

Алан КУПЕР

ПСИХ БОЛЬНИЦА



в руках пациентов

*почему высокие
технологии сводят нас
с ума и как восстановить
душевное равновесие*



По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru – Книги России» единственный легальный способ получения данного файла с книгой ISBN 5-93286-071-5, название «Психбольница в руках пациентов» – 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.

THE INMATES ARE RUNNING THE ASYLUM

Why High-Tech Products Drive Us Crazy
and How to Restore the Sanity

Alan Cooper

SAMS

ПСИХБОЛЬНИЦА

в руках пациентов

Почему высокие технологии сводят нас с ума
и как восстановить душевное равновесие

Алан Купер



Санкт-Петербург — Москва
2005

Серия «Профессионально»

Алан Купер

Психбольница в руках пациентов

Перевод М. Зислиса

Главный редактор
Зав. редакцией
Научный редактор
Редактор
Верстка
Корректор

А. Галунов
Н. Макарова
А. Копылов
В. Овчинников
В. Овчинников
С. Беляева

Купер А.

Психбольница в руках пациентов. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 336 с., ил.

ISBN 5-93286-071-5

Как противостоять натиску компьютерных технологий, проникающих в нашу жизнь с ужасающей скоростью? Наши телефоны, фотокамеры, автомобили – все, что нас окружает, автоматизируются, программируются, создаются людьми, которые, стремясь получить выгоду от применения микросхем, уклонились от своей прямой обязанности – делать эти продукты *простыми в применении*.

И это не преувеличение, это реальность. Наша жизнь все больше концентрируется вокруг превратностей, странностей, решений и катастроф индустрии высоких технологий. Разработчики программ, устройств и технологий думают не так, как мы. Облеченные полномочиями исполнительные лица ни на что не влияют в мире высоких технологий – здесь всем заправляют инженеры. *Мы разрешили пациентам завладеть психбольницей*. Алан Купер предлагает решение проблемы: программированию должно предшествовать проектирование.

ISBN 5-93286-071-5

ISBN 0-672-31649-8 (англ)

© Издательство Символ-Плюс, 2004

Original English language title: The Inmates Are Running the Asylum by Alan Cooper, Copyright © 1999, All Rights Reserved. Published by arrangement with the original publisher, Pearson Education, Inc., publishing as PRENTICE HALL.

Название оригинала на английском языке: The Inmates Are Running the Asylum by Alan Cooper, Copyright © 1999, все права защищены. Публикуется по соглашению с оригинальным издателем, Pearson Education, Inc. (PRENTICE HALL).

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 11.10.2004. Формат 70х90/16. Печать офсетная.

Объем 21 печ. л. Тираж 3000 экз. Заказ N

Отпечатано с готовых диапозитивов в ГУП «Типография «Наука»

199034, Санкт-Петербург, 9 линия, 12.

Оглавление

Предисловие научного редактора	16
Предисловие	18
Введение	21
Часть I. Компьютерная безграмотность	37
1. Загадки века информации	39
Что получится, если скрестить компьютер с самолетом?	39
Что получится, если скрестить компьютер с фотокамерой?	41
Что получится, если скрестить компьютер с будильником?	43
Что получится, если скрестить компьютер с автомобилем?	45
Что получится, если скрестить компьютер с банком?	46
Компьютер позволяет легко попасть в беду	47
Коммерческое программное обеспечение тоже страдает	49
Что получится, если скрестить компьютер с военным кораблем?	51
Техноярость	52
Индустрия в «несознанке»	53
Мотивы создания этой книги	54
2. Когнитивное сопротивление	57
Поведение, не связанное с физическими силами	57
Проектирование – слово емкое	59
Отношения между программистами и проектировщиками	61
Большинство программ проектируются случайным образом	61
Проектирование «взаимодействия» против проектирования «интерфейса»	62
Отличительные черты продуктов, основанных на программном обеспечении	63
Танцующий медведь	66
Стоимость дополнительных возможностей программного обеспечения	68
Апологеты и уцелевшие	70
Наша реакция на когнитивное сопротивление	74

Демократизация власти потребителя	75
Винovat пользователь	76
Программный апартеид	78
Часть II. Масштабные издержки	81
3. Пустая трата денег	83
Управление, ориентированное на крайний срок сдачи	83
Что такое «готово»?	84
Поздний выпуск – не беда	88
Торг за набор функций	89
Возможности не всегда нужны	91
Итерации и миф о непредсказуемости рынка	92
Скрытые издержки некачественного программного обеспечения	96
Издержки прототипирования	99
4. Танцующий медведь	105
Если это проблема, то почему ее до сих пор не решили?	105
Жертва бытовой электроники	106
Чем плохи почтовые клиенты	107
Чем плохи программы для планирования	109
Чем плохи календари	110
Массовая веб-истерия	111
Что не так с программным обеспечением?	112
5. Нелояльность клиентов	118
Привлекательность	118
Одно сравнение	122
Время выхода на рынок	125
Часть III. Как есть суп вилкой	129
6. Психбольница в руках пациентов	131
Вожделение на заднем сиденье	131
Подготовка катастрофы	134
Компьютеры против людей	138
Учим собак быть кошками	140
7. Homo Logicus	145
Авиационный тест	146
Психология программистов	147
Программисты пожертвуют простотой ради контроля	148

Программисты обменяют успех на понимание	150
Программисты сосредотачиваются на исключительных ситуациях	152
Программисты ведут себя грубо и прямолинейно	154
8. Отмирающая культура	158
Культура программирования	158
Повторное использование кода	159
Общепринятая культура	163
Культурная изоляция	170
Шкурный интерес	171
Обесчеловечивает процесс, а не технология	176
Часть IV. Проектирование взаимодействия – выгодный бизнес	177
9. Проектирование для удовольствия	179
Персонажи	179
Проектируйте для одного персонажа	181
Персонаж должен быть конкретным	185
Персонаж должен быть воображаемым	186
Описание должно быть подробным, а не идеальным	187
Реалистичный взгляд на уровень подготовленности	188
Персонажи закрывают споры о функциях	190
Персонаж пользователя, а не покупателя	193
Подбор персонажей	193
Ключевые персонажи	195
Пример: Sony Trans Com и P@ssport	196
10. Проектирование ради результата	207
Мы решаем задачи, чтобы достичь целей	207
Задачи не являются целями	208
Проектирование, ориентированное на цели	210
Цели личные и цели практические	213
Личные цели	215
Корпоративные цели	216
Практические цели	217
Ложные цели	218
И у компьютера есть человеческие черты	219
Проектирование и вежливость	220
Что делает программы вежливыми?	223
Пример: Drumbeat от Elemental	233

11. Проектирование для людей	242
Сценарии	242
Повседневные сценарии	243
Обязательные сценарии	244
Сценарии исключительных ситуаций	244
Адаптирующийся интерфейс	245
Вечные середняки	246
Словарь	250
Реальность смеется последней	251
Пример: Logitech Scanman	253
Преодоление разрыва между устройствами и программами	264
Меньше значит больше	266
Часть V. Возвращаемся на место водителя	269
12. В отчаянных поисках эргономики	271
Последовательность	271
Юзабилити-тестирование	274
Многопрофильные команды	276
Проектирующие программисты	276
Откуда вы знаете?	277
Руководства по стилю	278
Фокус-группы	280
Визуальное проектирование	280
Промышленный дизайн	282
Классная новая технология	283
Итерации	284
13. Управляемый процесс	286
Кто на самом деле самый влиятельный?	286
Поиск основы	292
Производство фильмов	293
Хорошая сделка	295
Кто отвечает за качество продукта?	302
Включение проектирования в процесс	303
14. Мощь и удовольствие	307
Пример налаженного проекта	308
Осознанное проектирование взаимодействия	310
Польза от перемен	312
Почему они не едят пирожных?	313
Алфавитный указатель	318

Посвящается

Сью, Скотт и Мери, с любовью

Отзывы на книгу Алана Купера

От этого становится не по себе, но это правда. Персональные компьютеры породили новую взаимозависимость Новой Эры. Они позорят нас, раздражают нас, но мы продолжаем тратить на них деньги. Книга Алана Купера объясняет, почему все должно быть не так и что мы можем сделать для исправления ситуации. Чтение доставляет удовольствие и помогает спуститься с небес на землю.

