

Стив Макконнелл

ПРОФЕССИОНАЛЬНАЯ РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ



Professional Software Development

*Shorter Schedules
Higher Quality Products
More Successful Projects
Enhanced Careers*

Steve McConnell

◆◆ Addison-Wesley



ПРОФЕССИОНАЛЬНАЯ РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

*Сокращение сроков
Повышение качества продукта
Больше удачных проектов
Расширение возможностей
успешной карьеры*

Стив Макконнелл



*Санкт-Петербург — Москва
2007*

Серия «Профессионально»

Стив Макконнелл

Профессиональная разработка программного обеспечения

Перевод В. Агапова

Главный редактор	<i>А. Галунов</i>
Зав. редакцией	<i>Н. Макарова</i>
Научные редакторы	<i>А. Сапегин, О. Цилюрик</i>
Редактор	<i>В. Овчинников</i>
Художник	<i>В. Гренда</i>
Корректор	<i>О. Макарова</i>
Верстка	<i>Д. Орлова</i>

Макконнелл С.

Профессиональная разработка программного обеспечения. – Пер. с англ. – СПб.: Символ-Плюс, 2006. – 240 с., ил.
ISBN 5-93286-085-5

Стив Макконнелл, автор бестселлера «Совершенный код», других книг и многочисленных статей о разработке ПО, убедительно показывает, что разработка ПО может быть стабильно успешной, если сделать совершеннее саму профессию разработчика ПО. Он не только показывает, почему и как отрасль пришла к своему современному состоянию, и описывает шаги, которые должен предпринять каждый, кто хочет подняться на новый уровень в создании ПО. Он также говорит о корпоративных методиках, призванных увеличить количество профессионально выполненных проектов, и о лицензировании организаций и академических учебных программ как о средстве повышения профессионализма и отдельных разработчиков, и в индустрии ПО в целом.

ISBN 5-93286-085-5

ISBN 0-321-19367-9 (англ)

© Издательство Символ-Плюс, 2006

Authorized translation of the English edition © 2004 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 31.08.2006. Формат 70х90^{1/16}. Печать офсетная.

Объем 15 печ. л. Тираж 3000 экз. Заказ N

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

*На высокую башню можно подняться только
по винтовой лестнице.*

ФРЕНСИС БЭКОН

*К успеху придет только тот, кто способен преодолевать
неудачи не теряя энтузиазма.*

УИНСТОН ЧЕРЧИЛЬ



Об авторе

Стив Макконнелл – первое лицо Construx Software, где он возглавляет работы по инженерии ПО и ведет занятия в рамках программы профессионального развития фирмы. Стив пишет книги и статьи. Он автор «Code Complete» (1993 г.)¹, «Rapid Development» (1996 г.) и «Software Project Survival Guide» (1998 г.). Его книги были дважды удостоены премии Джолта (Jolt Excellence Award) журнала *Software Development* в номинации «Книга года в области разработки ПО». В 1998 г. читатели этого журнала назвали Стива Макконнелла одним из трех самых влиятельных людей в отрасли ПО наряду с Биллом Гейтсом и Линусом Торвальдом. В 1998–2002 гг. Макконнелл являлся главным редактором журнала *IEEE Software*. Он вице-председатель Комитета профессиональных методик в Компьютерном обществе IEEE и входит в комитет экспертов проекта SWEВОК (Software Engineering Body of Knowledge, область знаний инженерии ПО).

Степень бакалавра Макконнелл получил в Колледже Уитмана, а степень магистра инженерии ПО – в Университете Сиэттла. Живет в Беллвью, штат Вашингтон.

Если у вас есть замечания или вопросы по этой книге, пишите Стиву Макконнеллу на stevemcc@construx.com или свяжитесь с ним через его сайт www.stevemccconnell.com.

¹ Стив Макконнелл «Совершенный код». – Пер. с англ. – СПб.: Питер, 2006.



