помните такую? Одно время это была вторая по величине в мире компания по производству компьютеров после IBM. Они недооценили революцию, произведенную PC, исчезли на какое-то время, потом в 1998 году остатки были поглощены Compaq, изготовителем PC, после чего Compaq была проглочена Hewlett-Packard в 2002 году.

Microsoft опасается такой же участи. Что ж, не исключено. Ни GM, ни DEC до сего дня не знают, что их погубило (тс-с-с, скажу по секрету: товары, которые не нравились покупателям, по ценам, которые не устраивали покупателей). К чести Microsoft этот страх заставляет их прислушиваться к своим клиентам в гораздо большей степени, чем прежде.

Больше всего на свете Microsoft хотела бы избавиться от того «игрушечного» образа, который сложился у нее в начале 1990-х. У каждого есть «больные места», при касании которых человек подпрыгивает до потолка. Не буду интересоваться вашими и не стану рассказывать о своих. Но для Microsoft это восприятие ее как изготовителя игрушек. Представьте себе ярость своего сына-подростка, которому в ресторане предлагают детское меню, да помножьте ее на 40-миллиардную компанию, и вы поймете, о чем я говорю. Microsoft считает себя готовой управлять критически важной инфраструктурой мирового масштаба, вести банковские операции, а не ограничиваться текстовыми процессорами. И если это заставляет их гордиться большим пройденным путем, независимо от того, сколько им еще предстоит пройти, и иногда демонстрировать бестактность, воспринимаемую со стороны как высокомерие (снова напоминая Израиль), то ничего с этим не поделаешь.

В Интернете бродит много «магических писем», написанных в виде пародий на деловые пресс-релизы, извещающие о предстоящем приобретении Microsoft той или иной крупной организации. Часто речь идет о Католической церкви с последующим назначением Папы старшим вице-президентом Microsoft, включением Стива Бальмера в коллегию кардиналов и раздачей Microsoft индульгенций всем, кто перейдет на Office. Иногда пишут, что Microsoft собирается получить патент на нолик и единичку, чередование которых образует все компьютерные программы. «После успеха с нулем и единицей юристы Microsoft собираются на этой неделе потребовать наложения ареста на бесконечность и „пи“».¹ Мне больше всего нравится объявление о том, что Microsoft покупает Китайскую Народную Республику: «Слухи о том, что новым титулом Билла Гейтса станет «император», в данное время ничем не подтверждаются».

Куда ни кинь — везде клин

Читатели, возможно, помнят, как Министерство юстиции США преследовало Microsoft в связи с невыполнением антитрестовского законодательства. Осенью 1998 года министерство юстиции и 20 генеральных прокуроров штатов открыли дело «США против Microsoft», обвинив ее в злоупотреблении монопольным положением для вытеснения конкурентов, в частности, путем включения броузера Internet Explorer в операционную систему Windows. Весной 2000 года судья Томас Пенфилд Джексон (Thomas Penfield Jackson) постановил, что Microsoft нарушила антитрестовское законодательство, и потребовал расчленить ее на две отдельные компании, которые шутники предложили назвать «Baby Bills» по аналогии с «Baby Bells», на которые была поделена AT&T.

¹ Можете смеяться, но Microsoft действительно пытается запатентовать слово *IsNot* для использования в компьютерных языках. Можете посмотреть патентную заявку номер 20040230959 на www.uspto.gov. В данный момент заявление находится в стадии рассмотрения, и пока неясно, сможет ли Microsoft запатентовать объединенное слово *IsNot* в языке, уже содержащем находящиеся в общественном владении слова *Is* и *Not* (используемые по отдельности). Билл Клинтон, вероятно, заявил бы, что все зависит от того, чем ваше определение *IsNot* не является.

Конечно, Microsoft опротестовала приговор. Апелляционный суд подтвердил изложенные Джексоном факты, но единогласно отменил его решение и отстранил его от дела за недостойное поведение, в основном за неподобающее выступление перед прессой во время рассмотрения дела. «Заключительный раздел [постановления апелляционного суда], [озаглавленный] «Юридический проступок», делает ему разнос за идиотизм», — писала юридический комментатор Далия Литвик (Dahlia Lithwick) в Slate.com.¹

Для определения мер воздействия был назначен новый судья, и с приходом к власти в Вашингтоне новой администрации в 2001 году было достигнуто урегулирование. Формулировка окончательного решения носит весьма технический характер и касается внутреннего устройства программного обеспечения, поэтому я не стану вас утомлять ее изложением, только замечу, что ее смысл остался неясен как истцу, так и ответчику. Не стану также разбираться здесь, были ли справедливы обвинения, но я хочу, чтобы, читая следующие четыре раздела, вы помнили, что главным обвинением против Microsoft было использование ею своего доминирования на рынке для ограничения возможностей выбора у покупателей.

Примерно в это же время Sun Microsystems начала судебное преследование Microsoft по поводу разработанного Sun языка программирования под названием Java. Вокруг Java в 1996 году возник большой шум, и главным его достоинством была заявленная возможность для программистов писать программы, которые могут выполняться на любой платформе: одна и та же программа может выполняться в Windows, Macintosh и Linux, и на любом компьютере, для которого существует Java.

Лозунгом Sun для этого языка было «Один раз написал — выполняй всюду». Лично я пытался сделать это несколько раз, но должен сообщить, что написать программу, которая правильно работает на нескольких платформах, намного труднее, чем обещает Sun (так было раньше, а сейчас это особенно справедливо), если дело касается интерфейса пользова-

¹ См. «Microsoft Bad, Judge Jackson Worse» (Microsoft — плохо, судья Джексон — еще хуже), Dahlia Lithwick, помещено на Slate.com 28 июня 2001 года.

теля. Правильнее было бы сказать не «выполняй всюду», а скорее «тестируй, отлаживай и всюду обрабатывай исключения».¹

Однако с точки зрения программирования Java как язык представляет собой большой шаг вперед по сравнению с C++, языком большинства разработок для Windows в те времена. Например, Java автоматически освобождает участки памяти, которые программист перестал использовать, снимая с него обязанность убирать после себя, как в C++. Представьте себе некую невидимую прислугу, которая ходит за вами по кухне, когда вы готовите праздничный обед ко Дню Благодарения, и, не путаясь у вас под ногами, автоматически собирает и моет тарелки, кастрюли и прочий инструмент, которые вам больше не нужны (но не раньше), благодаря чему у вас всегда есть чистая посуда, и тогда вы поймете, в чем привлекательность Java как языка и почему он так востребован. Меня мало волнует заявленная поддержка многоплатформенности: мне надо было быстро писать хорошие приложения для Windows.

В ответ на это пожелание Microsoft разработала ряд инструментов Java, названных J++, которые работали под Windows. J++ предлагала две возможности: в классическом, определенном Sun варианте программисты могли писать программы на «чистом» Java и хотя бы попытаться исполнять их на других платформах, и после существенной доводки они могли заработать кое-где. Либо программист мог повернуть в J++ некий переключатель, чтобы писать программы быстрее и проще, получив возможность воспользоваться определенными функциями операционной системы Windows, которых нет на других платформах, например OLE. При этом сохранялась замечательная функция управления памятью под названием «сборка мусора» и большинство других возможностей, делающих язык Java столь приятным. Заминка заключалась в том, что написанные таким способом программы могли выполняться только под Windows, но

¹ Вот лишь один маленький пример. У мыши Macintosh всего одна кнопка, а у мыши Windows — две или больше. В программах Windows щелчок правой кнопкой обычно вызывает меню команд, доступных в точке щелчка, т. н. «контекстное меню». Java-программа, обязанная работать на обеих платформах, должна знать, на которой из них она выполняется, и следовать соглашениям по пользовательскому интерфейсу, принятым на соответствующем типе компьютера. В ином случае программа на Java будет вести себя иначе, чем другие приложения для этой платформы, что неприемлемо для пользователей.

не на других платформах, поскольку тех элементов Windows, от которых они становились зависимыми, на других платформах не было. Если я решаю, что ускоренное завершение моей программы в версии для Windows принесет мне больше денег, я могу выбрать решение, предложенное Microsoft. Если же я решаю, что мне выгоднее дольше возиться с версией для Windows, но зато охватить рынки Macintosh и UNIX, можно было пойти путем Sun.

Sun не понравилось, что у меня есть такой выбор. Они возопили, что это нечестно, и потребовали через суд, чтобы Microsoft прекратила это, как они сказали, «загрязнение» Java. Как бы оно ни называлось, но это был выбор, которого раньше программисты не имели. Sun заявила, что мы, программисты, не понимаем, как J++ влияет на наши приложения. Мы очень хорошо это понимали. Мы гики: мы же умные, верно? Мы осмысленно сделали свой выбор, потому что успех большинства программных проектов зависит от того, чтобы выйти на рынок раньше своих конкурентов. А программа, написанная на J++ специально для Windows, выйдет на рынок раньше, чем написанная на J++ для всех платформ, и гораздо быстрее той, которая написана на C++.

Как обычно бывает с исками, и в этом случае было достигнуто соглашение. В 2001 году Microsoft заплатила несколько миллионов долларов (зарплата Дженнифер Гейтс за одну-две недели и, вероятно, меньше, чем Sun потратила на юристов) и согласилась больше не использовать слово «J». Microsoft после этого создала собственный Java-подобный язык, назвав его C# («си-шарп»; выглядит как два знака ++ один на другом, наклонившиеся вперед, как бы в движении), который позволяет строить только Windows-приложения. С ним я и провожу в основном то время, когда занимаюсь программированием, и нахожу, что он превосходит «кофейный» язык. Теперь, если я упоминаю java, то имею в виду ту штуку, которую я пью, чтобы не заснуть долгими вечерами, когда программирую на C#. По крайней мере до тех пор, пока Sun не потребует в судебном порядке, чтобы я назвал свой кофеинизированный напиток как-то иначе.

Нельзя не испытать здесь хоть каплю сочувствия Microsoft. С одной стороны, министерство юстиции грозит ей вивисекцией за якобы непредоставление достаточного выбора. С другой стороны, примерно в то же время Sun колотит ее за якобы предоставление чрезмерного выбора. И оба судебных разбирательства стоят им значительных трат времени и денег. Как говорится, «куда ни кинь — везде клин».

Человек из службы технической поддержки Microsoft призван в армию и проходит курс молодого бойца. На огневом рубеже он делает несколько выстрелов и каждый раз промахивается. «Ну и мазила, — кричит инструктор, давая ему под ребра, — да ты и в слона бы не попал!» Человек из технической поддержки Microsoft кладет палец на мушку винтовки и другой рукой нажимает спусковой крючок — бах! Кончик пальца отстрелен. «Так, здесь все работает, — говорит он, показывая окровавленный обрубок. — Должно быть, проблема на другом конце».

Мы любим их ненавидеть

Хороший козел отпущения нужен всегда. Нам всем необходим кто-то, на кого можно взвалить вину за последнюю непогоду, или пониженное либидо супруга, или новое ПО контроля за серфингом на рабочем месте, блокирующее наши любимые э-э-э... развлекательные сайты. В своей классической книге «Путешествие с Чарли» Джон Стейнбек пишет о встрече в Миннесоте с человеком, рассуждающим о том, как местный козел отпущения заменился иностранным. «А помните, было время, когда все валили на мистера Рузвельта? Мой сосед Энди Ларсен просто на стену лез — такой-сякой Рузвельт! — когда у него куры заболели крупом. Да, сэр! — Он оживлялся все больше и больше. — Этим русским нелегко приходится. Поссорился человек с женой и опять же клянет русских». На что Стейнбек отвечает: «Может быть, русские всем нужны? Даже в самой России. Только там их называют американцами!»