– *Жан-Луи Гассе, президент Be, Inc.*

И впрямь пациенты. Людям пора проснуться и сказать «Все, с нас хватит!» Снова Алан Купер освещает нам путь. Чтение его книг следует сделать обязательным во всех компаниях, имеющих отношение к высоким технологиям и считающих, что они работают на благо клиентов. Нам нужно больше книг, подобных этой, больше людей, похожих на Алана Купера.

– *Дон Норман, Nielsen Norman Group; автор книги «The Invisible Computer»*

Если вы когда-нибудь задумывались, почему программы, написанные для компьютеров, и многочисленные электронные штучки ведут себя так, словно были созданы болваном или пытаются сделать болвана из вас, прочтите эту книгу. Книга Купера – это манифест о том, как повысить качество жизни в век, когда мы проводим много времени во взаимодействии с технологией.

– *Питер Хуршберг, президент Elemental Software*

Подход к проектированию, предлагаемый Купером, быстро завоевывает последователей, среди которых и такие клиенты, как Sun Microsystems, Coca-Cola, Compaq и Dow Jones.

– *Fast Company Magazine*

Целеориентированная методология проектирования Алана Купера, сосредотачивающая внимание на конкретных конечных пользователях (персонажах), привлекла команду проектировщиков к элегантному решению невероятно сложной проблемы проектирования пользовательского интерфейса.

– *Карен Ивенсон, руководитель инженерных разработок Sony Trans Com*

Браво! Живой и проникновенный трактат о проектировании взаимодействия от того, кто стоит у истоков. Раздавайте экземпляры друзьям, коллегам, своим клиентам. Об этой книге обязательно будут говорить, а возможно, и спорить.

– *Клемент Мок, основатель Archetype Studio, креативный директор Sapient*

Эта книга изменит ваши отношения с технологией, будь вы ее создатель или потребитель. Технология призвана помогать человеку, а не приводить его в замешательство. Это лучшая из прочитанных мною книг о проектировании взаимодействий: в ней Алан рассказывает, почему компьютеры неверно взаимодействуют с людьми и как излечить эту хроническую болезнь. Обязательная книга для всех, кто имеет дело с продуктами высоких технологий.

– Джефф Хадфилд, главный редактор *Visual Basic Programmer's Journal*

Мы находимся под сильным впечатлением от энтузиазма, профессионализма и методов работы нескольких команд Cooper Interaction Design, сотрудничающих с нами над перепроектированием центральных модулей R/3.

– Маттиас Беринг, руководитель программы, SAP

Если бы Алан Купер присутствовал на сказочном шествии короля через город, то первым бы заявил о том, что король голый, после чего показал бы всем, как создавать наряды привлекательные, экономичные и приятные в ношении. В этой книге Купер бросает вызов индустрии программного обеспечения на всех уровнях – от тех, кто пишет код, до высшего руководства, демонстрируя изъяны современных программ и указывая на приемы проектирования, позволяющие исправить положение. Основываясь на своем богатом опыте в проектировании приложений, он излагает материал настолько ясно и практично, что книга может стать бесценным откровением для любого читателя – даже того, чей контакт с индустрией компьютерных приложений ограничен использованием продуктов, получивших название «танцующих медведей».

– Терри Виноград, профессор информатики Стэнфордского университета

Отличные программы создаются для пользователей. Работать с такими программами легко и приятно. Такая работа эффективна. Алан Купер четко показывает, что в массе своей программы создаются для программистов: на основе их догадок о потребностях пользователя, из любимых функций, индивидуальных особенностей стиля. Эта умная книга учит руководителей тому, что они должны знать, чтобы создавать системы, покоряющие рынок. Любой участник рынка – и самый удачливый и всего добившийся, и только еще полный надежд стать следующей сверхновой, поймет, что эта книга – одна из наиболее глубоких, практичных и полезных.

– Ларри Кили, президент *Doblin Group*

Сочетание острого ума автора и его высочайшего профессионализма делают эту книгу не только очень познавательной, но еще и весьма увлекательной.

– Хайди Ройзен, *Be, Inc.*

Алан Купер понимает разницу между интерфейсом и взаимодействием лучше, чем кто-либо из знакомых мне людей. Его идеи опираются на многолетний опыт в содействии разработке продуктов, элегантно и ненавязчиво проникающих в нашу жизнь. Он претворял свои идеи в жизнь много лет и теперь, наконец, нашел время превратить свой практический опыт в четкое описание задачи, с которой мы столкнулись, в методологию побега из психбольницы, которую мы с такой любовью выстроили. Читайте дальше – и обретете свободу.

– Пол Саффо, директор *Института Будущего*

Об авторе

По мнению Митча Уэйта, основателя Waite Group Press, Алан Купер – «Отец Visual Basic». Перу Алана Купера принадлежит книга «About Face: The Essentials of User Interface Design» (IDG Books, 1995). В 1994 году Билл Гейтс представил Алана к редкой и желанной награде Windows Pioneer Award, признав, что его вклад в изобретение Visual Basic способствовал успеху системы Microsoft Windows. В 1998 году Алан был представлен к награде Software Visionary Award. Сегодня он возглавляет Cooper Interaction Design, консультирующую фирму, которая выполняла работы по проектированию продуктов для таких компаний, как 3M, Elemental, Ericsson, Fujitsu, IBM, Logitech, McGraw-Hill, Sagent, SAP, Sony, Varian, VISA и Sun Microsystems.

Алан открыто выступает в защиту тех, о ком забывают в процессе разработки электронных продуктов, – *покупателей*.

В течение двадцати лет Алан Купер проектировал и создавал потребительские программные продукты, среди которых SuperProject, MicroPhone II для Windows, а также интерфейс визуального программирования для Visual Basic. В 1976 году Купер основал компанию Structured Systems Group Inc., которая, как сказано в книге «Fire In the Valley» (Пожар в Кремниевой Долине), выпустила «возможно, первое серьезное бизнес-приложение для микрокомпьютеров».

Купер состоит в организациях Corporate Design Foundation и American Center for Design. Он бывший директор калифорнийского отделения ассоциации проектирования программного обеспечения (Association for Software Design), а также член совета директоров этой организации. Купер – директор Software Design и Software Forum, а также основатель SEF Windows SIG, крупнейшей в мире группы разработчиков для Windows. Он часто и уверенно выступает как делегат индустрии и пишет о пользовательском интерфейсе и концептуальном проектировании программных продуктов.