Оглавление

Об авторе	6
Благодарности	12
Введение	14
Часть I Смоляная яма программного обеспечения.....	21
<i>Глава 1 Динозавры в смоляной яме</i>	<i>23</i>
<i>Глава 2 Ложное золото</i>	<i>27</i>
Перемещение каменных глыб	27
Каменные глыбы и программное обеспечение.....	30
Сначала напишем, потом исправим ошибки.....	31
Ориентир – качество	34
Иногда «ложное золото» оказывается серебром	36
Программное обеспечение – это не пластилин.....	38
К каким выводам приводит существование «ложного золота»	40
<i>Глава 3 «Культ карго» в разработке ПО.....</i>	<i>41</i>
Самозванцы от ПО	42
«Культ карго» в разработке ПО.....	43
Суть спора	44
<i>Глава 4 Разработка ПО – это не компьютерная наука.....</i>	<i>46</i>
«Есть» и «должно быть»	46
Инженерия и наука.	47
Что стоит за модным словечком?.....	49
Правильные вопросы.....	52

<i>Глава 5</i>	<i>Объем знаний</i>	53
	Суть и случайность.....	54
	Формирование устойчивого ядра.....	55
	Область знаний инженерии ПО.....	58
	Ставим зарубку.....	62
<i>Глава 6</i>	<i>Новый органон</i>	63
	Формирование профессии.....	65
	В поисках профессии инженерии ПО.....	66
	Проход через Геркулесовы столпы.....	72
Часть II	Индивидуальный профессионализм	73
<i>Глава 7</i>	<i>«Предпочтение отдается сиротам»</i>	75
	Характеристики типа личности по Майерс-Бриггс.....	76
	Результаты теста MBTI разработчиков ПО.....	77
	Личные качества великих изобретателей.....	78
	Полная и абсолютная отдача.....	80
	Демография ПО.....	82
	Образование.....	83
	Перспективы занятости.....	85
	Герои и узурпаторы программирования.....	86
	Культ личности.....	87
<i>Глава 8</i>	<i>Формирование сознательного отношения к ПО</i>	89
	Нет удовлетворения.....	90
	Возлюби тех, с кем работаешь.....	92
	Насколько вы опытни?.....	92
<i>Глава 9</i>	<i>Формирование сообщества</i>	94
<i>Глава 10</i>	<i>Архитекторы и строители</i>	98
	Стратификация профессии.....	98
	Специализация функций.....	100
	Специализации в коллективе.....	103
	Время покажет.....	104
<i>Глава 11</i>	<i>Программист пижущий</i>	105

Часть III	Организационный профессионализм	109
<i>Глава 12</i>	<i>Золотая лихорадка ПО</i>	<i>111</i>
	Золотая лихорадка в ПО	112
	Разработка после «лихорадки»	113
	Смысл и бессмыслица экономики золотой лихорадки	115
	Расширение и сжатие	116
	Назад к «золотой лихорадке»	117
<i>Глава 13</i>	<i>Необходимость совершенствования методик разработки ПО</i>	<i>118</i>
	Состояние на практике	119
	Выигрыш от совершенствования практических методик разработки ПО	120
	Показатели ROI для отдельных методик	122
	Что дает анализ бюджетирования ПО	122
	Косвенный выигрыш от улучшения практических методик	124
	Взгляд на лучших	124
	Суть вызова – организационная	125
	Последний великий рубеж	126
	Десять трудных вопросов	127
<i>Глава 14</i>	<i>Птолемеёво мышление</i>	<i>128</i>
	Обзор подхода SW-CMM	129
	Движение вверх	130
	Все риски, с которыми можно справиться	132
	Кто применяет SW-CMM?	133
	Бездушная разработка ПО	134
	Серьезная самоотдача	135
	Рейтинг организаций	136
	Форма и содержание	137
<i>Глава 15</i>	<i>Количественное выражение факторов, связанных с персоналом</i>	<i>139</i>
	Факторы персонала	139
	Слабосильные программисты	141
	Физические условия	142
	Мотивация	142