Были бы шутки этой главы такими же смешными, если бы вы загружали патч для фирменных кондомов Oracle или тот малый, который отстрелил себе кончик пальца, оказался механиком Chevrolet? Нет. Это свидетельствует о наших особых отношениях с Microsoft. Когда я, читая лекцию, хочу вызвать смех, то рассказываю шутку о Microsoft, и она всегда имеет успех, если только я сам не испортил ее, как анекдот про тещу.¹

Windows вездесуща, и это сильно повышает вероятность неприятного столкновения с Microsoft, чего не скажешь о других гигантских компаниях

¹ Помните, кого мы ругали до Microsoft, где-то в конце 80-х? По-моему, это были японцы, которые купили Рокфеллеровский центр на деньги, полученные от продажи огромного количества импортных машин. А до того это были большие нефтяные компании, поднявшие цену бензина выше доллара за галлон.

типа ExxonMobil или Citibank. Даже если программа, авария которой уничтожила результаты наших двухнедельных трудов, была написана не Microsoft, наше раздражение все равно обращено против Windows, и мы склонны свалить вину на Microsoft, даже если она ни при чем.

Кое-кто впадает в крайности, и ненависть к Microsoft делается стержнем его личности, если это слово к нему применимо. Однажды я вместе с двумя другими известными авторами книг о программировании систем Microsoft проводил презентацию в книжном магазине в Сан-Хосе, Калифорния. Для Microsoft эта область Кремниевой долины служит входом в логово зверя: Sun Microsystems размещается рядом в Лос-Гатосе, Apple Computer — по дороге в Купертино, то, что осталось от Netscape, находится немного севернее в Маунтин-Вью, Oracle — в Редвуд-Шорз, тут же рядом Yahoo! и Google. Мы решили, что если наш книжный тур пройдет в таком месте, то он пройдет и в любом другом, и нам удалось собрать достаточно народа в Barnes & Noble. Когда начались вопросы, какой-то малый, очевидно ослепленный ненавистью к Microsoft и любовью к собственному голосу, произнес нечто вроде следующего (передаю своими словами, поскольку утратил точный текст): «Мой вопрос следующий: поскольку Microsoft — это империя зла, то когда злобная монополия занимается такими монополистическими злобными вещами, как Microsoft, эта злобная монополия, поскольку она монополия и злобна...» Мне пришлось прервать его словами: «Где здесь вопрос? До сих пор это было выступлением, а выступать здесь буду я, спасибо. Следующий вопрос — пожалуйста, вы, сэр».

Комментатор «Boston Globe» Гайавата Брей (Hiawatha Bray), чью точку зрения я всегда считал взвешенной, писал об этой тенденции в 2004 году:

«Есть в Microsoft Corp. что-то такое, из-за чего умные люди глупеют... Аура высокомерной абсолютной власти, исходящая от крупнейшего в мире производителя программного обеспечения, ... приводит некоторых людей в состояние безрассудной ярости, вызывает стремление навредить Microsoft, какой бы ущерб это ни нанесло клиентам индустрии ПО».

В качестве примера он приводит попытку антимонопольной комиссии Европейского союза заставить Microsoft убрать из Windows их музыкальный и видеоплеер. Мне не вполне понятно, каким образом лишение операционной системы возможности воспроизводить музыку может пойти на пользу потребителям, но оставим пока это в стороне. Microsoft предложи-

ла компромисс, согласно которому Media Player остается, но Microsoft одновременно будет устанавливать главного конкурента — Real Player — и еще один плеер, который будет назван позднее.¹

Брэй продолжает: «Любой правительственный чиновник, который просто хотел бы блага многочисленным пользователям Microsoft, должен был вскочить с места и расцеловать президента Microsoft Стива Балмера в обе щеки. Вам нужна реальная конкуренция среди медиаплееров? Получите! Но ЕС отклонил компромисс, возможно, по той же причине, по какой многие другие с презрением относились к компромиссу с Microsoft. Им не нужна сделка, которая всего лишь будет полезна для публики и даже некоторых других компаний-производителей ПО. Они хотят навредить Microsoft, унижить ее, чтобы эта самодовольная и всезнающая ухмылка сошла с лица миллиардера Билла Гейтса».

Я долгие годы повторял, что Microsoft должна решать не только технические задачи, но и задачи связей с общественностью. Второе у них получается значительно хуже, чем первое, — видимо, из-за недооценки его важности. Например, в 2001 году Microsoft объявила о создании службы, которая предлагала единственную регистрацию из любой точки мира, одно хорошо защищенное хранилище для личных данных пользователя, таких как расписания контактов и номера кредитных карточек, и доступ к этим данным из любой точки света с использованием любой платформы или устройства, способного работать с Web. Одни из предлагаемых функций были удобны, другие — не очень, и успех мероприятия не представлялся бесспорным. Однако все технические достоинства проекта оказались перечеркнуты невероятно глупым названием, данным ему Microsoft, — «Nailstorm» (буря с градом). Это было всего лишь внутреннее название проекта, но оно быстро стало известно публике, породило всевозможные ассоциации с «империей зла» и вызвало бурю критики, аналогичную той, которая возникла, когда ФБР назвало свою программу просмотра электронной почты «Carnivore» (хищник). Microsoft отключила больного от систем жизнеобеспечения примерно годом позже. Назови они этот сервис как-нибудь вроде «Всегда к вашим услугам», и все могло обернуться совсем иначе, и пресса, возможно, не стала бы нагнетать обстановку, так же как ФБР могла назвать свою программу, скажем, «Верный защитник».

¹ Виноват, но это правда. Планировался третий плеер для установки, но его выбор должен был быть сделан поставщиком системы позднее.

Гики судят друг о друге по качеству мозгов, и им редко приходит в голову, что остальные люди могут мерить все своим аршином. Никого не должно удивлять, что общественное поведение компании, основанной и управляемой гиками, оставляет желать лучшего, но Microsoft дошла до того, что ее неумение вести себя в обществе вредит коммерческим результатам. «Но мы же хорошие, — говорят они мне. — Посмотрите, сколько денег мы только что дали в помощь жертвам катастрофы. Почему вы решили, что нас не любят? Конечно, любят. Нас не любит лишь горстка завистливых дураков. В конце концов, мы сами себе нравимся, а ведь все остальные — такие же, как мы, верно?»

Ну сколько можно твердить, Microsoft? Ваш. Пользователь. Это. Не. Вы. Никакие фокусы не сделают плохой продукт хорошим или хотя бы посредственным. Но даже от самого технически правильного продукта в мире не будет никакого толка, если вы не понимаете своих пользователей, если вы решили не ту проблему или решили ту, но так, что ваши пользователи пожалели об этом. Неважно, каким вам представляется образ мыслей ваших пользователей или каким вы хотите, чтоб он был. Нужно соотносить свои действия с тем, что пользователи думают на самом деле. Нельзя продолжать существование, как часто отмечалось, на пяти квадратных милях Редмонда, оторвавшись от реальности. Перемены можно осуществить, если признать их необходимость. Взгляните на Johnson & Johnson, остающуюся одной из самых уважаемых компаний в мире несмотря на смерть семерых человек в 1982 году, когда неизвестный преступник подменил их обезболивающее лекарство тилол на аптечной полке. Программы Microsoft пока еще никого не убили, по крайней мере, мне неизвестны такие случаи. Если J&J может справиться с трудностями, несмотря на ряд случайных и ужасных смертей, сможете и вы, встретившись с гораздо меньшими неприятностями. Правда, для этого требуется повзрослеть, что всегда болезненно.

Вот вам реальные инструкции. Запустите Сапера (Minesweeper) из меню Пуск (Start) и дождитесь, пока на экране появится его окно. Затем тремя пальцами левой руки нажмите одновременно левые клавиши Ctrl, Alt и Shift. Удерживая их, правой рукой наберите пять символов XYZZY. Отпустите все клавиши, нажатые левой рукой. Теперь нажмите и отпустите одну левую клавишу Shift. Вы включили скрытый миноискатель. Передвиньте курсор мыши на любую клеточку поля игры, закрытую или открытую. Если в этой клеточке нет

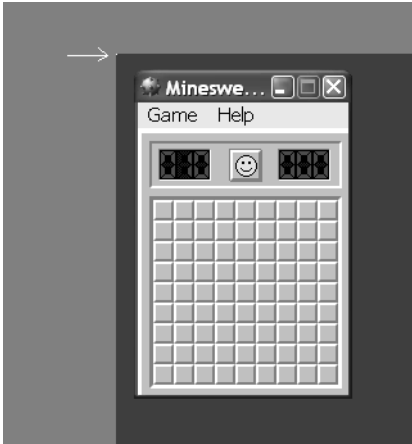


Рис. 8.3. Включив этот миноискатель, вы можете больше никогда не играть в эту игру

мины, в левом верхнем углу экрана (не окна Сапера, а всего рабочего стола — возможно, придется переместить другие окна, чтобы увидеть его) появится белая точка (рис. 8.3). Если переместить экран на клеточку с миной, точка вернет свой темный цвет. В чем соль этого абзаца? Ну, я просто сломал Сапера для вашего блага. Теперь, когда вы можете выигрывать каждый раз, нет смысла играть, правда? Я просто раз и навсегда повысил вашу производительность. Не стоит благодарности. Бесплатная услуга. И не злите меня, а то я сделаю достоянием гласности секретные коды в Солитере и испорчу вам жизнь.

Plus ça change¹

Все, кто критикуют Microsoft или хотят оказать на нее влияние, должны учитывать ограничения, которыми она связана. Ведь считали вы когда-то всемогущими своих родителей, как мои дочери (шести и трех лет) считают в данный момент меня (дай Бог им счастья), не замечая ограничений. Представьте себе Соединенные Штаты, которые часто называют

¹ Это первая часть французской поговорки: plus ça change plus c'est la même chose — чем больше перемен, тем больше все остается по-старому.

единственной сверхдержавой в мире, пытающимися прихлопнуть целый рой противных ос. Мы могли бы их раздавить, если бы они немного посидели спокойно, но они это знают и потому все время двигаются. Когда вы возмущаетесь — «почему эти [ругательство] в Microsoft делают [это] и не прекратят делать [это]», — вы должны понимать, что они трудятся при наличии, по крайней мере, четырех ограничений, которые не сразу заметны случайному наблюдателю.

Во-первых, выпуск ими новых продуктов, и особенно модификаций имеющихся, весьма сдерживается, как это ни странно, самими пользователями. Очень сложно выпустить новый продукт, который нарушает их привычки. «Как Господу Богу удалось создать мир всего за шесть дней? — говорится в старом гиковском анекдоте. — Он это смог потому, что у него не было предыдущих инсталляций, о совместимости с которыми нужно было заботиться».