Благодарности

Я не смог бы написать эту книгу без помощи и участия многих замечательных друзей и коллег. В частности, несколько человек выполнили трудную и кропотливую работу – прочитали и прокомментировали рукопись, иногда по несколько раз. Их комментарии вынуждали меня отвечать на сложные вопросы, предлагать читателю новые темы, резюмировать свои соображения, умиротворяли мой пыл, усыряли мое дикое негодование. Эта книга стала гораздо лучше благодаря вкладу Ким Гудвин (Kim Goodwin), Лейн Хэлли (Lane Halley), Келли Боумен (Kelly Bowman), Скотта Мак-Грегора (Scott McGregor), Дэвида Уеста (David West), Майка Нельсона (Mike Nelson), Марка Джирска (Mark Dzierisk), Алана Карпа (Alan Karp), Терри Суок (Terry Swack), Луи Вейцмана (Louie Weitzman), Уэйна Гринвуда (Wayne Greenwood), Райана Ольшавски (Ryan Olshavsky), Джона Майера (John Meyer), Лайзы Сондерс (Lisa Saunders), Винни Шоуз (Winnie Shows), Кевина Вандрайка (Kevin Wandryk), Глена Халстеда (Glenn Halstead), Брайана О’Салливана (Bryan O’Sullivan), Чака Оуэна (Chuck Owen), Майка Суэйни (Mike Swaine), а также Скипа Уолтера (Skip Walter). Спасибо вам за время, участие и мудрость. Особенно ценными для меня в плане кристаллизации тем книги стали комментарии и советы Джонатана Кормана (Jonathan Korman). Я должен также поблагодарить всех талантливых и усердных сотрудников CID¹, которые делали за меня мою работу, пока я был занят книгой. Особой благодарности заслуживает Уэйн Гринвуд (Wayne Greenwood), который замечательно работал под жестким прессингом, сохраняя наш боевой дух и высокое качество наших проектов.

Создание иллюстраций стало одной из наиболее интересных задач. Чед Кубо (Chad Kubo), мастер изобразительного искусства, проделал замечательную работу – превратил мои расплывчатые идеи в четкие, запоминающиеся образы. Они многое привнесли в книгу. И эти иллюстрации никогда не появились бы без неутомимого художественного руководства со

¹ Компания Cooper Interaction Design.

стороны Пенни Бейлес (Penny Bayless) и Дэвида Хейла (David Hale). Были и другие люди, решавшие производственные задачи. Спасибо Брит Кацен (Brit Katzen) за изучение и перепроверку фактов, спасибо Майку Генри (Mike Henry) за литературное редактирование.

Создание книги – это деловое предприятие, и за то, что оно стало успешным, я должен поблагодарить команду сведущих в технологии предпринимателей, возглавляемую моим агентом, Джимом Левайном (Jim Levine), и включающую Глена Халстеда (Glenn Halstead), Линн Боумен (Lynne Bowman), Келли Боумен (Kelly Bowman), а также Сью Купер (Sue Cooper). Со стороны издательства Macmillan поддержку на протяжении всего проекта осуществлял Брэд Джоунс (Brad Jones), однако наибольшего внимания заслуживает Крис Вебб (Chris Webb), чья целеустремленность, сосредоточенность и тяжелый труд сделали возможным создание этой книги.



Я очень ценю людей, оказавших мне моральную поддержку, предоставивших реальные истории, советы и потративших свое время. Большое спасибо Дэниелу Эпплману (Daniel Appleman), Тодду Баше (Todd Basche), Крису Байеру (Chris Bauer), Джеффу Безо (Jeff Bezos), Элис Блэр (Alice Blair), Мишель Борк (Michel Bourque), По Бронсону (Po Bronson), Стиву Калде (Steve Calde), Дэвиду Карлику (David Carlick), Джеффу Карлику (Jeff Carlick), Кэрол Кристи (Carol Christie), Клэй Коллье (Clay Collier), Кендаллу Косби (Kendall Cosby), Дэну Крэйну (Dan Crane), Роберту Крингели (Robert X. Cringely), Трою Дэниелсу (Troy Daniels), Лайзе Пауэрс (Lisa Powers), Филипу Энглхарту (Philip Englehardt), Карен Ивенсен (Karen Evensen), Риджели Эверс (Ridgely Evers), Ройял Фаррос (Royal Farros), Пэт Флек (Pat Fleck), Дэвиду Фору (David Fore), Эду Форману (Ed Forman), Эду Фридкину (Ed Fredkin), Жану-Луи Гасце (Jean-Louis Gasse), Джиму Гею (Jim Gay), Рассу Голдину (Russ Goldin), Владу Горелику

(Vlad Gorelik), Марсии Грегори (Marcia Gregory), Гарренту Грюнеру (Garrett Gruener), Чаку Хартледжу (Chuck Hartledge), Теду Харвуду (Ted Harwood), Уиллу Хирсту (Will Hearst), Тамре Хитершоу-Харт (Tamra Hethershaw-Hart), ДжейДи Хильдебранду (JD Hildebrand), Лори Хиллз (Laurie Hills), Питеру Хиршбергу (Peter Hirshberg), Ларри Кили (Larry Keeley), Гэри Краткину (Gary Kratkin), Деборе Курата (Deborah Kurata), Тому Лефлеру (Tom Lafleur), Полу Лотону (Paul Laughton), Элен Леви (Ellen Levy), Стивену Листу (Steven List), Ти.Си. Мангану (TC Mangan), Дэвиду Майстеру (David Maister), Роберту Мэю (Robert May), Дону МакКини (Don McKinney), Кэтрин Медоуз (Kathryn Meadows), Лайзе Митчелл (Lisa Mitchell), Джефффри Муру (Geoffrey Moore), Брюсу Мовери (Bruce Mowery), Нату Майерсу (Nate Myers), Эду Нихаусу (Ed Niehaus), Констанс Петерсен (Constance Petersen), Кейту Плису (Keith Pleas), Роберту Райнману (Robert Reimann), Джону Ривлину (John Rivlin), Говарду Рейнгольду (Howard Rheingold), Гейди Ройцену (Heidi Roizen), Нилу Рубенкингу (Neil Rubenking), Полу Сафо (Paul Saffo), Джошу Зайдену (Josh Seiden), Рассу Зигельману (Russ Siegelman), Донне Слоут (Donna Slote), Линде Стоун (Linda Stone), Тони Уокеру (Toni Walker), Кевину Уиксу (Kevin Weeks), Кевину Уэлшу (Kevin Welch), Дэну Уиллису (Dan Willis), Хэзер Уинкль (Heather Winkle), Стефану Уайлдстрому (Stephen Wildstrom), Терри Винограду (Terry Winograd), Джону Цикеру (John Zicker) и Пьерлуиджи Заппакосте (Pierluigi Zappacosta).

Этот «проект на год» продлился двадцать месяцев, и моя семья проявила большое терпение. На мне величайший долг любви и благодарности перед моей супругой Сью Купер и моими красивыми юными сыновьями Скоттом и Марти. Люблю вас всем сердцем.

Предисловие научного редактора

От своего лица благодарю издательство «Символ-Плюс» за выпуск этой в высшей степени актуальной книги.

Молодая, бурно развивающаяся отрасль информационных технологий (ИТ) все глубже проникает в нашу жизнь. Стремительно расширяется аудитория потребителей электронных продуктов: например, мобильными телефонами сейчас пользуются дети и пожилые люди, хотя совсем недавно этот продукт относился к разряду элитных. Но, к сожалению, большинство производителей до сих пор делают свои товары по принципу «главное – быстрее продать». При этом основными конкурентными преимуществами продукта и продавец и покупатель, как правило, считают его функциональные возможности, иными словами – «навороченность». На рынке появляются сотни моделей сотовых телефонов со встроенными камерами и цифровыми проигрывателями. В то же время отсутствуют телефоны с крупными кнопками, большим экраном, увеличенным размером шрифта – именно такой телефон многие хотели бы купить для своих немолодых родителей, а может быть, и для себя. Неудобные, сложные продукты окружают нас, и этот круг становится все более тесным.

Подобная ситуация сложилась не только на западном рынке, о котором пишет Алан Купер, но в еще большей степени и на российском, где действуют дополнительные факторы, усугубляющие проблему.

Во-первых, дешевизна многих рабочих мест, например кассиров супермаркета или работников операторских центров (call-centres), снижает мотивацию работодателя, перестающего требовать от программных продуктов высоких эргономических характеристик. Часто работодатель нанимает несколько дополнительных сотрудников вместо того, чтобы вложить средства в повышение эффективности труда уже работающих.