Опытность персонала	144
Что в итоге	144
<i>Глава 16 Программа профессионального развития фирмы Construx</i>	145
Области знаний в Construx	146
Уровни способностей	147
Ступени лестницы профессионального развития	149
Развитие карьеры на основе продвижения по лестнице	151
Требования СКА для различных уровней способностей	153
Выводы, сделанные по результатам лестницы профессионального развития	157
Преимущества лестницы профессионального развития	160
Использование лестницы профессионального развития в других компаниях	161
Часть IV Индустриальный профессионализм	163
<i>Глава 17 Построение профессии</i>	165
Необходимость инженерии	165
Искусство и инженерия	167
Инженерные дисциплины достигают зрелости	169
Наука для разработки ПО	171
Зов инженерии	173
<i>Глава 18 Школа жизни</i>	174
Подготовка профессиональных инженеров	176
Первые шаги	178
Аттестация	180
Конструирующие программисты или программирующие инженеры?	181
Полировка жетона	183
Некоторые перспективы	184
<i>Глава 19 Кому нужны дипломы?</i>	185
Сертификация	185
Лицензирование	186
Возможно ли лицензирование инженеров ПО	189
Правильна ли сама идея лицензирования?	191
Раскрутка лицензирования	194

Ваша ставка	195
Как заслужить диплом	197
Три пути	198
Вонючие дипломы или стальное колечко?	200
<i>Глава 20 Кодекс профессионала</i>	<i>201</i>
Кодекс для кодировщиков	202
Преимущества этического кодекса поведения	205
Достижение совершеннолетия	207
<i>Глава 21 Алхимия</i>	<i>208</i>
Зачем передавать технологии практикам	208
Распространение инноваций	210
Пропать	211
Несколько жестких вопросов	212
В чем риск?	213
Опыт работы представителей на местах по программе расширения консультационной деятельности в сельском хозяйстве США	216
Принижающая роль прогресса	218
<i>Библиография</i>	<i>220</i>
<i>Алфавитный указатель</i>	<i>229</i>



Благодарности

Хотел бы поблагодарить многих специалистов, приславших свои замечания по ключевым разделам книги, среди которых Дон Багерт (Don Bagert), Джон Бентли (Jon Bentley), Стивен Блэк (Steven Black), Роберт Бернс (Robert C. Burns) из компании «Boeing», Тревор Берридж (Trevor BurrIDGE), Аугусто Коппола (Augusto Coppola), Алан Корвин (Alan B. Corwin) из «Process Builder», Райан Флеминг (Ryan Fleming), Пэт Форман (Pat Forman), Роберт Гласс (Robert L. Glass) из «Computing Trends», Дэвид Гудман (David Goodman), Оуейн Гриффитс (Owain E. Griffiths), Брейди Хонсингер (Bradey Honsinger), Ларри Хьюз (Larry M. Hughes) из компании «Sprint», Роберт Ли (Robert E. Lee), Эйвонелл Ловхог (Avonelle Lovhaug), Марк Лутц (Mark Lutz), Стив Мэттингли (Steve Mattingly), Грант Мак-Лахлин (Grant McLaughlin), Брайан Мак-Лин (Brian P. McLean), Хэнк Мьюрет (Hank Meuret), Х. Фернандо Наведа (J. Fernando Naveda), Энтон Пэнг (Anthon Pang), Дэвид Парнас (David L. Parnas), Мэтт Пелоквин (Matt Peloquin), Том Рид (Tom Reed), Кейти Роуд (Kathy Rhode), Стив Ринн (Steve Rinn), У. Пол Роджерс (Wm. Paul Rogers), Джей Силвермен (Jay Silverman), Андре Синтцофф (Andrй Sintzoff), Тим Старри (Tim Starry), Стив Токи (Steve Tockey), Леонард Трипп (Leonard L. Tripp), Том Вентцер (Tom Ventser) из группы «DMR Consulting Group», Карл Вигерс (Karl Wiegiers) и Грег Уилсон (Greg Wilson).

Мои благодарности также многочисленным рецензентам, высказавшимся по отдельным конкретным вопросам.