Возьмите, например, старую функцию под названием DDE (динамический обмен данными). Появившись в программе электронных таблиц Microsoft Excel примерно в 1988 году, она стала первой технологией, позволявшей данным автоматически направляться из одного приложения Windows в другое без ручной работы человека. В момент своего появления она стала величайшим изобретением после резаного хлеба, и я помню, как демонстрацией ее я заманивал клиента на крупный проект. Примерно в 1993 году ее вытеснила технология под названием OLE (очередной TLA, см. главу 7), более надежная и простая для правильного программирования, когда с ней разберешься. Но к этому времени огромное количество пользователей, особенно в финансовом секторе, написало электронные таблицы Excel, зависящие от DDE, и Microsoft не могла просто отказаться от нее. Пользователи сочли бы версию без этой функции не обновлением, а откатом назад, и можно понять, что за откат назад им платить не хотелось. До сего дня оператор финансового рынка Bloomberg продолжает предоставлять в сети биржевые курсы в этом формате наряду с новыми. Когда Microsoft добавляет в Excel новые функции, или укрепляет ее защиту, или как-то изменяет внутреннее устройство, ей приходится следить, чтобы не нарушилась работа DDE. Один короткий миг неконтролируемого восхищения отличной функцией, а результат приходится тащить еще лет 20, а то и больше (напрашивается аналогия с сексом). Помножьте это на количество других устаревших функций и вы поймете, что все это обходится чертовски дорого.

Второе ограничение — это невероятное разнообразие программных и аппаратных средств, в которых приходится выполняться Windows и Windows-приложениям. Различия между компьютерами, работающими под управлением Windows, гораздо многочисленнее, чем вам может показаться: скорость процессора, размер памяти, установленное оборудование (принтеры, мыши, сети или звуковые карты и т. д.), а также другие программы, выполняемые в то же самое время. Как нет двух одинаковых снежинок, так и среди сотен миллионов компьютеров не найдется двух одинаковых. Apple ограничила для себя это разнообразие, тщательно контролируя аппаратную часть Макинтошей, и поэтому они надежнее. Но пользователи не вознаградили Apple за такую надежность покупкой большого их количества. Чтобы они там ни говорили, потребители проголосовали своими кошельками за низкие цены и предположительно большую гибкость Windows в соотношении 20:1.

Как бы тщательно ни тестировал производитель свои продукты, установка его программы на ваш РС почти наверняка окажется первым случаем в истории, когда эта программа очутилась именно в такой вычислительной среде. И то, что чаще всего при этом программы так или иначе работают, является просто чудом. В среде Windows крайне трудно воспроизводить программные ошибки. Теперь, когда в Windows появилось средство регистрации аварийных ситуаций, о котором я расскажу ниже, мы можем точно узнать, чем занималось приложение, когда ему настал конец. Влияние этого ограничения уменьшается.

В-третьих, это разнообразие и сопутствующие проблемы применимы и к поголовью пользователей. То, что хорошо, правильно, необходимо для одного класса пользователей, может быть совершенно неприменимо для другого класса. Возьмите инвестиционный фонд Vanguard, который управляет вкладами на сумму почти триллион долларов, некоторая ничтожная доля которой принадлежит мне. На их сети совершается огромное количество атак (в одной финансовой фирме мне сообщили, что на главный сайт их компании совершается более сотни атак в секунду), и убытки от несанкционированного проникновения могут быть огромны. Потребность Vanguard в защите колоссальна, и они это понимают, тратя большие средства на решение этой проблемы. Например, сотрудникам, работающим на компьютерах, обычно не разрешается самостоятельно устанавливать программы или получать почту от внешних провайдеров типа Yahoo! и Hotmail. Целая армия профессионалов занимается опреде-

лением того, какие приложения безопасны, и установкой их тем, кто в них нуждается. В учреждениях такого типа косо смотрят на Solitaire, а об играх в Star Trek можете забыть. Все эти меры безопасности затрудняют работу на их компьютерах, но они вынуждены идти на них исходя из специфики бизнеса. Поскольку денежное вознаграждение высоко, то и «бюджет неудобств» пользователей (см. главу 3) гораздо больше. Они терпят гораздо большие неудобства, чтобы избежать неприятностей, поскольку вынуждены это делать.

Напротив, обычные пользователи не испытывают и малой доли такого количества атак, а цена взлома системы безопасности несравнимо меньше. Компьютеры используются для более разнообразных задач, нежели только офисные приложения. Пользователи не могут, да и не захотят прилагать больших усилий для обеспечения своей безопасности. Баланс между удобством и безопасностью смещается в сторону первого по сравнению с тем, что разумно в инвестиционной компании. Бюджет неудобств домашнего пользователя гораздо меньше. Windows XP вынуждена быть достаточно надежной для использования в инвестиционной компании и в то же время достаточно простой для установки и работы в домашних условиях. Очень трудно соблюсти здесь правильное равновесие, и я иногда сомневаюсь, что один и тот же продукт сможет долго продержаться, обслуживая оба рынка.

Наконец, следует помнить, что Windows стоит очень дешево, а это значит, что на существенную поддержку рассчитывать не приходится. Цена Windows XP Home составляет около \$90, если покупать ее отдельно на Amazon.com. Большинство пользователей получает свою операционную систему уже установленной при покупке PC, и тогда она обходится еще дешевле. Dell продает сейчас домашний компьютер с приличными возможностями (достаточно памяти, разумного размера жесткий диск, сетевой адаптер, звуковая карта, некоторое программное обеспечение и даже 17-дюймовый монитор — вполне рабочая вещь) и установленной Windows XP Home за \$299. За такие деньги существенной поддержки не получишь.

Три женщины рассказывают друг другу, как их мужья ведут себя в постели. Первая говорит: «Мой муж оперный певец, он всегда поет мне серенады. Это замечательно.» Вторая говорит: «Мой муж полицейский, и ему нравится, когда на мне надеты наручники. Я к ним уже привыкла. А у тебя?» Третья дама немного краснеет и говорит:

«Мой муж работает в Microsoft, и мы ничем таким не занимаемся. Он просто сидит на краешке кровати и показывает мне презентацию PowerPoint, рассказывая, как будет замечательно, когда у него все за-работает.»

Трудности роста

Вечные поиски бесплатной закуски и выпивки недавно привели меня на ужин, который давало подразделение Microsoft, консультировавшее медицинские фирмы. Атмосфера показалась мне несколько странной, неуютной, враждебной, и я не мог понять почему. Это не было легким товарищеским общением ведущих программистов или докладчиков на Tech Ed, людей, которых я изучил вдоль и поперек за истекшие годы, читавших книги и статьи друг друга и имеющих много общих знакомых. Я несколько раз попробовал завязать обычные гиковские разговоры («Вы не знаете, какие изменения внесли в алгоритмы сборки мусора при переходе от версии 1.1 к версии 2?»), но отсутствующий взгляд, который я видел в ответ, подсказывал мне, что мы говорим на разных языках.

«Платт, это Новый Microsoft, — сказал мне товарищ, пригласивший меня на это сборище, терапевт-рenegат, которого я когда-то давно научил программировать.¹ — Мы с тобой тут последние из могикан. Глянь, тут собрались специалисты по организации производственного процесса, специалисты по законодательству, клиницисты, торговые агенты, но фактически нет программистов, исключая того малого, который сидит в углу и пускает слезу в свое пиво, что-то бормоча себе под нос. Мы такие же динозавры, как те, что на рекламных щитах, не обновившие свои версии Office. Черт, и как это только получилось?»

Microsoft изначально была компанией, ориентированной на продукт, и в значительной мере остается ею. Я имею в виду, что программисты компании разрабатывают операционную систему, затем кидают ее через забор пользователям, а сами начинают делать новую — как акушер, принявший роды и передавший младенца родителям. Разница в том, что младенец был создан чистой работой мысли акушера и непохож ни на что, ра-

¹ Возможно, он станет моим соавтором, когда я возьмусь за книгу «Почему медицина — отстой», закончив продолжение данной — «Почему программы так и остались отстоем».

нее существовавшее на свете. Новые родители не располагают выработанными за миллион лет эволюции инстинктами обращения с новой операционной системой, а спрашивать у своих родителей, как они поступали в подобной ситуации (читали руководство и усиленно молились), вероятно, бесполезно. Лучше всех знает младенца акушер, но он вернулся в свою мастерскую, где творит нового младенца, и в голове у него множество идей, не воплощенных в предыдущем из-за нехватки времени, и судьба старого его теперь мало интересует, особенно если он собирается через пару лет, по мере готовности, заменить его новым. У сотрудников, вступающих в контакт с клиентами, которым достался новорожденный, «технических евангелистов», как называют их в Microsoft, очень мало влияния на производственные бригады. Ситуация начинает меняться с осознанием Microsoft того, что а) нельзя просто выкидывать на рынок новый продукт и надеяться, что его купят, и б) текущая поддержка и консультативные услуги могут приносить большую прибыль, как показывает опыт IBM.

Microsoft основана в 1975 году. Им было лет десять, когда они выпустили первую версию Windows, 15 лет, когда начался подъем Windows, немногим больше 30 сейчас, когда я пишу эту книгу. Когда вам было 15, вы хотели заниматься только тем, что круто и интересно. Вы мало думали о своей ответственности, разве что проявляли (или изображали) какое-то минимальное ее количество, чтобы получить ключи от семейного автомобиля (тоже круто и интересно). Не стоит удивляться, что в возрасте 15 лет Microsoft ценила программистскую работу, которая казалась ей крутой и интересной, чему способствовала возможность получать за нее кучу денег, а с такими деньгами можно было заниматься еще более крутыми и интересными вещами. Вот так мы получаем всякие пасхальные яйца¹ в Excel.

В 1990 году операционные системы Microsoft не управляли больницами или воздушным движением, чего и нельзя было бы доверить подросткам. Они заявляли, что им это под силу, как и свойственно подросткам, но, описывая те дни, один нынешний служащий Microsoft говорит: «Мы страдали иллюзиями своей адекватности». Лишь сейчас Microsoft набира-

¹ «Пасхальное яйцо» — скрытый участок программы, помещаемый в нее программистами ради забавы. Специальная комбинация клавиш для «сапера», описанная мною ранее в этой главе, относится к разряду пасхальных яиц. Особенно славилась искусными пасхальными яйцами Excel.

ет силу в таких проектах, которыми должны заниматься 30-летние. Вся эта область производства повзрослела, должна была повзростеть. Как писал Стив Балмер в письме 2002 года: «...еще пять лет назад мы были склонны думать о себе как о маленькой начинающей компании, которой не так давно были. Сегодня мы понимаем, что наши решения оказывают влияние на многие другие технологические компании. Мы играем важную роль лидера в своей отрасли и вынуждены играть по новым правилам — как в юридическом плане, так и в силу господствующих в отрасли тенденций». Требуется время, не обходится без фальстартов и тупиков, но они движутся в этом направлении, а не в обратном. Сегодня Microsoft берет на работу больше юристов и меньше гиков. Они более ответственны и менее интересны — как котята, ставшие взрослыми кошками. Так и должно быть, если они собираются управлять миром, — вы же не хотите, чтобы 15-летний смешливый мальчишка командовал атомной подводной лодкой? («А что будет, если нажать эту кнопку? Ой!») Но часто, пытаясь бороться с их внутренней бюрократией по поводу того или иного загадочного правила, я любовно вспоминаю прежние времена стрельбы с бедра, «внимание-пли-целься», как с любовью вспоминаю свои юные годы (не все подряд, конечно).