Во-вторых, менталитет российского человека, с раннего возраста привыкшего бороться с неудобными в быту вещами, запрещает ему жаловаться на такие «мелочи», как нелогичность работы устройства или бессмысленность сообщений об ошибках, выдаваемых программой. Жалобы на что-то для наших соотечественников равносильны признанию в собственной слабости и глупости. Они не пеняют на сложные бытовые условия и на

неудобство тех или иных продуктов; в крайнем случае они пытаются обойти проблему или просто игнорируют ее. Это порождает скрытый спрос на приятные в использовании продукты, о наличии которого большинство разработчиков информационных систем даже не догадывается.

В-третьих, многие заказчики до сих пор живут сегодняшним днем, не задумываясь над тем, что вложения в эргономику – как в виде дополнительных требований к поставщику программных продуктов, так и в виде средств, потраченных на грамотное проектирование пользовательского интерфейса, – вернутся ощутимой прибылью. Производители, в свою очередь, предпочитают работать «по старинке» и не предлагать заказчику полноценное проектирование пользовательского интерфейса на начальном этапе работы, еще до набора команды программистов или выбора программной платформы. Нередко главная цель состоит в том, чтобы «отхватить» проект и поскорее начать его делать, надеясь, что в процессе все проблемные места уж как-нибудь сами «утрясутся». Обучать, воспитывать заказчика, помогать ему и его бизнесу в долгосрочной перспективе – непопулярное дело в среде разработчиков. И, честно говоря, не всегда благодарное. По собственному опыту знаю, что при разработке пользовательских интерфейсов значительные ресурсы расходуются именно на воспитание разработчиков и заказчиков продуктов.

Несмотря на перечисленные выше факторы в последние год-два ситуация быстро меняется: благодаря росту экономики работодателю теперь выгоднее вкладывать средства в уже работающих сотрудников, чем нанимать новых; растет искушенность российских потребителей, а следовательно, и их требования к покупаемым товарам; многие сегменты рынка ИТ развиваются, и эргономическая составляющая проектов становится конкурентным преимуществом для компаний-разработчиков.

Алан в своей неподражаемой, провокационной манере критикует устаревшие методы создания программных продуктов. Он предлагает новый, революционный способ разработки, результатом которого станут востребованные пользователями продукты. Можно без преувеличения сказать: подход Купера задает новый вектор развития индустрии ИТ – индустрии, облегчающей, а не усложняющей нам жизнь.

Первое издание книги вышло пять лет назад (1999). С этого момента многие ведущие компании, например Microsoft, SAP, Oracle, взяли предлагаемые Купером методы на вооружение. Так, Microsoft использует персонажей при разработке Longhorn.

Я надеюсь, что российский рынок ИТ также услышит голос Алана Купера.

*Алексей Копылов,
ведущий специалист компании UIDesign Group,
разрабатывающей пользовательские интерфейсы*

Предисловие

Спасайся кто может – началось вторжение компьютеров. Компьютеров, мощность которых приводит в трепет и которые решают все более важные задачи посредством неуклюжих, старомодных интерфейсов. Проникая во всевозможные сферы нашей жизни, компьютеры будут раздражать нас, приводить в ярость, а кое-кого даже убьют. В свою очередь, мы испытаем соблазн уничтожить свои компьютеры, но не посмеем, потому что и сегодня уже полностью, необратимо зависим от этих многообещающих монстров, делающих современную жизнь возможной.

К счастью, у нас есть другой выход. Необходимо коренным образом изменить представления о взаимодействии людей и машин. Изменить эти отношения на глубинном уровне, создать не существовавшие ранее подходы, ибо причина наших растущих проблем не в машинах, а в нас самих. Люди создали ненавистные интерфейсы; люди продолжают использовать некорректно функционирующие машины, мирясь с нагрузкой на глаза, создаваемой неуклюжими интерфейсами, мирясь с болью в спине и разрушением сухожилий в запястьях. Как подсказывает название книги, все мы пациенты технопсихушки, которую сами же создали.

Эта книга открывает путь к спасению. Вернее, Алан Купер показывает, что дверь на свободу широко распахнута. Мы можем уйти в любой момент, но в своем безумии мы до сих пор этого не замечали. Секрет в том, чтобы переопределить способ взаимодействия с компьютерами в более широком контексте.

Алан Купер – не просто один из пациентов, он еще и отступник, чьи идеи, вероятно, приведут в ярость тех, кто хочет держать нас под замком – инженеров, создавших ненавистные нам системы и по-прежнему считающих совершенствование интерфейсов выходом из этого кошмара. Но само понятие *интерфейса* – пережиток эры, когда компьютеры встречались редко и были слабыми, когда они почти не были способны взаимодействовать со своими хозяевами, людьми. *Интерфейсы* имели смысл, когда все взаимодействие происходило в неизведанном мире за стеклянным экраном компьютера. Сегодня же они стали просто опасными, пото-

му что мы живем в мире, где компьютеры проникают во все уголки жизни. Компьютеры более не соединяются с людьми посредством интерфейсов, но взаимодействуют с нами, и это взаимодействие будет постепенно углубляться, становиться более тонким и более критичным для нашего душевного равновесия и, в конечном итоге, выживания.

Алан Купер понимает разницу между интерфейсом и взаимодействием лучше, чем кто-либо из знакомых мне людей. Его идеи опираются на многолетний опыт в содействии разработке продуктов, элегантно и ненавязчиво проникающих в нашу жизнь. Он претворял свои идеи в жизнь много лет и теперь, наконец, нашел время превратить свой практический опыт в четкое описание задачи, с которой мы столкнулись, в методологию побега из психбольницы, которую мы с такой любовью выстроили. Читайте дальше – и обретете свободу.

*Пол Саффо (Paul Saffo),
директор Института Будущего*

Введение

Книга-обоснование

Я собирался написать совсем другую книгу – книгу-руководство о процессе проектирования взаимодействия. В мае 1997 года во время поездки в Тоскану двое моих друзей, Дон Мак-Кини (Don McKinney) и Дэйв Карлик (Dave Carlick), уговорили меня написать книгу, которую вы держите в руках. Они убедили меня, что следует, прежде всего, обращаться к деловой аудитории.

Они знали, что я хотел написать руководство, и, поощряя эту идею, все-таки выражали сомнение в необходимости книги о проектировании взаимодействия (interaction design). Они хотели, чтобы я написал книгу, убеждающую в ценности этого процесса. Аргумент, конечно, интересный, но я не был уверен, что смогу.

Однажды поздней ночью на веранде нашей общей желтой виллы над Фиренце у меня состоялся серьезный разговор с Дэйвом и Доном. На столе несколько пустых бутылок из-под кьянти и остатки хлеба, сыра и оливок. Сияют звезды, светлячки порхают над лужайкой, а вдалеке подмигивают огни древних куполов столицы Тосканы. Дэйв снова предлагает мне отложить идею книги-руководства и вместо этого «дать деловое обоснование проектированию взаимодействия».

Я энергично протестую: «Дэйв, я же не знаю, как писать такую книгу». И начинаю загибать пальцы: «Мне придется объяснять, насколько отвратителен существующий процесс разработки, как компании теряют деньги на неэффективном создании программного обеспечения, как ненадежны неудовлетворенные клиенты и как все эти проблемы может разрешить более совершенный процесс проектирования».

Дэйв перебивает меня: «Алан, это называется главами».

Эта реплика лишает меня воли. Я вдруг осознаю, что повторяю избитый сценарий и что Дэйв прав. Книга, содержащая «бизнес-обоснование»,

необходима и будет более своевременна, чем книга-руководство. Дэйв и Дон убедили меня, что я действительно способен написать такую книгу.

Инженер, сведущий в бизнесе, либо бизнесмен, сведущий в технологии

Успешный профессионал XXI века – это либо инженер, сведущий в бизнесе, либо бизнесмен, сведущий в технологии, и именно такому человеку я адресую свою книгу.