Особо хочу поблагодарить великолепный коллектив по подготовке книги издательства Addison-Wesley: Майка Хендриксона (Mike Hendrickson), Ребекку Гринберг (Rebecca Greenberg), Эйми Флейшер (Amy Fleischer), Кэрин Хансен (Karin Hansen) и Дженис Оуенс (Janis Owens). Рабо-

тать с каждым из них одно удовольствие, и книга стала значительно лучше в результате их труда.

Я также высоко ценю опыт сотрудничества при подготовке первого издания этой книги – «After the Gold Rush». Хотел бы напомнить об огромной работе редактора проекта Виктории Тульман (Victoria Thulman) и других сотрудников издательства: Бена Райана (Ben Ryan), Роба Нанса (Rob Nance), Черил Пеннер (Cheryl Penner) и Полы Горелик (Paula Gorelick).



Введение

Кажется, что это просто... пока не попробуешь.

Из журнала IEEE SOFTWARE¹

Я сидел в самолете, стоявшем на взлетной полосе, когда прозвучало объявление капитана: «У нас неполадки в системе кондиционирования самолета. Эта система поддерживает уровень кислорода на борту, поэтому она должна заработать раньше, чем мы взлетим. Перезапуск кондиционеров не удался, поэтому мы сейчас выключим и снова включим электропитание. *Знаете, все эти новые самолеты управляются компьютерами, поэтому они не слишком надежны.*»

Пилот выключил и снова включил питание – по сути «перезагрузил» самолет, и рейс продолжился без происшествий. Нечего и говорить, что по окончании воздушного путешествия я с большой радостью вышел из самолета.

Лучшие времена и худшие

Лучшие разработчики ПО ведут свои проекты так, чтобы обеспечить достижение целевых показателей качества. Они точно планируют сроки сдачи ПО на месяцы и годы вперед. Проекты разработки ПО укладываются в выделенный бюджет, и производительность таких разработчиков постоянно растет. Моральный дух их персонала высок, и клиенты очень довольны.

¹ Из книжного обзора, посвященного [137].

- Телекоммуникационной компании понадобилось изменить около 3 тысяч строк в базовом ПО объемом примерно в 1 000 000 строк. Изменения были внесены столь тщательно, что через год работы не обнаружилось ни одной ошибки. Время, которое потребовалось для внесения изменений, включая анализ требований, планирование, реализацию и тестирование, составило 9 часов [110].
- Группа разработчиков ПО для ВВС США взялась реализовать некий проект за год с бюджетом \$2 000 000, хотя другие вполне достойные разработчики предлагали срок до 2 лет при бюджете до \$100 000 000. Когда же эта группа сдала ПО на месяц раньше срока, менеджер проекта заявил, что успех достигнут за счет методик, известных уже несколько лет, но редко применяемых на практике [49], [131].
- Авиастроительная компания разрабатывает ПО для клиентов по фиксированной цене, при этом только 3% ее проектов превышают сметную стоимость; 97% из 100 укладываются в бюджет.¹
- Организация, твердо следующая политике достижения исключительного качества ПО, в течение 9 лет добивалась ежегодного снижения на 39% количества дефектов, обнаруживаемых после выпуска версий; итоговое снижение составило 99% [56].

Вместе с огромными успехами, примеры которых приведены выше, отрасль ПО приносит в экономику миллиарды долларов как за счет прямых продаж самого ПО, так и в результате повышения эффективности и производительности, а также создания продуктов и услуг, которые возможны только при использовании соответствующего ПО.

Методики, необходимые для создания качественного программного продукта, известны уже 10, а то и 20 лет. Тем не менее, несмотря на впечатляющие достижения, отрасль ПО не использует весь свой потенциал. Между передовыми разработчиками и общей массой существует огромный разрыв, а многие широко применяемые методики сильно устарели и не обеспечиваются достаточными ресурсами. Эффективность среднего проекта ПО оставляет желать лучшего, о чем свидетельствуют многие хорошо известные провалы.

- Налоговая служба США провалила программу модернизации ПО стоимостью \$8 000 000 000, что обошлось в \$50 000 000 000 несобранных доходов в год [3].

¹ Из разговора с автором.