Google сейчас переживает свои юношеские годы, как когда-то Microsoft. Google живет интереснее, отчасти потому, что у них меньше ответственности. Разумеется, я рекомендую Google и сам пользуюсь им постоянно, но если завтра его не станет, я это переживу: Yahoo!, MSN Search, Alta Vista или Lycos более или менее справятся с моими потребностями в поиске. А вот если исчезнет Microsoft, мне будет гораздо сложнее.

Один пастух пасет свое стадо, когда на луг въезжает Range Rover со спутниковой тарелкой на крыше и останавливается рядом с ним. Из него выскакивает яппи, одетый в деловой костюм, с планшетным ПК в руках. «Скажи мне, добрый человек, — заявляет прибывший, — если я посчитаю, сколько у тебя овец, ты дашь мне одну из них?» Пастух кивает, выражая согласие. Человек царапает что-то на экране ПК, составляя программу, которая в реальном времени получает спутниковые фотографии пастбища, считает тени, отброшенные ногами овец, и делит результат на четыре. «У тебя ровно 197 овец, включая ту трехногую, правильно?» Пастух снова кивает, поэтому человек выбирает себе овцу и тащит ее к своему джипу. Тут пастух произносит первые слова: «Ты ведь работаешь в Microsoft, верно?» —

«Да, а как ты узнал?» — «Ты явился непрошеным и взял с меня плату за ответ, который я знал, на вопрос, который я не задавал. И ты ни хрена не понимаешь в моем бизнесе». — «Что!?! — в гневе кричит яппи, багровея. — Microsoft — мировой лидер в программном обеспечении. Наши программы установлены на 97% ПК и половине серверов. Наш босс — самый богатый человек в мире. И какой-то неотесанный мужлан будет говорить мне, у которого две докторских степени, что мы не разбираемся в твоём бизнесе?» — «Отпусти собаку, которую ты схватил, — отвечает пастух, и я тебе все объясню.»

Что мы можем сделать

Поставщики не улучшат качество программ, пока пользователи не потребуют этого от них. «Когда народ говорит нам о чем-то, что это плохо, мы начинаем делать его лучше, — недавно сообщил мне один евангелист Microsoft, с которым я обсуждал эту книгу за завтраком в Исландии. — Для нас выяснить, что плохо, а что нет, гораздо труднее, чем можно себе представить».

Новоорлеанская газета «New Orleans Times-Picayune», требуя (после урагана «Катрина») строить дамбы категории 5, посоветовала читателям писать письма своим членам законодательного собрания. «Утопите их в письмах, как Катрина утопила нас», — написала газета. То же самое пользователи должны сделать с Microsoft и всеми другими поставщиками ПО. Кроме того, нужно быть максимально конкретным. Простые крики «меня тошнит» — какое бы облегчение они вам ни приносили — не дают поставщику никаких сведений относительно того, что именно вам не нравится. Постарайтесь успокоиться. «Нам нравится это. Нам не нравится то. Нам очень нужно то-то и то-то, но вот эта штука сильно нам мешает». И не удивляйтесь, если они прислушаются к вам не сразу, а то и вовсе не обратят внимания. Количество пользователей таких массовых программ, как скажем Excel, огромно, и очень трудно уравновесить требования разных типов пользователей. Вам может казаться, что некая функция совершенно бессмысленна и даже вредна, но кому-то другому она может быть нужна постоянно. Вечный парадокс, конечно, состоит в том, что производителю программы трудно узнать, что думает «молчаливое большинство». Оно молчит по определению.

Поток писем от пользователей может привести к нужным результатам, и вот тому пример. В первую неделю 2006 года некто опубликовал сообщение о найденной им уязвимости в Internet Explorer, названной «уязвимостью метафайлов Windows». Microsoft изготовила для нее заплатку, но не стала выпускать, решив получить ее протестировать и включить в плановое обновление, которое должно было появиться неделей позже. Microsoft не сочла эту уязвимость критической, поскольку, как она заявила, большинство антивирусных программ отлавливало зараженные файлы. Сообщество пользователей с этим не согласилось. В посвященных безопасности форумах и списках рассылки поднялась волна критики. Разработчик из России сделал заплатку, которая должна была заткнуть брешь и которую одобрили несколько авторитетов в области безопасности и рекомендовали для немедленной установки пользователями, не дожидаясь заплатки от Microsoft. Широкая огласка и опасение, что ей придется поддерживать браузер с заплаткой стороннего разработчика, заставили Microsoft ускорить работы, произвести интенсивное тестирование и срочно выпустить заплатку.

Дело не в том, кто прав в споре о том, достаточно ли серьезна была эта ошибка, чтобы ускорить график выпуска обновлений, а в том, что пользователи считали, что она серьезна, а Microsoft не оценила настроение пользователей. В итоге она подпортила себе репутацию — примерно, как когда-то Intel, когда выяснилось, что в процессорах Pentium есть маленькая математическая ошибка в младших десятичных разрядах, а Intel виляла и пыталась игнорировать ее, вместо того чтобы честно отнестись к своим клиентам. (*Вопрос: Что такое 665,99999999999842? Ответ: Число Сатаны на Pentium, но для обывателей это достаточно приближение.*) Все решило давление со стороны общественности. Если бы Microsoft заявила *mea culpa* а la Johnson & Johnson, все бы быстро прошло, но попытка ничего не замечать привела к преувеличенной оценке серьезности.

Хорошо, возьмите приложения не от Microsoft. Загрузите бесплатную версию WordPerfect фирмы Corel. Попробуйте поработать с веб-браузером Firefox. Я не выношу, когда он начинает терроризировать меня своими окнами с сообщениями об ошибках, но некоторые очень рекомендуют его. (У меня он не вызывает ничего, кроме ругани, но это, возможно, мое личное впечатление.) Но его наличие побудило Microsoft изменить свои планы относительно следующей версии Internet Explorer.

Хотите полностью избавиться от Microsoft? Можно. Тони Боув (Tony Bove) написал книжку «Just Say No to Microsoft: How to Ditch Microsoft and Why It's Not as Hard as You Think», No Starch Press, 2005 (Скажи «нет» Microsoft: как освободиться от Microsoft и почему это не так сложно), в которой излагает свой рецепт. В рейтинге популярности Amazon.com она занимает позицию под номером 265 413, что не свидетельствует о наличии большого количества пользователей, поставивших себе целью научиться жить без Microsoft. Это еще раз показывает (но уже с другой стороны), что большинству пользователей безразличен инструмент и интересен результат. Home Depot, возможно, считает, что продает дреды, но покупатели идут в Home Depot, потому что им нужны отверстия.

Чего никто не любит, — кроме консультантов, зарабатывающих себе этим на жизнь, — так это аварии (краха) программы, когда она без всякой видимой причины перестает работать сама, а иногда и «подвешивает» весь компьютер. По этой причине в свежие версии продуктов Microsoft включают «регистратор аварий», идея которого напоминает черные ящики, устанавливаемые на самолетах. Наверное, вы видели его в действии (рис. 8.4). Он показывает, что где-то глубоко внутри программы (или самой операционной системы) произошла ошибка, и приложение не смогло справиться с ней и продолжить работу. В Windows 3.0 и ранее вся система целиком застывала с сообщением «Неустраняемая ошибка приложе-



Рис. 8.4. Регистратор аварий запрашивает разрешение отправить отчет в Microsoft

```

A problem has been detected and windows has been shut down to prevent damage
to your computer.

The problem seems to be caused by the following file: SPCMDCON.SYS

PAGE_FAULT_IN_NONPAGED_AREA

If this is the first time you've seen this Stop error screen,
restart your computer. If this screen appears again, follow
these steps:

Check to make sure any new hardware or software is properly installed.
If this is a new installation, ask your hardware or software manufacturer
for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware
or software. Disable BIOS memory options such as caching or shadowing.
If you need to use safe Mode to remove or disable components, restart
your computer, press F8 to select Advanced Startup options, and then
select Safe Mode.

Technical information:

*** STOP: 0x00000050 (0xFD3094C2,0x00000001,0xFBFE7617,0x00000000)

*** SPCMDCON.SYS - Address FBFE7617 base at FBFE5000, DateStamp 3d6dd67c

```

Рис. 8.5. Синий экран смерти

ния», и приходилось выключать и заново включать машину. Windows NT обычно удавалось закрыть одно аварийное приложение, сохранив работу всего остального, но не всегда — помните нашего старого друга «синий экран смерти» (рис. 8.5), когда NT слетала с катушек? Даже если NT продолжала работать, обычно не оставалось никакой информации об аварии, чтобы помочь избежать ее в будущем. В отличие от них, Windows XP почти всегда сохраняет работоспособность, а регистратор аварий делает фотоснимок почившей программы и отправляет его в Microsoft для анализа. В этом отчете содержится, помимо всего прочего, точная строка программы, в которой произошла ошибка, и все уровни, через которые она прошла.

Пользу этого автоматического отчета об аварии для команды разработчиков переоценить невозможно. Когда я начинал заниматься программированием, наша линия поддержки получала звонки такого рода:

Клиент: *«По-моему, что-то случилось с моей программой».*

Я: *«А какие симптомы?»*

Клиент: *«Не знаю. Она, типа, как-то странно перестала работать».*

Я: *«А что вы делали с ней в тот момент?»*

Клиент: *«Не помню, вроде работал с ней, и все такое».*

Я: *«А раньше это случалось?»*

Клиент: «Да».

Я: «Ну, значит, оно случилось снова. Это все, что я могу вам сказать. До свидания».

Из таких сообщений мало что можно было почерпнуть, кроме того что программа работает неустойчиво, но, уверяю вас, мы и так это знали. Но мы не знали, где и как, поэтому не могли определить, почему. Не одна компания погибла из-за такой вот неспособности отладить приложения. Часто ошибки сильно зависят от временных параметров или присутствия в системе других программ — условий, которые невозможно воспроизвести за пределами машины пользователя.

Как писал Стив Балмер в открытом электронном письме: «Нас поразили закономерности, обнаружившиеся в отчетах об отказах, которые нам присылали клиенты. Отчеты указывали на ошибки, присутствовавшие не только в наших собственных программах, но и в приложениях для Windows от независимых поставщиков аппаратных и программных средств. Весьма удивительно было обнаружить, что из всех программных ошибок, связанных с отчетами, можно выделить относительно небольшую часть, которая ответственна за большинство отказов. Примерно 20% ошибок обуславливает 80% отказов и — это меня ошеломило — 1% ошибок является причиной половины отказов». Думаю, что если бы я был поставщиком ПО и Балмер стучал по моему столу, размахивая перед моим носом отчетом об ошибке с криком «идиот, да вот ведь — в этой строчке!», я дал бы повышенный приоритет ее исправлению.

Главное опасение, которое регистратор аварий вызывает у пользователей, связано с тем, что иногда в его отчете содержатся данные из документа, с которым они работали в тот момент, когда произошла авария, а эти данные могут быть конфиденциальными. В бюллетене Министерства энергетики США¹ рассказывается о такой возможности, отмечается, что данные документа попадают в отчет об аварии примерно в одном случае из трех, и рекомендуется отключить эту функцию. В заявлении о неразглашении информации, полученной от регистратора аварий, Microsoft клянется использовать эти данные исключительно в «мирных» целях. Все же там отмечается, что «служащим Microsoft, субподрядчикам и поставщикам, которым в интересах дела понадобятся данные отчета об отказе, будет предоставлен к ним доступ. Если из отчета следует связь ошибки

¹ Есть сейчас в сети по адресу www.ciac.org/ciac/bulletins/m-005.shtml.