Бизнесмен, сведущий в технологии, осознает, что его успех зависит от качества доступной ему информации и изощренности, с которой он сможет эту информацию использовать. С другой стороны, инженер, сведущий в бизнесе, – это предприимчивый конструктор или ученый, специализирующийся на технологии, но при этом обладающий острой деловой хваткой и осознанием силы информации. Оба новых архетипа будут доминировать в современном бизнесе.

Всех деловых людей можно разделить на две категории: на тех, кто овладеет высокими технологиями, и тех, кто скоро покинет деловую арену. Руководящий работник уже не может делегировать обработку информации специалистам, ведь бизнес и есть обработка информации. Сегодня вы можете выделиться качеством своих систем обработки информации, но не качеством систем производства. Если вы производите что-либо, то весьма вероятно, что продукт содержит микросхему. Если вы предлагаете услугу, то, скорее всего, вы используете компьютеризованные инструменты. Попытка выявить предприятия, зависящие от высоких технологий, – столь же глупая затея, как попытка выявить предприятия, зависящие от телефона. Революция высоких технологий затронула все сферы деловой деятельности, а информация в цифровом формате стала основой вашего рабочего дня.

Кто-то сказал: «Человеку свойственно ошибаться, но чтобы провалить дело капитально, необходим компьютер». Неэффективные механические системы способны приносить убытки в пару центов на каждой производимой детали, однако из-за некачественных информационных процессов можно потерять целую компанию. Влияние на вашу компанию продуктов, основанных на программном обеспечении, как и влияние инженеров, создающих эти продукты, огромно.

Жаль, что наши цифровые инструменты сложны для изучения и понимания, сложны в применении; они часто препятствуют достижению наших целей. Мы теряем деньги, время, возможности. Будучи инженером, сведущим в бизнесе, или же бизнесменом, сведущим в технологии, вы создаете или потребляете продукты, основанные на программном обеспечении,

а скорее, потребляете и создаете одновременно. Обладание более совершенными, более простыми в освоении и применении высокотехнологичными продуктами – в ваших личных и профессиональных интересах. Более качественные продукты не требуют больших затрат времени на создание, и их создание не обходится дороже. Ирония в том, что они не должны быть сложными и таковыми являются лишь потому, что устарели процессы конструирования, и эти процессы требуют ремонта. Лишь давние традиции, основанные на заблуждениях, препятствуют сегодня получению более качественных продуктов. В книге показано, как можно требовать и получать лучшие продукты, которых вы заслуживаете.

Идея этой книги не сложна: мы можем создавать мощные и приятные продукты, основанные на программном обеспечении, используя простой прием: *проектируя* взаимодействие продукта с пользователем *до* этапа программирования. Сегодня мы этого не делаем, несмотря на распространенное мнение. Проектирование интерактивных продуктов, использующих программное обеспечение, – специализация столь же сложная, как собственно разработка.



Сделав выбор в пользу книги-обоснования и отказавшись от книги-руководства, я молю о прощении каждого проектировщика взаимодействия, которому придется читать эту книгу. Из почтения к деловой части аудиторией она содержит лишь краткие отступления в область практической методологии проектирования взаимодействий (в основном в главах части IV «Проектирование взаимодействий – выгодный бизнес»). Информации достаточно, чтобы показать, что такая методология существует, что она применима в любой предметной области и что ее преимущества очевидны всем, независимо от компетенции в технической области.

Недавно я познакомился с руководителем высокого ранга одной из крупнейших технологических компаний мира. Официальное название его должности – вице-президент по вопросам удобства использования продуктов (юзабилити), и он несет ответственность за очень многие программные продукты, как небольшие, так и крупные. Это выдающийся и состоявшийся человек, выходец из сообщества HCI (Human-Computer Interaction, взаимодействие человека и компьютера), где принят формализованный подход. Он, как и его компания в целом, искушен в «эргономике» – в тестировании и наблюдении из-за односторонних зеркал. Однако он пришел говорить со мной не о тестировании, а о проектировании, и не о пользователях, а о персонажах. Он рассказал, что в его компании полностью прекращено эргономическое тестирование (называемое в этой книге юзабилити-тестированием) на завершающей стадии разработки; вместо этого усилия прикладываются до начала разработки – при проек-

тировании. Более того, он упомянул, что все его люди, умеющие и привыкшие наблюдать за пользователями в искусственных условиях, проходят переподготовку и будут заниматься этнографическими исследованиями в «полевых» условиях.

Этот руководитель и его компания – символ тех огромных изменений, которые произошли в отрасли за пять коротких лет с момента выхода в 1999 году первого издания книги «The Inmates Are Running the Asylum» («Психбольница в руках пациентов»). Книга послужила одновременно манифестом для революции и учебником для науки. Не сосчитать, сколько писем я получил от менеджеров среднего звена, в которых рассказывалось, как после прочтения книги они покупали по экземпляру каждому из вышестоящих руководителей. Между тем разработчики программного обеспечения и университеты восприняли три главы части IV «Проектирование взаимодействий – выгодный бизнес» как исходный материал для руководства «сделай сам» по претворению в жизнь целеориентированного (Goal-Directed®) проектирования при помощи персонажей.

Я весьма признателен всем менеджерам, программистам, руководителям и практикам эргономики за то, что они воспользовались идеями этой книги и вывели удобные программы из лабораторий в жизнь, сместили фокус с тестирования на проектирование. Благодаря их усилиям полностью изменился ландшафт профессии эргономиста. Сегодня в большинстве организаций, с которыми мне приходится иметь дело, на полной ставке работает один или несколько профессиональных проектировщиков взаимодействия, и влияние этих людей на качество и поведение программных продуктов и услуг постоянно растет. Мне приятно осознавать, что эта книга способствовала их успеху.

Мне вспоминается, как я читал программную речь на конференции программистов в 1999 году, вскоре после опубликования книги. Название речи совпадало с названием книги, и речь я начал с утверждения, что «психбольница находится в руках пациентов, и эти пациенты – вы». В этот момент, когда более 2500 инженеров пытались воспринять мое обвинение, можно было услышать, как летит муха. В молчании, охватившем зал, я продолжил представление основных посылок книги, и час спустя толпа хомо логикус настолько прониклась моими аргументами, что аплодировала стоя. Удивительно, но большинство программистов с энтузиазмом восприняли идеи проектирования и проектировщиков взаимодействия. Они понимают, что нуждаются в содействии там, где речь идет о человеческой стороне конструирования программ, и счастливы получить, наконец, некоторые полезные указания. Программисты признают, что любая практика, повышающая качество и успех программ, не является для них угрозой.

В прошлом руководители считали проектирование взаимодействия задачей программирования и делегировали ее решение программистам, которые прилежно трудились, хотя их опыт, подготовка, настрой и рабочее расписание не позволяли добиться успеха. С целью диагностирования проблемы эта книга подробно описывает такой провал, который всегда оказывается провалом программиста. Некоторые из программистов обиделись на мои описания, вообразив, что я злословлю или пытаюсь возложить на программистов вину за некачественные программы. Определенно, они являются агентами в создании некачественных программ, но никоим образом не заслуживают осуждения. Я не виню программистов за сложные в использовании программы, и мне очень жаль, что у некоторых из них сложилось обратное впечатление. За некоторыми исключениями, знакомые мне программисты прилежны и добросовестны в своем желании угодить конечным пользователям, равно как и неустанны в стремлении повышать качество своих программ. Подобно пользователям, программисты – лишь жертвы неверного процесса, оставляющего слишком мало времени, дающего слишком много противоречивых приказов и слишком мало действительно ценных указаний. Мне очень жаль, если у кого-то из программистов сложилось впечатление, что я их обвиняю.