- Улучшенная АСУ Федерального управления авиации (FAA) превысила выделенный бюджет примерно на \$3 000 000 000 [17], [53], [48].
- Неполадки в системе обработки багажа привели к задержке открытия международного аэропорта в Денвере более чем на год. Потери оцениваются в \$1 100 000 в день [48], [53].
- Ракета «Ариан-5» взорвалась при первом пуске из-за ошибки в ПО [99].
- Бомбардировщик «Б-2» также не взлетел с первого раза из-за проблем с ПО [44].
- Управляемые компьютером паромы в г. Сиэттл (штат Вашингтон) около полутора десятка раз врезались в доки, нанеся ущерб на сумму свыше \$7 000 000. Власти штата рекомендовали выделить более \$3 000 000 на перевод паромов обратно на ручное управление [98].

Подобным ошибкам подвержены и другие проекты. Около четверти из них терпят полную неудачу с самого начала [132], [66]. Очень часто к моменту сворачивания проекта выявляется двукратный перерасход бюджета. Примерно половина всех проектов либо затягивается, либо превышает сметную стоимость, либо обеспечивает меньше функциональных возможностей, чем предусматривалось [132].

Для предприятий такие свернутые проекты означают упущенные возможности: если бы закрытие проекта обходилось в 10% выделенных средств, а не в 200%, нетрудно представить, чего можно было бы добиться, просто перенаправив эти ресурсы в проекты, которые были завершены.

На национальном уровне отмененные проекты представляют собой чудовищную и бесполезную трату сил и ресурсов. Грубые подсчеты показывают, что свернутые проекты ПО обходятся экономике приблизительно в \$40 000 000 000.¹

¹ Этот грубый расчет основан на статистике занятости, представленной в табл. 7.2 «Структура занятости работников сферы ПО» по должностям ученых компьютерно-информационной отрасли, проводящих исследования; программистов-компьютерщиков; инженеров прикладного ПО; инженеров системного ПО и аналитиков компьютерных систем. Другие должности в этом анализе не учитывались. Итоговые расходы экономики США на разработку ПО рассчитывались путем умножения средних совокупных расходов на оплату труда (\$95 000) на 1 741 000 работников этих должностей. Четверть итоговой суммы в \$160 000 000 000 25% – это доля, затраченная на отмененные проекты. В этом анализе может не учитываться влияние отмененных проектов, поскольку риск отмены увеличивается с ростом объема проекта, поэтому отмененные проекты могут оказаться более дорогостоящими по сравнению со средними расчетами.

Но и успешные проекты могут представлять угрозу безопасности и общественному благополучию. Руководителю проекта в Lotus звонил хирург, который использовал электронную таблицу для анализа состояния пациента во время операции на открытом сердце [142]. В журнале «Ньюсвик» публиковались фотографии военных, планирующих военные операции при помощи Microsoft Excel на своих переносных компьютерах; группа технической поддержки Excel принимала телефонные звонки с поля боя во время активных военных действий.

Цель данной книги

Процесс разработки ПО можно сделать прогнозируемым, контролируемым, экономичным и управляемым. Обычно разработка ПО ведется иначе, однако есть все возможности делать именно так. Эта книга посвящена зарождающейся профессии – инженерии ПО (т.е. технологии разработки ПО) и практической профессиональной методологии, которая обеспечивает экономичное создание высококачественного программного обеспечения.

В книге обсуждаются следующие вопросы:

- Что такое инженерия разработки ПО?
- Как инженерия ПО связана с компьютерной наукой, т.е. с наукой о вычислительных системах?
- Почему обычного программирования недостаточно?
- Зачем нужна *профессия* инженерии разработки ПО?
- Почему *инженерия* является наилучшей моделью профессиональной разработки ПО?
- В чем отличия эффективных методик, используемых в различных проектах (или различных компаниях), и какие принципы практически одинаковы?
- Что могут предпринять организации, чтобы поддержать профессиональный подход к разработке ПО?
- Что нужно сделать индивидуальным разработчикам ПО, чтобы стать профессионалами?
- Что могут предпринять представители отрасли разработки ПО в целом, чтобы создать настоящую профессию «инженерия ПО»?