с продуктом стороннего разработчика, Microsoft может послать данные этому разработчику, который, в свою очередь, может послать их своим субподрядчикам и партнерам». Иными словами, если вам было, что скрывать в момент, когда обрушилась программа, оно вполне может обнаружиться, поэтому при выполнении приложений с весьма секретными данными я бы отключил эту функцию.¹ Однако если вы хотите, чтобы врач облегчил ваше состояние, придется объяснить ему, где у вас болит, и, возможно, даже показать это место. И, как и ваш врач, я гарантирую, что какими бы сомнительными ни были ваши занятия, когда случилась эта неприятность, это все уже было видно тысячу раз до того и почти наверняка надоело тем, кто это видел. (Ох уж это «почти».)

Вы снова видите, как Microsoft иногда попадает в безвыходные ситуации. Если они не получают данных регистратора аварий, то не смогут ни исправить ошибки, ни даже найти их. С другой стороны, если им начнут присылать эти данные, то пользователи придут в ужас из-за возможности разглашения их конфиденциальных данных. Я надеюсь, что достаточное количество пользователей, не озабоченных секретностью, оставит функцию отчета включенной, чтобы Microsoft могла выяснить достаточно для исправления проклятой ошибки. Я оставляю у себя функцию генерации отчетов включенной, если только клиент в контракте не требует иного и не платит мне за причиненные неудобства. Все как в «Крестном отце» Марио Пьюзо, когда наркодилер Солоццо берет в заложники советника семьи Корлеоне Тома Хагена. Он говорит: «Садись в машину. Если бы я собирался тебя убить, ты был бы уже мертв».

В Интернете есть истерическое видео, очевидно снятое в европейском офисе Microsoft. Оно показывает программу Microsoft «We Share Your Pain» («мы разделяем вашу боль», или WSYP (произносится «WIZ-zip»)), расширяющую возможности регистратора аварий таким образом, что пользователь видит автора того фрагмента кода, где случилась ошибка, и получает возможность выразить свое возмущение путем прокалывания его иголкой или удара электрошоком. «Мы полагаем, что в Microsoft работают лучшие программисты. И эта программа поможет им понять чувства своих пользователей самым непосредственным образом», — гово-

¹ Откройте Control Panel и выберите System. Затем выберите закладку Advanced. В правом нижнем углу щелкните по кнопке Error Reporting. Выберите переключатель с меткой Disable Error Reporting.

рит один из персонажей. Я несколько раз находил этот фильм на сайте Microsoft, но он все время перемещается, поэтому я не могу привести в книге его точный адрес. Воспользовавшись поисковым механизмом, вы быстро его обнаружите, а я попытаюсь держать ссылку на него на сайте этой книги. Вполне возможно, что они прислушались к моим словам «познай своего пользователя».

В: Сколько человек из службы технической поддержки Microsoft нужно, чтобы заменить электрическую лампочку?

О: Боюсь, что мы не сможем воспроизвести вашу проблему. У нас все лампочки светят нормально.

В заключение

Покойный автор колонки ответов на письма читателей Энн Лэндерс (Ann Landers, 1918–2002) обычно сводила бесчисленные вопросы о мужьях и бойфрендах к простому ответу: «Решите, милочка, сами, как вам лучше, — с ним или без него». Я предлагаю вам так же ответить на вопрос относительно себя и Microsoft. Представьте себе, что можно щелкнуть пальцами, и все на свете, созданное в Microsoft, чудесным образом исчезнет. Станет ли мир от этого лучше или хуже? Не все со мной согласятся, но мне кажется, что второе. Поэтому я надеюсь, что этого не произойдет. И мои кредиторы тоже.

Лет 40 назад моя бабушка, когда я жаловался на плохое качество телефонной связи, говаривала: «Я помню время, когда телефонов вообще не было. Сейчас же ты можешь в любой момент снять трубку и позвонить мне. Думаешь, мне нравится такое качество? Но оно гораздо лучше, чем то, которое было раньше — никакое». Не помню времен, когда не было никаких компьютеров, но помню, что компьютеры были такими гигантскими штуками размером с автофургон в стеклянной будке с кондиционированным воздухом, и пользовались ими только службы социального страхования и налогов, а вычислительной мощности в них было меньше, чем в сегодняшней зубной щетке. Тридцать лет назад появились мини-компьютеры DEC размером с холодильник, по-прежнему в стеклянных будках (меньшего размера, но все равно с кондиционированием), и некоторые колледжи могли себе такие позволить, если у них было много богатых выпускников с большим сердцем. Двадцать лет назад мы получили первые персональные компьютеры в результате того, что IBM и Apple

смогли поместить вычислительную машину на рабочий стол. Десять лет назад Windows предоставила вам возможность одновременно раскладывать пасьянс и проверять электронную почту через модем, дозванивающийся по телефонной линии. Теперь у нас есть XP и Всемирная паутина, а сообщения об авариях и загрузка обновлений программ происходят автоматически. Ну разве не становится жизнь все лучше и лучше? Вспомни, где мы были и куда пришли, — вот что сказала бы мне бабушка, будь она жива и способна это понять. Придется мне вам рассказать от ее имени.

Умирает Билл Гейтс, и с учетом его пестрого прошлого св. Петр не может решить, куда его отправить — наверх или вниз. Он решает предоставить выбор между адом и раем самому Биллу. Естественно, Билл хочет осмотреть то и другое, прежде чем выбрать для себя место вечного пребывания, и посещает оба места. В раю все в порядке: крылья ангелов смотрятся хорошо, но по сравнению с тем, к чему он привык, управляя Microsoft, все скучновато. Спустившись вниз, он видит разгульные вечеринки на побережье, реки пива, горы еды, сменяющие друг друга музыкальные группы и танцовщиц, турниры по стрельбе на игровой консоли Xbox и все мыслимые удовольствия. В итоге он выбрал ад. Как только за его спиной захлопывается дверь, его охватывает адское пламя, душит запах серы, окружают визжащие черти, похожие на Жанет Рено и колющие его вилами, начинаются прочие мучения. «В чем дело?! — пишет он в электронном письме Дьяволу, пользуясь 300-бодным модемом и Commodore 64, единственным имеющимся компьютером (новая мука). — Где вечеринки? Где выпивка? Танцующие девочки? Xbox?» — «Все правильно, мистер Гейтс, — отвечает князь тьмы. — Кому, как не вам, знать разницу между демо-версией и законченным продуктом.»

ГЛАВА ДЕВЯТАЯ

С ЭТИМ НАДО ЧТО-ТО ДЕЛАТЬ



Индустрия ПО существует благодаря пользователю, то есть вам. Разработчики иногда об этом забывают, и некоторые из них никогда особо не задумываются об этом. Однако их основная задача состоит в том, чтобы писать программы, которые вам нравятся и которые вы купите, или создавать сайты, которые вам придутся по вкусу, а поэтому вы будете ими пользоваться, поощряя таким образом разработчиков и их работодателей.

Как скажет любой излечившийся от алкоголизма больной, первый шаг в лечении — признание существования самой проблемы. Для многих программистов это необходимо. («Меня зовут Боб, и я пишу поганый софт». — «Здравствуйте, Боб».) Иногда они сами приходят к этому умозаключению («Боже, неужели я в самом деле написал этот бред?»), но чаще всего им стоит в этом помочь. («Боб, вы хоть представляете, насколько идиотский сайт вы сверстали для нас? Вы заставляете пользователя подумать пять раз перед тем, как он сможет начать заниматься делом, ради которого пришел на сайт. Никто так и не смог закончить свою работу полностью с того момента, как открыли сайт, хотя пять тысяч человек попытались и остановились на полпути. Этого бы не произошло, если бы вы прочитали книгу Платта.»). Вот тут появляется вы.

Вы не программист и не хотите им быть, но у вас есть ценные — нет, *бесценные* — предложения по улучшению программного обеспечения. Разработчики нуждаются в том, чтобы вы им рассказали, что вам нужно и чего хотите, что любите и что терпеть не можете, на что надеетесь и чего опасаетесь. Они сами по себе не могут этого знать, хотя бы потому, что

думают, что они все это и так знают. Подменяя ваши ценности своими из-за незнания ваших, они обрекают себя на вечные муки, как вы уже подметили, я надеюсь.

Вышеупомянутый алкоголик вам также скажет, что второй шаг к излечению — это вера в то, что проблема решаема. Я старался в этой книге приводить как хорошие примеры (Google автоматически определит вашу страну, совершенно не требуя от вас каких-либо действий), так и плохие (UPS.com заставляет вас сообщить страну с помощью множества щелчков мышью, а потом явно указать, должны ли они запомнить ваш выбор, дабы в следующий раз не проделывать эту безрассудную процедуру), чтобы показать вам, что хорошее программное обеспечение не только может существовать, а в самом деле существует, по крайней мере иногда. Когда программисты что-то делают плохо, то почти всегда из-за того, что не понимают, чего хотят их пользователи, что им нужно и что они готовы терпеть. Эта глава о том, как обеспечить разработчикам обратную связь с пользователем — как вам рассказать им, что же такое они делают и что это нужно сделать лучше. Вот пять независимых шагов, которые вам надо сделать и которые я называю «шпаргалкой Платта».

1. *Купите*

Первый способ, которым потребители программного обеспечения или иных продуктов демонстрируют свои предпочтения, — это приобретение или не приобретение тех или иных продуктов и услуг. Автомобили не улучшались, пока потребители не начали требовать этого, покупая надежные Toyota и Honda и оставляя хрупкие Pinto и Nova ржаветь на дилерских стендах. Как поговаривал Ли Якокка в телевизионной рекламе возрождающегося в то время Chrysler: «Если вы можете найти машину лучше — купите ее». Мы так и поступили. В итоге все автомобили стали лучше.

Программное обеспечение не улучшится, пока производители не почувствуют наши предпочтения своим карманом. Google стал доминировать над Yahoo! и Microsoft в сетевом поиске, так как предложил лучший поисковый движок с более прозрачным и понятным пользовательским интерфейсом. Пользователи вознаградили Google, отдав предпочтение его движку. Эти пользователи не платили Google напрямую, но своими щелчками по интересующей их рекламе побудили это делать рекламодателей

(в том числе и меня). Масса кликов. Масса денег. Большой прибыльный Google, потому что он понял, как и что.

Деньги говорят лучше слов. Можно заявить устно, что компьютеры Макинтош надежнее, или лучше сконструированы, или просто в целом лучше, чем ПК на базе Windows. Может быть, это и так, но если копнуть поглубже, окажется, что не так уж и много людей их покупают. В целом предпочтение, отдаваемое пользователями Windows, измеряется как 20:1. Это дает Microsoft понять единственным доступным для нее способом, что она делает все более или менее верно. Если бы это соотношение заметно изменялось, вы заметили бы, что Microsoft поднажала, чтобы остаться на том же уровне. Если за вашими словами не стоят деньги, не удивляйтесь, что на них никто не обращает особого внимания.