Неэластичность процесса создания программ, а в особенности высокая стоимость программирования и низкое качество взаимодействия – проблема, попросту говоря, не технического плана. Это результат применения бизнес-практик в том направлении, в котором они устарели – в разработке программ. С чистым сердцем, наилучшими намерениями и с благословения высшего руководства программисты пытаются решить эту проблему инженерным путем. Но более интенсивная или качественная работа в этом направлении не позволяет добиться успеха. Программисты ощущают все большую тщетность своих усилий, и их отчаяние нарастает.

В своих недавних путешествиях я заметил нарастающую тревогу в сообществе программистов. С прискорбием могу сообщить, что хуже всего чувствуют себя лучшие и самые опытные программисты. Они прикладывают титанические усилия, но впадают в цинизм и испытывают тоску, понимая, что их умения растрачиваются впустую. Они могут и не знать точно, как именно получается, что их квалификация не находит правильного применения, но они не могут не видеть очевидных фактов. Многие из лучших программистов вообще перестали программировать, поскольку работа раздражает их. Они ушли в преподавание, стали проповедниками, писателями, консультантами, потому что эти занятия не оставляют ощущения пустой траты времени и сил. Но этих неприятнейших потерь вполне можно избежать. (В некотором смысле движение свободных программ с открытым кодом можно назвать раем для этих отчаявшихся программистов – тут они могут писать код по своим стандартам и быть суди-

мыми только равными, не выслушивая советы и не имея необходимости терпеть вмешательство маркетологов или руководителей.)

Программистам не хватает времени, четких указаний или адекватных планов, позволяющих добиться успеха. Эти три вещи – на совести руководителей работников, именно они не дают всего этого программистам, причем вовсе не по недомыслию или злему умыслу, а по причинам, которые можно было бы обойти. Они просто не вооружены адекватными инструментами, позволяющими решать сложные и уникальные проблемы информационной эры. Ну вот, звучит так, будто я снова на кого-то нападаю, но на этот раз в мой прицел попали деловые люди, а не программисты. Повторюсь, чтобы решить проблему, ее следует сначала разобрать на составляющие. Я ищу решения, а не козлов отпущения.

Мудрый руководитель Питер Дрюкер (Peter Drucker) в свои девяносто два года, большую часть которых он провел, наблюдая и направляя руководителей, смотрит на эту проблему со своей, уникальной точки зрения. В недавнем интервью журналу «CIO» он упомянул о наивном оптимизме руководителей в пятидесятые и шестидесятые годы, когда компьютеры только пробились в деловой мир. Эти руководители думали, что компьютеры «окажут огромное воздействие на способы ведения бизнеса», но Дрюкер говорит, что «произошло совсем не это. Очень немногие руководители задавали вопрос: “Какая информация мне требуется, чтобы выполнять эту работу?”» Цифровые машины дали руководителям небывалые объемы данных, но лишь немногие поинтересовались, подходят ли эти данные для управления корпорацией. Образ существования бизнеса менялся очень быстро, однако менеджмент при этом не менялся. Дрюкер атакует наши устаревшие бухгалтерские системы, рожденные в эпоху меркантилизма, повзрослевшие в век пара и стали и угасающие на пороге XXI века, эры информации. Дрюкер утверждает: «Самая нужная вам информация – информация об окружающем мире, и этой информации у вас нет».

В последние несколько лет XX века, по мере раздувания мыльного пузыря доткомов, целые цистерны чернил уходили на рекламу, твердившую, что в Интернете существует «новая экономика». Знатки утверждали, что продажа вещей в Паутине, где магазины строят из страниц, а не из кирпичей, принципиально отличается от привычных стилей бизнеса и что «старую экономику» уже не оживить. Разумеется, почти все компании новой экономики мертвы, финансисты, поддержавшие их, пережили шок, а эксперты, пропагандировавшие новую экономику, теперь заявляют, что это была пустая мечта. Новая-преновая линия такова, что нам суждено пока оставаться со старой-престарой экономикой.

Вообще говоря, я считаю, что мы живем при новой экономике. Более того, я думаю, что доткомы никогда не были ее частью. Напротив, они стали последним вздохом старой экономики, экономики производства.

В индустриальную эру, до появления программ, продукты создавались из реальных материалов – из атомов. Затраты на добычу, плавку, приобретение, транспортировку, нагрев, формовку, сварку, окраску и снова транспортировку преобладали над всеми прочими расходами. В бухгалтерском учете эти расходы называются «переменными», поскольку они различны для каждого созданного продукта. «Фиксированные расходы», как вы, наверное, догадываетесь, очевидным образом не варьируются и включают такие затраты, как корпоративное администрирование или начальная стоимость завода.

Классические правила управления бизнесом корнями уходят в производственные традиции индустриальной эры. Увы, эти правила не учитывают новые реалии эры информационной, в которой продукты уже не создаются из атомов, а состоят в основном из программного кода, из наборов битов. А биты не подчиняются тем же экономическим правилам, что атомы.

Некоторые фундаментальные истины остаются справедливыми и для новой экономики. Цель любого бизнеса – стабильные прибыли, и есть лишь один законный способ получать их: продавать товары или услуги дороже, чем обходится их создание. Из этого следует, что есть два пути повышения прибыльности: снижение затрат или повышение прибылей. В старой экономике лучшим способом было снижение затрат. В новой экономике намного, намного лучше работает повышение прибылей.

Сегодня наиболее нужные и дорогие продукты состоят (полностью или почти полностью) из программного обеспечения. Они не требуют сырья. У них нет стоимости производства. Они не требуют затрат на транспортировку. Не требуют литья, обработки, покраски. Вот она, реальная разница между экономикой индустриальной эры и экономикой эры информационной: в информационную эру отсутствуют или невелики переменные затраты, тогда как в позднюю индустриальную эру именно эти затраты были главным фактором. Действительно, именно отсутствие переменных затрат делает новую экономику новой.

Зарплата программистов в вашем штате – фиксированные затраты или переменные? Один час работы программистов нельзя связать с одной продажей продукта – один и тот же код можно продавать много раз. Вложение в программирование можно амортизировать продажей миллионов копий продукта – точно так же, как продажа продуктов, созданных на заводе, амортизирует вложения в этот завод.

Стоимость создания программ не переменна, но и не фиксирована тоже. Разработка программ – для компании процесс непрерывный, генерирую-

щий прибыли, и это совсем не то же самое, что строительство завода. Высокооплачиваемые строители завода после завершения работ уходят на другую рабочую площадку. Программисты стоят гораздо дороже плотников и сварщиков и никогда не исчезают, потому что, по всей видимости, их работа никогда не завершается. Кто-то может сказать, что программирование – это исследования и разработка, и сходство действительно есть. Однако же исследования и разработка – это размышления и эксперименты, призванные оценить теоретическую жизнеспособность продукта, и они происходят совсем не так, как настоящее создание продуктов. Эта мысль подтверждается тем, что традиционный бухгалтерский учет разделяет исследования/разработку и ежедневную деятельность, приносящую прибыли. Создание программ не вписывается как следует ни в одну из этих категорий учета, приемлемых для прежних предприятий.

Да, этим маленьким терминологическим несоответствием можно было бы и пренебречь как софизмом, уместным в беседе счетоводов за кружкой пива, но в действительности оно оказывает огромное влияние на финансирование разработки программ, на управление проектами по разработке и, что самое важное, на то, как к разработке программ относятся высокопоставленные руководители.