Структура книги

Материал книги постепенно переходит от рассмотрения программирования как ремесла в его сегодняшнем состоянии к изучению технологии ПО как профессии, которая может сформироваться в будущем.

Часть I «Смоляная яма программного обеспечения» содержит рассказ о том, как отрасль эволюционировала к своему нынешнему состоянию. Этот процесс определялся многими факторами, понимать которые необходимо, для того чтобы ускорять, а не замедлять наступление перемен, призванных сделать успешные проекты ПО повседневной реальностью.

Часть II «Индивидуальный профессионализм» рассматривает шаги, которые специалист может сделать самостоятельно для достижения высокого профессионального уровня в разработке ПО.

Проекты ПО настолько сложны, что многие ключевые факторы невозможно обсудить на уровне индивидуального разработчика. В части III «Организационный профессионализм» представлены организационные методики, необходимые для поддержания более высокого профессионализма в программных проектах. Часть IV «Индустриальный профессионализм» рассматривает меры, которые должны быть предприняты в масштабе отрасли, чтобы обеспечить профессиональный подход на индивидуальном и организационном уровнях.

У этой книги есть партнерский сайт, www.construx.com/profession, на котором размещены материалы, связанные с этой книгой, включая списки профессиональной литературы для чтения, планы самообучения, описание существующих инициатив по сертификации и лицензированию, а также ссылки на многие другие полезные сайты.

Что я узнал с 1999 г.

Книга «Профессиональная разработка программного обеспечения» представляет собой обновленный и значительно расширенный вариант моей книги, вышедшей в 1999 г. [86]. С 1999 г. я усвоил несколько положений, которые нашли отражение в моей новой книге.

- Вопрос лицензирования разработчиков ПО оказался более спорным, чем я ожидал. Я по-прежнему считаю, что лицензирование небольшого количества инженеров-программистов – это важная часть защиты жизнедеятельности людей и их безопасности. Я старался разъяснить, что лицензирование представляет собой всего лишь

одну из множества инициатив, направленных на повышение профессионализма разработчиков ПО, и не самая важная.

- Образовательную подготовку инженеров-программистов не обязательно жестко увязывать с лицензированием. Программы подготовки на младших и выпускных курсах могут быть направлены на формирование инженерного склада мышления у разработчиков ПО, но при этом не обязательно на их подготовку к получению лицензии профессионального инженера. Если лицензию в конечном итоге получают менее 5% разработчиков ПО, что кажется вполне вероятным, то ориентация большинства программ подготовки и обучения на получение обучающимися лицензии представляется неразумной.
- Мир не рухнул первого января 2000 г., когда считалась актуальной угроза масштабных сбоев в работе компьютерных систем (я не думаю, что проблемы, связанные с наступлением 2000 г., будут катастрофическими). Мрачные прогнозы не подтвердились. Более того, сама проблема Y2K была в определенном смысле вызвана *успешной* технологией разработки ПО. Она не возникла бы, если бы столь многие системы ПО не просуществовали значительно дольше, чем предполагалось.
- Современные разработки ПО действительно во многом впечатляют, поэтому любые рассуждения о профессионализации отрасли должны учитывать ее успехи. Необходима чрезвычайная осторожность, чтобы в попытках усилить слабые стороны процесса не отказаться от проверенных и успешных методик.

Кому адресована эта книга

Тем, для кого *разработка ПО служит источником средств к существованию*, эта книга даст представление о шагах, которые следует предпринять, чтобы стать настоящим профессионалом в этой области.

Менеджеры проектов разработки ПО найдут здесь свод отличительных особенностей, по которым хорошо управляемые проекты ПО можно отличить от проектов, управляемых плохо, а также обзор методов, которые могут сделать проекты более успешными.

Руководителям организаций-разработчиков ПО эта книга укажет преимущества и выгоды системного подхода к разработке ПО, а также действия, необходимые для их реализации.

Студенты, которые хотели бы работать в области создания ПО, познакомятся здесь с основами технологии ПО и получат представление о карьере в этой отрасли.