Хороший пример конкуренции, повлиявшей на поведение даже Microsoft, — текущая война браузеров. Браузеру Microsoft Internet Explorer версии 6, который поставляется с Windows, уже больше 5 лет, то есть по меркам Интернета это просто ископаемое. Другие веб-браузеры, например Firefox (ему пресса уделяет наибольшее внимание, однако Opera и Safari тоже постепенно захватывают рынок), предлагают полезные функции, которыми Internet Explorer не обладает, и, кстати, более безопасны. И когда я говорю о полезных функциях, то имею в виду не безделушки, вроде парящих меню или танцующих бумажных скрепок, а такие, которые нравятся реальным пользователям, например функции отображения нескольких страниц в одном окне одновременно, позволяющие переключаться между ними при помощи закладки вверху экрана, как показано на рис. 9.1. Как и Internet Explorer, эти браузеры тоже бесплатны, но пользователям приходится их загружать, поскольку они не поставляются вместе с Windows. Если руководствоваться данными Дэвида Пога (David Pogue) из «New York Times»¹, они заставили упасть долю Microsoft на рынке с 95% до примерно 85%.

Вот так пользователи и сказали Microsoft: «Подключите больше людей к работе над улучшением Internet Explorer. Это нас волнует». Micro-

¹ Правильные показатели доли рынка браузера определить трудно, и важно, кто занимается этим вопросом, как составлены вопросы и кому вы хотите верить. Сложилось, однако, общее мнение, что доля Microsoft хоть и преобладает, но несколько снизилась. Естественно, увеличиваться ей было особенно некуда.



Рис. 9.1. Firefox одновременно отображает три разные страницы. Посмотрите на вкладки прямо над ними

soft отреагировала ускоренным выпуском Internet Explorer 7. Он содержит большинство функций, которые нравятся пользователям в других браузерах, и более безопасен, чем Internet Explorer 6. Такое соперничество делает свое дело, заставляя всех быть на пике активности.

Не нравится текстовый процессор или электронные таблицы пакета Microsoft Office? В помойку их. Быстрый поиск в Google по «Microsoft Office» и «альтернативы» выдает почти 9 миллионов результатов. Corel WordPerfect — это полноценный текстовый процессор, лидер рынка до появления Windows. Пакет поддерживает функции, не доступные в Microsoft Word, такие как конвертирование документов в популярный формат Adobe PDF и обратно, и к тому же дешевле. Sun Microsystems StarOffice

стоит еще меньше, еще менее функционален и позволяет всем, кто испытывает фанатические чувства к программам на Java, удовлетворить свои непреодолимые желания. OpenOffice, продукт общественного труда, может еще меньше, зато бесплатен. IBM предлагает SmartSuite, содержащий электронную таблицу 1-2-3, с которой началась революция ПК, полученную при покупке компании Lotus. Все эти программы умеют работать с документами Microsoft Office. У всех у них есть достоинства и недостатки, они точно так же, как и Microsoft Word, исторгают из вас вопли: «Какого черта они там думали!», но только по другим поводам. Конкуренция со стороны этих приложений заставляет Microsoft усердно работать над Office 12 (очередной версией пакета), чтобы это стало существенным продвижением вперед, коим Office 11 определенно не являлся.¹ Получится у Microsoft следующий Office успешным или нет, но все вынуждены усиленно работать. Что касается меня, то я настроил Word под себя так, чтобы он автоматически исправлял опечатки, которые я делаю чаще всего, например «hte» вместо «the», и я не перейду на другую программу или обновление, пока кто-нибудь, хоть Microsoft, хоть иной производитель, не приведет мне убедительные доводы в пользу этого.

Терпеть не можете Windows? Перестаньте ныть и купите Макинтош. У моей мамы он есть, как у многих моих коллег в Гарварде. Они говорят, что обожают свои Маки. Не хотите переплачивать за них? Купите ПК Dell, на которые устанавливается бесплатная операционная система с открытыми исходными текстами, вроде Linux. Цена от \$239 и выше, включая восстановленный 17-дюймовый монитор, но придется поработать, чтобы найти, установить и проверить программы для него, позаботиться об их совместимости. Если вы любите покупать мебель в Ikea, где прихо-

¹ Office 11 — это тот, который Microsoft рекламировала, изображая пользователей, не желавших обновлять свой пакет, с пластиковыми головами динозавров, что должно было свидетельствовать об их отсталости. «Динозавр» — это оскорбление в словарном запасе гика, наверное, следующее по грубости после «дурака». Но это не подействовало на пользователей, которые в массовом порядке игнорировали этот продукт. Эй, Microsoft, ваш пользователь — не вы. Их страхи — это опасность сломать работающую систему или получить бесполезный мусор, вроде танцующих скрепок, тормозящих компьютер так, что им приходится покупать новый, и они совсем не те, что ваши страхи — отстать от времени. Познайте вашего пользователя, ибо он — не вы. Тогда, может, вы и сможете продать какие-нибудь обновления.

дится выбирать тяжелые коробки на складе и тащить их до кассы, а потом до дома, а там самостоятельно ее собирать, но зато можно сэкономить, то вам эта затея может понравиться. Нет времени, или терпения, или силы для таких фокусов? В таком случае надо полагать, что Windows вам нравится больше, чем вы думали. Мне кажется, что за сотню баксов она много чего полезного делает, но не все с этим согласятся.

Если не обращать внимания на чокнутых борцов с мировым господством, то можно обнаружить, что на рынке представлен богатый выбор ПО. Не выбирайте автоматом программу с самым длинным списком возможностей, если только среди них нет такой, без которой вы просто жить не можете, и ее нет ни в одной другой программе. Настаивайте на бесплатном испытательном сроке, который сейчас предоставляется почти для каждой программы, и постарайтесь сделать так, чтобы у вас было время, чтобы полностью протестировать программу. Откройте приложение, попробуйте выполнить несколько задач и убедитесь, что работать просто. Если нет, выкиньте программу на помойку и найдите такую, работать в которой легко. Вы своими деньгами вознаграждаете добро и наказываете зло.

Если сайт какой-нибудь фирмы плох, не пользуйтесь им. Вместо этого позвоните им в службу работы с клиентами и поговорите с живым человеком, что обойдется фирме куда дороже. В следующий раз купите телефон с громкой связью, чтобы вам не пришлось держать трубку у уха, ожидая ответа сотрудника. А еще лучше, потратьте свои деньги в той фирме, чей сайт сделан хорошо или хотя бы лучше. Хорошо сверстаный сайт облегчает всем жизнь, а сверстаный плохо затрагивает финансы обеих сторон. Один из первых рецензентов этой книги сообщил: «Самый ужасный рецензист, заставивший меня заполнить несколько страниц с формами, а потом объявивший, что «Сервер перегружен; попробуйте позже», не сохранив моей информации, — это USAirways.com. Она не один раз это со мной проделывала, так что я теперь бронирую билеты на другие авиалинии, когда только можно, а если нужна именно USAir, то заказываю по телефону». Это стоит компании лишние \$20, если он прибегает к их услугам, и потерянного клиента, если нет. Они просто разбазаривают деньги!

Покупайте хорошие вещи, или, по крайней мере, те, что лучше, или, на худой конец, не такие плохие. Нет способа показать свое отношение более недвусмысленно.

2. Расскажите

Каждая успешная компания прислушивается к своим клиентам. Я помню, как мальчишкой еще открыл коробку с Cheerios и обнаружил, что многие колечки горелые. Родители помогли мне написать (относительно) вежливое письмо по адресу, указанному на коробке. Через несколько дней я был вознагражден ответом: официальное письмо с извинением, разъяснением причин и самое главное с купоном на получение двух коробок Cheerios бесплатно. Я был в восторге, когда получил их, правда, рассердился, когда мои родители заставили меня поделиться с младшим братом.

В большинстве своем производители *хотят* слышать отзывы от клиентов. Они понимают необходимость знать, что их покупатели думают, что они любят и не любят, что хотят увидеть изменившимся, а что оставшимся прежним. Как пишет Генри Петроски (Henry Petroski), профессор механики, в книге «Success Through Failure: the Paradox of Design» (Успех благодаря провалу: парадокс проектирования), Princeton University Press, 2006: «Неудовлетворенные пользователи — это источник важных для производителя отзывов о его продукции. На самом деле пользователи предоставляют такую информацию, которую трудно получить другими способами, а разработчики и производители зачастую слишком близки к продукту, чтобы оценить в полной мере, в каких отношениях он не оправдывает ожиданий. [У разработчиков] развивается близорукость по мере того, как они научаются обращаться с устройством таким образом, чтобы не сталкиваться с его изъянами. Они определяют некоторые границы и в этих рамках тестируют продукт. Покупатели же редко признают эти ограничения или читают мелкий шрифт в документации. Никакой набор тестов в лаборатории никогда не сможет обнаружить брешь или слабое место так быстро и успешно, ...как это сделает чья-нибудь творческая голова».

Некоторые разработчики ПО, которые не читали ни Петроски, ни Алана Купера, ни эту книгу, думают, что обойдутся без того, чтобы прислушиваться к своим пользователям. Они имеют дело с продуктом каждый день; только они знают, как он работает, стало быть, только они знают, что важно. В данной книге я доказываю, что это в корне неверно. К счастью, большинство разработчиков понимает, что им нужно и ваше участие. Они хотят продавать больше продуктов, хотят, чтобы вы чаще посещали их сайты, так что если вы отправите им вежливые комментарии, то они, скорее всего, оценят их. Это не означает, что они реализуют каж-

дую функцию, о которой их только попросят. Вы можете захотеть что-нибудь невозможное, несовместимое с другими вещами или прямо противоположное желаниям других пользователей. Разработчики расставляют приоритеты, руководствуясь общим числом пожеланий. В частности приоритет имеют те проблемы, которые порождают большую часть звонков в службу технической поддержки (что обходится очень дорого). Так что почаще им звоните, если что-нибудь ломается.

Компаниям необходимо читать почту, хотя в этом можно усомниться ввиду сложностей, которыми они обставляют процесс отсылки данных форм обратной связи, как на UPS.com, о чем говорилось в главе 2. UPS требует, чтобы вы пробирались через пять уровней форм, затем заполняли семь различных полей, чтобы послать им письмо. Облегчая вашу задачу, я разместил прямую ссылку на формы отзывов на веб-сайте своей книги, www.whysoftwaresucks.com. Я сделал это и для других компаний, которые подверг критике, вроде Starbucks. Я постараюсь поддерживать актуальность этих ссылок. И я добавлю другие, как только вы мне о них скажете. И мы еще организуем движение на этой основе (подробности в конце этой главы).

Отклики пользователей сильнее всего влияют на продукт на этапе раннего публичного тестирования, часто называемого «бета-тестированием». К примеру, Microsoft только что закончила бета-тестирование своего нового продукта «OneCare» — службы сетевой подписки, смысл которой поддерживать безопасность компьютера. Я считаю, что ее функциональность оставляет желать лучшего в сравнении с лидерами рынка Norton и McAfee, но это можно скомпенсировать простотой использования услуги. Тысячи пользователей участвовали в этом тестировании, в ходе которого мы установили ранние версии этого ПО на наши компьютеры, опробовали его и отослали отчеты. Microsoft открыла форум, на котором пользователь мог рассказать о своих впечатлениях, высказать свое мнение, задать вопросы и получить ответы представителей фирмы. Что касается именно этого тестирования, то частые запросы пользователей привели к заметному улучшению системы резервного копирования файлов.