Программисты создают приложения, а руководящие работники создают потоки прибылей и структурные подразделения. Программисты оценивают свой успех по качеству продукта, а руководящие работники – по прибыльности вложений. Эту прибыльность они оценивают на языке математических терминов, позволяющем учитывать фиксированные затраты, переменные издержки, затраты на корпоративное администрирование, исследования и разработку, но, к сожалению, не описывающем подходящие модели для программ и программирования. Бухучет – основной язык бизнеса, и перечисленные категории настолько фундаментальны для всех измерений и коммуникаций в бизнесе, что современные руководители полностью их усвоили. Программирование для них – еще одна статья корпоративных расходов, которую следует причислить к одной из существующих категорий. На практике большинство руководителей расценивают программирование как производственный процесс, имеющий переменные издержки. (Для целей налогообложения в большинстве компаний, создающих программные продукты, программирование проходит по статье исследований и разработок, но во всех остальных отношениях расценивается как деятельность с переменными издержками.) Это худший выбор из всех существующих, поскольку он наносит серьезный ущерб возможности принимать эффективные решения, связанные с бизнесом.

В индустриальную эру ключевым преимуществом была массовость, которая позволяла снижать цены и делать продукцию доступной многим людям. Покупатель при этом получал возможности, ранее не существовав-

шие или получаемые в результате дорогостоящей ручной работы. Компании соревновались в области продажных цен, непосредственно связанных с переменными затратами – затратами на производство и доставку. В информационную эру доступность продукции по разумным ценам считается обстоятельством само собой разумеющимся. В конце концов, программы можно распространять через Интернет – практически бесплатно и почти не прилагая усилий.

Бизнес может повышать доходность за счет увеличения прибылей или сокращения издержек. Другими словами, предприятие может увеличивать инвестиции в области фиксированных затрат, повышая качество продукции и укрепляя таким образом ценовые позиции, или же снижать переменные затраты, что означает снижение стоимости производства. В старой, «атомной» экономике снижение издержек давалось легко и было эффективным и предпочтительным. Сегодня же руководители, равняющие программирование с производством, воображают, будто снижение стоимости программирования дается так же легко и оказывается таким же эффективным. К сожалению, старые правила больше не действуют.

Поскольку производство программ сопровождается незначительными переменными издержками, снижение этих издержек не дает преимущества в бизнесе. С точки зрения бухгалтера зарплаты программистов – переменные затраты, однако в действительности их зарплаты представляют собой долгосрочные вложения, фиксированные издержки. Снижение стоимости программирования и снижение стоимости производства – разные вещи. Первое можно сравнить, скорее, с раздачей работникам дешевых инструментов, чем со снижением зарплат. Компании, заказывающие разработку в других странах с целью снижения зарплат, просто не понимают сути дела.

Более того, единственный возможный источник экономического подъема – это повышение качества и, как следствие, привлекательности продукта или услуги, а повышения качества невозможно добиться, сокращая затраты на проектирование и программирование. Правда в том, что в исследования, анализ, планирование и проектирование следует вкладывать больше времени и денег, чтобы полученный результат лучше соответствовал потребностям покупателей.

Разумеется, такой подход требует мышления, с которым не знакомы деловые люди XXI века. Им следует не снижать затраты на создание каждого объекта в отдельности, но повышать затраты на создание всех объектов в совокупности. Это сущность новой экономики, и именно об этом говорил Питер Друкер.

Современные фармацевтические компании, работающие над созданием высокотехнологичных лекарств, имеют что-то общее с новой экономикой

программного обеспечения. Действительная стоимость производства одной таблетки мизерна, однако разработка лекарства может стоить миллиарды долларов и продолжаться более десяти лет. Подъем после выхода на рынок нового волшебного лекарства может длиться до бесконечности, а вот выпуск недоработанного лекарства способен принести только катастрофический спад. Фармацевтические компании знают, что снижение издержек на разработку – нежизнеспособная стратегия.

Как и разработка лекарств, разработка программного обеспечения совсем не похожа на строительство завода. Завод – физический актив, который принадлежит компании, а работники завода в общем случае легко заменяются. Неосвязаемые, но невероятно сложные узоры мыслей, составляющие программное обеспечение, обладают ценностью только в сочетании с написавшим код программистом. Ни одна компания не может себе позволить относиться к программистам так же, как к заводу. Программисты требуют постоянного внимания и поддержки, причем гораздо больших, чем какой бы то ни было завод.

Чаще всего пытаются сэкономить на архитектуре программного продукта, а эта часть проектирования (во время которой изучаются пользователи, определяются сценарии работы, проектируется взаимодействие, определяется форма, описывается поведение) выполняется человеком. Конечно, иногда проектированию уделяют слишком большое внимание, но сокращение этой фазы точно не принесет пользы. Каждый доллар и каждый час, потраченные на архитектуру, обернутся десятикратной экономией на этапе программирования. Кроме того, вложение в достаточно качественное проектирование делает ваш продукт привлекательным, а это означает, что продукт принесет больше денег. Его привлекательность станет основой для вашего бренда, расширит возможности для повышения цен, сделает клиентов лояльными, подарит вашему продукту более долгую и насыщенную жизнь. И хотя здесь нет экономии средств, вы получаете большое преимущество в смысле качества. По иронии судьбы лучший способ увеличить прибыльность в информационную эру состоит в том, чтобы больше тратить.

К сожалению, в большинстве руководителей живет практически непобедимое стремление сокращать вложения времени и денег в программирование. Они ошибочно считают устаревшую тактику сокращения издержек подходящей и не понимают, что сокращение инвестиций в программирование оказывает сильное негативное влияние на качество, привлекательность и прибыльность продукта в долгосрочной перспективе. Разумеется, простым повышением затрат не добитесь улучшений, а часто ситуация и ухудшается, если дополнительные деньги вливаются в обход мудрости, анализа, правильного руководства. Мой первый наставник, Дэн Хоакин (Dan Joaquin), любил повторять, что правильный обратный вариант ста-

рой истины «получаешь то, за что платишь» звучит так: «не получаешь того, за что не платишь». Действия без планирования всегда чреваты риском потратить слишком много. Фокус в том, чтобы потратить нужное количество денег, и он требует значительных познаний в управлении созданием программного обеспечения. Он требует также и процессов, обеспечивающих руководителей пониманием и информацией для принятия верных решений. Дать компаниям такие процессы – цель этой книги.

В буме доткомов участвовали компании с бизнес-моделями, полностью ориентированными на снижение переменных затрат. Хотя многие доткомы рекламировали преимущества покупок через Интернет, их сайты, тяжеловесные и неудобные, выглядели бледно по сравнению с обычной поездкой в торговый комплекс. Основатели доткомов просто лучились от восторга (и пресса, кстати, тоже), потому что им удалось создать предприятия розничной торговли с невероятно низкими переменными затратами. Феерический провал этих предприятий, несомненно, доказывает, что информационной эрой правят иные экономические правила.

В старой экономике более низкие переменные затраты приводили к более широкому распространению товара и снижению розничных цен. Это двойное преимущество было выгодно непосредственно покупателям, а покупатели – фундамент экономического успеха индустриальной революции. В новой экономике успех бизнеса зависит от способности дать потребителям что-то новое и более качественное. Реальное качество каждого шага транзакции – от просмотра страниц до сравнения товаров – должно быть ощутимо более высоким для пользователя. Гораздо приятнее сделать покупку обычным путем, чем продираться через 11 экранов и в конечном итоге выяснить, что все равно придется звонить в компанию. Покупки в сети становятся совершенно ненужными и непривлекательными, если требуется набрать свое имя, свой адрес и ввести информацию о кредитной карте три или четыре раза, а затем обнаружить, что сайт не позволяет купить все необходимое и все равно придется ехать в обычный магазин, сделанный из атомов. Сегодня простое снижение цен на продукты уже не дает гарантии успеха.

Компания *Pets.com*, специализирующаяся на продаже корма для собак через Интернет, не предлагала более качественный корм, как не предлагала и шоппинг, более приятный, чем в обычном местном магазине; она не предлагала новую информацию, новые возможности, новую уверенность. Она предлагала лишь дешевую доставку, складирование и торговлю (все это переменные затраты) на сайте *Pets.com*. Компания применила классическую тактику снижения затрат, присущую экономике индустриальной эры, полностью игнорируя фундаментальные принципы новой экономики. Конечно, это было еще не первое дыхание новой экономики, но для старой это было последнее издыхание.