Microsoft начала бета-тестирование вышеупомянутого Internet Explorer 7, как только эта книга пошла в печать. Она объявила об этом на своей домашней странице, www.microsoft.com, и любой заинтересованный пользователь мог присоединиться к процессу. В Сети можно найти и другие предложения бета-тестирования, но не каждый может принять в них

участие. Это что-то вроде клинического исследования. Производитель считает, что новое лечение лучше, иначе бы его не тестировали, но он может ошибаться (или они его вообще бы не тестировали?), и зачастую так и случается. Если вам нравится быть первопроходцем, влиять на процесс разработки нового продукта, как та семья, которая разрешает компании Nielsen записывать, какие ТВ-передачи они смотрят, чтобы составлять рейтинги популярности, то вам, может быть, понравится быть бета-тестером. Напротив, если вы почувствовали себя подопытным кроликом, тогда, наверное, нет. Это означает, что на результаты бета-тестирования всегда влияют любители новинок и технологии. А такие больше интересуются насыщенной функциональностью, чем молчаливое большинство пользователей, и их меньше беспокоит простота использования. Разработчикам не так просто выяснить, что думает молчаливое большинство пользователей. Поэтому в результаты бета-тестирования нужно вносить поправки в сторону меньшей нагруженности функциями и большей простоты применения, как в случае с переписью населения Соединенных Штатов, где применяются статистические методы для оценки той части населения, которая физически недостижима.

В предыдущей главе я рассказывал о регистраторе ошибок Windows — компоненте, который обнаруживает сбой программы и отправляет в Microsoft результаты для отладки. Там я говорил, что такой регистратор надо оставлять включенным. Представьте себе, как было бы трудно выяснять причину крушения самолета без бортовых самописцев. Данные о катастрофе вряд ли помогут жертвам крушения, но, будем надеяться, в будущем уберегут кого-то другого. Я надеюсь, вы оставите этот компонент включенным.

Как разработчик программного обеспечения скажу: мы знаем, что влезть в шкуру нашего пользователя труднее, чем мы считали первоначально. Так что *говорите* нам, какие достоинства и недостатки есть у наших продуктов.

3. Высмейте

Никогда не пренебрегайте силой «эго», особенно если речь идет о компьютерных гиках. Они ценят свой интеллект выше всех прочих достоинств. Выглядеть глупо в глазах своих коллег — вот их величайший страх, ночной кошмар, которого они будут стремиться избежать любой

ценой. Вот почему попадание в качестве плохого примера в такую книгу, как эта, или в «Зал Позора» на сайте типа *WebPagesThatSuck.com* оказывает сильное влияние на компанию. Управляющих и программистов увольняют за такие вещи.

Интернет дает гораздо большую, чем когда-либо раньше, возможность голосу отдельного человека быть услышанным. Почти все сайты, которые что-нибудь продают, предоставляют возможность оставить отзывы о продуктах, и я часто принимаю во внимание эти комментарии, когда решаю, что купить. К примеру, одна из моих книжек о программировании на *Amazon.com* получила оценки от «первоклассное введение в предмет» до «я хотел поставить ноль звездочек, но в *Amazon* нет такой возможности». Там есть огромное количество высказываний такого типа, а нейтральных не так много, так что потенциальному покупателю становится ясно, что эту книгу или любят, или терпеть не могут. (Так как вы уже дочитали ее почти до конца, такое разделение мнений вас не должно сильно удивлять). Эти отзывы подсунуты прямо под нос потенциальному покупателю, раздумывающему о покупке, что весьма усиливает впечатления предыдущих клиентов.

К примеру, ко дню рождения отца я однажды купил стерео-систему для Интернета. Это обычный АМ-FM-CD-ресивер, который ставится на полку, разве что туда можно вставить кабель с высокоскоростным подключением к Интернету и слушать потоковое аудио из сети (например, *RadioMargaritaville.com* или онлайн радиостанцию *Jimmy Buffett*). К сожалению, устройство плохо работало, и отец его в конце концов вернул обратно. Единственным положительным результатом стало то, что мать разрешила ему увеличить скорость подключения, чтобы можно было нормально слушать, а потом разрешила оставить это подключение, когда отец выкинул систему, так что теперь он наслаждается куда больше, чем раньше, высокой скоростью Интернета.

В прежние времена, до Интернета, все бы этим и закончилось: один недовольный покупатель, единичный случай возврата товара, и никто бы об этом и не знал. Но сегодня, будучи весьма разозлен, он написал на *Amazon.com* отзыв, который звучит примерно так: «Я был полностью приведен в замешательство этим замысловатым устройством с совершенно непостижимой документацией. У меня три высших образования, и я считаю себя грамотным в технике, но это было выше моего воображения... Этот продукт изготовлен в Китае, видимо, инструкцию писали там

же. Бойтесь его, как чумы китайской». Другой пользователь написал об этом же продукте: «Если вы техногик, любите гаджеты ради них самих, у вас есть пачка денег, которую вы готовы пустить на ветер, пядь волос, которую можно выдрать, тогда вперед — покупайте, чтобы можно было похвастаться. Но лучше дождитесь, пока кто-нибудь выпустит более приличный продукт и сможет его поддерживать». Я думаю, что дело сделано, и Amazon не много продаст таких устройств. Производитель прекратил производство этого продукта.

Высмейте, причем публично, производителей, которые выпускают плохие продукты. Так у вас появятся рычаги управления ими. Кроме того, это еще и приятно. Но при этом не забывайте хвалить хороших производителей.

4. Доверяйте

Вы не станете тратить \$10 на билет в кино, не прочитав отзывы о фильме, так почему вы поступаете так при покупке ПО, особенно если учесть, что с программой вы обычно проводите куда больше времени, чем на любом сеансе в кино, и что плохой фильм не утащит ваши пароли, опустошив ваш банковский счет. Вы обязаны постараться разузнать все о покупаемой программе не менее старательно, чем если бы это был фильм.

Так как у каждого кинокритика свой вкус, вы быстро найдете тех, с которыми он у вас общий.¹ Я обнаружил для себя, что мы с Уолтером Моссбергом (Walter Mossberg) из «Wall Street Journal» обычно думаем схоже (хотя я сомневаюсь, что он выразил бы это теми же словами). Он ценит простоту и удобство использования так же, как и я, в чем можно убедиться, прочитав первую строчку его дебютной колонки многолетней давности: «Персональные компьютеры просто слишком трудны в использовании, и в этом нет вашей вины». У него пристрастие к Макинтошам, которого я с ним не разделяю. Мне также нравится Ли Гомес (Lee Gomes), который ведет другую колонку в WSJ, и у Дэвида Пога из «New York Times», похоже, голова на месте. Еще меня восхищает Джерри

¹ Когда я переехал в Бостон, кинокритик из «Boston Globe» дал «Monty Python and the Holy Grail» всего одну звездочку из четырех. Я быстро приспособился воспринимать его отзывы наоборот: если ему нравилось, то мне нет, и наоборот.

Пурнель (Jerry Pournelle), писатель-фантаст и ведущий обозреватель на *Byte.com*. Он выдумал Закон Пурнеля, который я цитировал много раз и старался выполнять в своих книгах: «Примеров не бывает слишком много». Есть и другие авторы, которых вы быстро найдете, едва взявшись за дело.

Когда вы загружаете и устанавливаете программное обеспечение из Сети, обычно появляется окно, как на рис. 9.2, в котором спрашивается, доверяете ли вы поставщику и хотите ли установить данное ПО. Бесплезная и раздражающая трата времени, не предоставляющая никакой безопасности. Если вы знаете, какому производителю доверяете, то этот механизм может дать уверенность, что процесс безопасен. Но это не главный вопрос, который вы задаете себе, путешествуя по Интернету. Вместо этого вы обычно задумываетесь: «Я никогда не слышал об этой компании, пока не поискал [эту фирму] в Google. Сейчас я на их сайте и не против опробовать их программу, но я не знаю, можно ли им верить». Это окошко не ответит на ваш вопрос и вообще ничем вам не поможет.

Откуда у вас могут быть данные, чтобы правильно ответить на этот вопрос? Не может, не должно и не будет. Программисты, которые взваливают груз решения этой проблемы на вас, а сами снимают с себя всякую ответственность, не заслуживают теплых чувств в свой адрес. Так как определить, кому можно верить в Сети?

В различных развитых отраслях промышленности эту проблему решили путем делегирования функций контроля осведомленным третьим сторонам, которым клиенты доверяют. К примеру, когда вы покупаете в ма-



Рис. 9.2. Бесплезная, дурацкая и досадная трата времени, и никакой безопасности

газине еду, маркированную как «экологически чистая», откуда вам знать, что она на самом деле такая? Если вы не специалист в пищевой промышленности и не занимаетесь всю жизнь изучением состава продуктов, то не сможете самостоятельно сделать обоснованные выводы. Вместо этого вы ищете сертификат, выданный компетентной организацией, которой вы доверяете и которая может производить необходимые анализы, например Oregon Tilth (www.tilth.org). Такие программы сертифицирования третьей стороной существуют для многих других качеств, которые волнуют покупателей, когда нельзя верить заявлениям производителя: древесина из восполняемых лесных ресурсов, кофе по честным ценам, пища, приготовленная в соответствии с религиозными верованиями потребителей. В индустрии ПО надо поступить аналогичным образом.

Программное обеспечение можно сертифицировать направленным либо распределенным тестированием. Для первого способа понадобилась бы мощная тестовая лаборатория, по принципу схожая с Underwriters Laboratories, что сертифицирует электроприборы в США. Она бы выпустила свод рекомендаций, например, относительно применения методов безопасного программирования и приемлемой политики сохранения конфиденциальности личных данных. Она бы тестировала приложения на соответствие этим стандартам и присоединяла цифровые сертификаты к тем, что удовлетворяют требованиям. Лаборатория потребует оплаты своих услуг, а производитель включит эти расходы в цену приложения. Возможно, один только факт траты больших сумм на сертификацию отпугнет большинство злодеев. Возможно, для привлечения клиентов агентство по проверке предлагало бы какие-то страховые выплаты в случае нанесения ущерба приложением.

Такая узконаправленная модель поставит на грань разорения малых производителей, т. к. они не выдержат соответствующих расходов. Альтернативой может оказаться распределенная модель, похожая на применяемую eBay, где покупатели оценивают продавцов, и наоборот. Покупатели, которые устанавливают программу, оценивают путем голосования, можно ли доверять этому приложению, а собирающиеся загрузить его смогут опереться на данные голосования при вынесении решения о покупке. Это не поможет самым первым пользователям, да и результаты не могут считаться непогрешимыми. Однако это очень хороший способ, его легко реализовать и он пригоден для малых производителей и даже для испытательных версий программ. Получится как в ресторанном обозре-

нии Zagat, которое не посылает своих критиков в рестораны, как это делают газеты, а суммирует отзывы тысяч клиентов.

Система должна быть устроена так, чтобы обманщик ничего не смог подтасовать. Возможно, право голосовать будет предоставлено только зарегистрированным пользователям, оплатившим услуги, путем отправки на электронный адрес избирательных бюллетеней. Может быть, этим займется некий общественный фонд либо eBay или Amazon предоставят программное обеспечение ради возможности выступить в качестве спонсора. Я предлагаю печатать на одобренном ПО штамп «Doesn't Suck». Он будет иметь вид леденца в перечеркнутом кружке.

Недостаточно поместить сертификат на сайт для всеобщего обозрения, потому что пользователи никогда их не читают. Финальный штрих — заставить операционную систему взаимодействовать с этими сертификатами, выдаваемыми третьими сторонами, свободно загружать и устанавливать сертифицированные программы, блокируя остальные. Это уже дело разработчиков операционных систем.

Пока мы не нашли рабочие варианты таких систем проверки, лучшим вариантом будет отыскать оценщиков ПО, которым вы доверяете, и следовать их советам.

5. Сплачивайтесь

Антрополог Маргарет Мид (Margaret Mead, 1901–1978) как-то сказала: «Никогда не сомневайтесь, что небольшая группа умных, преданных своему делу людей может изменить мир. В истории только так всегда и было». Да, Интернет повысил возможности отдельных личностей. Но важнее то, что он помогает людям собраться вместе, образуя группы, обладающие реальной силой. Теперь, когда каждый подключен к Сети, это элементарно. Еще десять или даже пять лет назад бывшие жертвы сексуальных домогательств священнослужителей едва ли могли сплотиться, чтобы заставить Католическую Церковь в Бостоне и других городах выполнить свои требования.

Современное общество действует, собираясь в силовые группы для последующей борьбы на рынке. Вспомните об Американском союзе защиты гражданских свобод или о Национальной стрелковой ассоциации; о Гражданах за разумный экономический рост и Гражданах за ограничение налогов. Эти группы делают много шума, подают иски, организуют послания

и бойкоты, пытаются воздействовать на членов конгресса, а иногда даже сочиняют законы вместо перегруженных штатных законодателей. Однако очень немногие из них занимаются программным обеспечением — гораздо меньше, чем можно ожидать, учитывая глубину его проникновения в нашу жизнь. Некоторые группы, например Privacy Rights Clearinghouse (www.privacyrights.org) и Electronic Frontier Foundation (www.eff.org), занимаются правом на неприкосновенность частной жизни в Сети и вне ее. Но ни одна не высказалась по поводу качества ПО, по крайней мере на моей памяти.

Это надо исправить, и мы с вами это сделаем. Мы должны привлечь силу современного Интернета, чтобы наши слова о качестве ПО были услышаны. Чем больше людей прочтет плохие отзывы, тем сильнее компании, выпустившие плохой продукт или сайт, будут стараться избежать таких отзывов. Если я резко критикую UPS.com, то она думает, что я ненормальный и слишком тупой, чтобы понять их блестящий, интуитивный интерфейс. Но если мы все вместе зададим им жару, может, до них и дойдет что-нибудь, и они поймут, что непонятливые тут они, а не мы, и что им пора меняться. *(Так что покупайте побольше экземпляров этой книжки для своих друзей!)*

Чтобы достичь уровня критической массы и влияния, я организовал Suckbusters по адресу Suckbusters.com (рис. 9.3). Наше кредо — «Because Software Shouldn't Suck» (ПО не должно быть отстоем). В ответ на список ФБР «Десять Самых Разыскиваемых» я составляю список «Десять Наименее Желанных» поганых софтинок. Мне нужна ваша помощь в этом деле. Присылайте мне самые ужасные программы, опишите их как можно подробнее. Я постараюсь поддерживать сайт актуальным. Я опубликую ссылки на электронные адреса производителей этого ПО, а также заведу блог, в котором мы обсудим, в чем проблемы каждой отдельно взятой программы и как она должна работать.

Перефразируя Арло Гатри (Arlo Guthrie) из «Ресторана Alice», если я один сделаю это, обо мне подумают как о сумасшедшем (но это вы и так знаете). Если мы сделаем это втроем, они подумают, что это организация. А вот если 50 человек это сделают, они подумают, что это движение. Так и будет. Так что присоединяйтесь ко мне. И мы превратим мир ПО в такой, каким он может и должен быть.

Вы не знали, о каких улучшениях просить, вы даже не знали, что это вообще возможно, а теперь знаете. Пять или шесть десятков отзывов

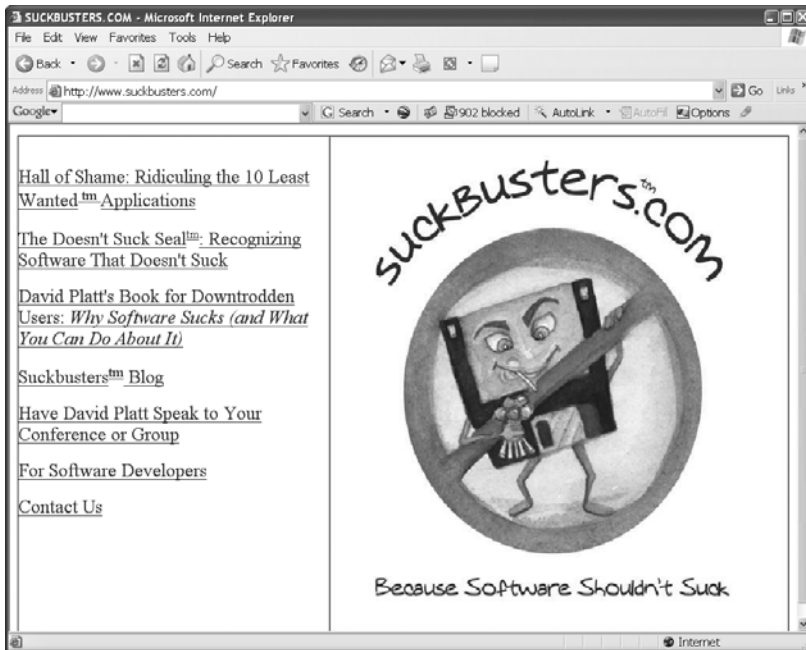


Рис. 9.3. Веб-сайт *Suckbusters.com*, основанный Дэвидом Платтом

в день — со словами «секция выбора языка на вашем сайте куда сложнее, чем на Google, исправьте это и, кстати, сделайте так, чтобы до этой формы обратной связи было проще добраться» — привлекут их внимание. Переключая свой бизнес, будь то логистика, торговля кофе или копирование документов, на работу с компаниями, которые делают легкие для работы программы и сайты, вы ударяете некомпетентные фирмы по их самому больному месту, а выставляя плохие работы на посмешище, тыкаете разработчиков носом в их же идиотизм.

Все зависит от нас, покупателей, — сделать так, чтобы рынок ПО регулировался так же, как это случилось с автомобильной промышленностью, банковским делом и розничной торговлей. Мы сможем это сделать. Начнем. Сейчас же.



Эпилог

Спасибо, что вы прочли эту книгу. Знаю, как вы заняты и сколько еще книг ждут вашего прочтения. Я признателен, что вы выбрали мою. Как разработчики программ работают для вас, клиента, так и я работаю для вас, читателя. Надеюсь, вы не жалеете о потраченном времени. Еще раз спасибо. Противно было бы писать книгу для себя самого.

Видя, как растут мои дочки, я испытываю противоречивые чувства. Я так любил, когда годовалая дочурка выбегала на берег в деревенском платьице, и никогда больше не смогу этого увидеть. На метках их детских пижамок была надпись «Если бы только они оставались маленькими...». И вместе с тем на самом деле всем нам хотелось бы другого.

Программная индустрия тоже взрослеет, и это зрелище тоже вызывает смешанные чувства. Черт побери, мы беспечно поживали, когда нашей главной проблемой было сгладить уголки карт, которые рассыпались веером, если сходилась пасьянс «солитер». Теперь же мы несем на своих плечах всю тяжесть мира, как и положено взрослым людям. Но я надеюсь при этом, что несмотря на зрелость, обретаемую нашей отраслью, мы не утратим полностью это ощущение игры — так же, как я надеюсь на это в отношении своих дочерей.

Я был так же счастлив и горд наблюдать их и помогать им в их рождении и в детстве и, даст Бог, в юности и в зрелые годы, как испытал счастье работать в тот же период в нашей отрасли. Не при зарождении: здесь

постарались такие гиганты, как Алан Тьюринг¹ и Джон фон Нейман.² Не в детстве: оно связано с такими гигантами, как адмирал Хоппер, которая столь удачно была наречена именем Грэйс.³ Но в подростковом возрасте, который всегда бывает бурным, переходящем во взрослые и зрелые годы.

¹ Алан Мэтисон Тьюринг (1912–1954) считается основоположником компьютерной науки, заложившим ее математические основы в статье 1936 года «On Computable Numbers» (О вычислимых числах). Широкой публике он известен по вкладу в британский проект по взлому нацистской шифровальной машины Enigma во время Второй мировой войны. В 1952 году он был осужден за нарушение английских законов, касающихся гомосексуализма, из-за чего лишился права работать с секретными материалами и был отстранен от своей работы. Двумя годами позднее он совершил самоубийство, отравившись яблоком с цианистым калием.

² Джон фон Нейман (1903–1957) преподавал математику в Принстоне, где был коллегой Эйнштейна и профессором у Тьюринга. Участвуя во время Второй мировой войны в Манхэттенском проекте, он открыл важные уравнения, сделавшие возможным создание атомной бомбы, сброшенной на Нагасаки, и большей части атомного оружия в последующем. Чтобы решать эти уравнения при конструировании еще более сложных видов оружия, он разработал принципиальную схему цифрового компьютера, на которой основано большинство современных компьютеров. Он в какой-то мере предвидел мировоззрение современных гиков, когда сказал: «Математические объекты нельзя понять, к ним можно лишь привыкнуть».

³ Грэйс Мюррей Хоппер (Grace Murray Hopper, 1906–1992, пробег намного лучше, чем у Тьюринга или фон Неймана) получила докторскую степень по математике в Йейле в 1934 году. Титул — не дань вежливости или кличка, как бывает у спортсменов. Это ее звание (контр-адмирал, если быть точным, тарифный разряд O-7), которое она получила, находясь на действительной военной службе в ВМС США. Военную службу она начала в 1943 году и продолжала с перерывами до окончательной отставки в 1986 году. Она разработала COBOL, первый современный язык программирования, который применяется до сих пор и позволяет программистам писать программы, используя синтаксис, напоминающий английский язык. Грэйс приписывается авторство (1947) термина «отладка компьютера» («debugging the computer»), когда она вынула казенную электрическим током бабочку (сейчас ее можно видеть в Смитсоновском институте) из вычислительной машины Harvard Mark II Calculator (один из первых компьютеров, работавший на электромеханических реле). Флот США в 1996 году назвал в ее честь ракетный эсминец класса «Арли Берк» (USS Hopper, DDG-70).

Это уже мое время и время моих собратьев-гиков. Трудиться, ориентировать, служить.

До новых встреч, дорогие читатели. Берегите себя. Берегите друг друга. И смотрите на вещи проще, но — смотрите.

Дэвид С. Платт

Ипсвич, Массачусетс, 2006

По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 5-93286-097-9, название «Софт – отстой! И что с этим делать?» – 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), где именно Вы получили данный файл.