Я совершенно убежден, что любой товар можно продавать через Интернет успешно и прибыльно. Фокус в том, чтобы в онлайн-магазине было бы ощутимо приятнее покупать, чем в конкурирующих розничных сетях, и цена здесь – всего лишь одна из составляющих. Есть лишь один способ добиться этого: архитектуру системы следует создавать с целью максимально удовлетворить конечного пользователя. Отношение к любому аспекту проектирования и создания программного обеспечения как к производственному процессу служит причиной провала. Проектирование и программирование – попросту неподходящие цели для традиционных методов сокращения издержек. Конечно, можно потратить на создание программ слишком много времени и денег, но опасность потратить меньше необходимого гораздо серьезнее.

Скорее всего, эта опасность вам знакома и удивления не вызывает, но для большинства высокопоставленных руководителей крупных компаний нелегко сама идея. Эти руководители до сих пор работают по моделям бухгалтерского учета, вошедшим в моду в эпоху паровых машин, тогда как все аспекты жизни их компаний – функционирование, принятие решений, коммуникации и финансы – полностью зависят от программного обеспечения. Термины и понятия, которыми оперируют эти руководители, просто не принимают во внимание уникальную природу бизнеса в эпоху, когда инструменты и продукты коммерческой деятельности являются собой неосознанные переплетения битов вместо вагонов с железом. Отмечу, впрочем, что куклы из носков – идея классная.

Корпорации уже нанимают проектировщиков взаимодействия и начали применять целеориентированный подход, однако качество программных продуктов не слишком улучшилось. Более того, высокие затраты на программирование и неподатливость процесса создания программ остаются на своих местах. Почему?

Перемены невозможны, пока администраторы компаний не осознают, что проблемы с программами – это не технические сложности, а значимые для бизнеса вопросы. Наши проблемы останутся неразрешенными до тех пор, пока мы не изменим процесс и организацию.

Компании живут не только по устаревшим финансовым моделям, но еще и по негодной организационной модели. Эта модель – копия той, что имеет хождение в учебных заведениях, в ней создание программы смешивается с планированием и решением инженерных задач. Такова природа исследований. Прискорбно, что эта парадигма без объявления войны была перенесена в неизменном виде в мир бизнеса, где ей не место.

Корни всех современных производственных дисциплин уходят в доиндустриальные времена. Всех, кроме дисциплины программного обеспечения, которая возникла, когда индустриализация уже закончилась. Толь-

ко программирование произошло сразу из учебных заведений, где время исследований не ограничивалось, студенческой рабочей силы было хоть отбавляй, о прибылях вообще не говорили, а неработающая программа могла сойти за весьма удачный эксперимент. Неслучаен тот факт, что Microsoft, IBM, Oracle и другие ведущие компании-производители ПО расположены в «кампусах». Университетам не нужны прибыли, они не стараются успеть вовремя создать привлекательные и полезные продукты.

Любой бизнес, не связанный с программным обеспечением, начинается с исследований и заканчивается распространением продуктов или предложением услуг. Компания тщательно планирует время между этими двумя событиями, осознавая, что преждевременный выпуск непродуманного продукта опасен как для банковского счета, так и для ее репутации. Руководители знают, что время, размышления и деньги, вложенные в планирование, обернутся крупными дивидендами – гладким и быстрым процессом производства, популярностью и прибыльностью конечных продуктов.

Во всех других конструкторских дисциплинах инженеры создают стратегию, а ремесленники претворяют ее в жизнь. Инженеры не занимаются строительством мостов, этим занимаются другие специалисты. Только в области программного обеспечения перед инженером ставится задача создать собственно продукт. Только в области программного обеспечения перед «строителем мостов» ставится задача определить, как следует создавать продукт. Только в области программного обеспечения эти две задачи решаются не последовательно, а одновременно. Однако компании, создающие программы, похоже, не осознают существования такой аномалии. Инженерное дело и конструкторское дело так плотно пересекаются, что специалисты и руководители их не разделяют и, вероятно, не различают. Планированием любого рода здесь пренебрегают или откладывают его до тех пор, пока не станет слишком поздно. Считается нормальным откладывать решение очень сложных инженерных проблем до момента, когда уже экономически слишком накладно откатываться к фазе проектирования, потому что полным ходом пишется код для коммерческой версии продукта.

Проектирование архитектуры следует начинать на ранних стадиях инженерного планирования. Более того, именно оно должно быть движущей силой на этих стадиях, но такие разработки обычно откладываются до момента старта проекта и ведутся параллельно с созданием кода, поэтому не занимают должного места в процессе конструирования продукта. Компании нанимают проектировщиков взаимодействия и обучают эргономистов создавать персонажей, однако работа этих людей почти не влияет на стоимость разработки и качество заверченного продукта.

Решение проблемы – в руках президентов и исполнительных директоров корпораций. Делегируя решение своим техническим директорам или вице-президентам разработки, они поступают неверно. Эти достойные исполнители – технари, а проблема не имеет отношения к технике. Как сказал Дрюкер, инструменты для бухгалтерии, на которые полагаются директора компаний, попросту не отражают истинной природы этих компаний. Нельзя ведь на основе точных показаний спидометра утверждать, что автомобиль движется в нужном направлении. В мире бизнеса цифровых технологий такой подход не может быть эффективным.

Одна из серьезных проблем применения неверных методов бухучета и организации для разработки программ состоит в том, что руководители не осознают, во что обходится каждый доллар затрат на программирование. Точная система показала бы, что из каждого доллара около пятидесяти центов тратится неправильно и что еще два или три доллара требуется, чтобы исправить проблему, вызванную этим некорректным вложением средств. В любом другом бизнесе подобная статистика вызвала бы революцию, однако отрасль программного обеспечения продолжает жить в состоянии блаженного неведения.

За последние тринадцать лет фирма Соорег выступала в роли консультанта сотен компаний. Мои талантливые проектировщики создали для большинства клиентов «чертежи» продуктов, позволяющие коренным образом изменить ситуацию, но лишь немногие сумели воспользоваться всеми полученными преимуществами. В большинстве этих компаний проектирование взаимодействия и архитектуру программ считают лишь советом, в этих компаниях последнее слово всегда остается за программистами и инженерами. Ни один из президентов в этих компаниях не имеет ни малейшего понятия о том, что происходит в боксах инженеров, и потому расписание ужимается безо всякой причины. Программисты постоянно работают в условиях дефицита ресурсов и прежде всего – времени на хорошее программирование, а также времени, чтобы определить, где вообще требуется программирование. Они вынуждены защищаться, отвергая советы, и изворачиваться, общаясь со своими менеджерами.

На мой взгляд, существует два типа руководителей: инженеры и запуганные инженерами. Первые множат знакомые проблемы, поскольку их точка зрения безнадежно испорчена конфликтом интересов. Вторые множат проблемы, поскольку не умеют говорить на языке программистов. И я не имею в виду языки Java и C#. Я имею в виду, что у деловых людей и программистов нет общих инструментов и общих целей. Человек разумный делегирует человеческие проблемы хому логикус, не осознавая, что решение могло бы оказаться намного более приятным в случае применения – на исполнительном уровне – уместных финансовых и организационных моделей.

У компаний есть отличная возможность сдвинуться с мертвой точки и сосредоточить усилия на удовлетворении потребностей клиентов, а не на программах, на персонажах, а не на технологиях, на выгоде, а не на программистах. Я с нетерпением ожидаю появления просвещенного руководителя, который ухватится за эту возможность и навсегда изменит способ создания программного обеспечения, подав смелый и успешный пример.



Алан Купер,
Пало-Альто, Калифорния
www.cooper.com
inmates@cooper.com