На пути к профессиональной разработке ПО

Исследователи, специализирующиеся в этой области, давно указывали, что производительность компаний, конкурирующих в одной отрасли, может отличаться более чем в десять раз [91]. Последние исследования показали, что разница может достигать и 600 раз [145]. Самые эффективные компании действительно процветают.

Выгоды формирования настоящей профессии разработчика ПО весьма убедительны. Согласно традиционным воззрениям, наибольший риск представляют перемены. Если же говорить о программном обеспечении, то наибольший риск представляет как раз их отсутствие и погружение в болото порочных, экстравагантных технологий разработки ПО вместо принятия методов, давно уже доказавших свою практическую эффективность.

Как реализовать такой переход? Это и есть главная тема книги.

*– Беллвью, штат Вашингтон.
День Поминовения, 2003 г.*

ЧАСТЬ ПЕРВАЯ

Смоляная яма программного обеспечения

ГЛАВА ПЕРВАЯ

Динозавры в смоляной яме



*Тот, кто не хочет прибегать к новым средствам,
должен ожидать новых бед.*

ФРЕНСИС БЭКОН

Фредерик Брукс еще в 1975 г. сравнил разработку крупных систем программного обеспечения с борьбой динозавров, мамонтов и саблезубых тигров с засасывающей липкой смолой в яме [21], предсказав на долгие годы вперед сохранение подобной ситуации в проектировании и разработке ПО, когда одну ногу можно освободить, только завязнув другой ногой.

Проблемы, которые Брукс выявил почти тридцать лет назад, уже тогда не были новы, так что у разработчиков ПО было более четверти века, чтобы поработать над ними. Так далеко ли удалось продвинуться с тех пор?

Многие проблемы, буквально преследующие почти любой проект разработки ПО, практически не изменились. К примеру, вопрос сроков выполнения остается типичным и для проектов сегодняшних дней. По некоторым оценкам примерно 75% средних и свыше 90% крупных проектов ПО сталкиваются с проблемами жестких сроков.¹ При этом сверхурочные переработки программистов являются скорее правилом, чем исключением.² Хорошо известно, что современные зарождающиеся предприятия, работающие в области ПО, ожидают от своих сотрудников значительной сверхурочной работы, а рассказы о программистах, засыпающих ночью

¹ Данные о статистике напряженности в связи со сроками выполнения взяты из работы [64]. Данные о типичном соблюдении сроков сдачи проектов ПО имеются в работах [64], [66], [132].

² Этот вопрос подробно обсуждается в книге автора [83].

за своими дисплеями, можно услышать почти всюду [18]. Однако еще в середине 60-х годов XX века в одном из исследований утверждалось, что «многие программисты, не укладываясь в сжатые сроки, просиживают ночи на рабочих местах» [22]. В 1975 г. Фредерик Брукс указывал, что «из-за нарушения сроков исполнения срывается куда больше проектов, чем в силу всех остальных причин вместе взятых» [19]. Так что превышение сроков исполнения проектов наблюдается уже более тридцати лет, а если учесть, что люди нетерпеливы по природе, то и значительно дольше.

Размах сегодняшних проектов ПО кажется беспрецедентным, что, разумеется, наталкивает их создателей на мысль, что никто и никогда не сталкивался с проблемами подобного масштаба. Тем не менее даже у гигантских проектов, вроде Windows NT, были предшественники. Объемы современных крупных проектов разработки ПО действительно впечатляют, причем каждый разработчик естественно полагает, что он первый берется за столь масштабный проект. Однако, как я уже сказал, прецеденты существовали и ранее. Например, первоначальный проект Windows NT потребовал около полутора тысяч человеко-лет трудозатрат,¹ однако разработка OS/360 фирмы IBM, завершенная в 1966 г., оказалась в три раза более трудоемкой.²

Данные последних исследований показывают, что наиболее часто встречающиеся причины неудач проектов по разработке ПО связаны с проблемами их соответствия заданным техническим условиям, которые выстраивают неверную систему. Эти требования слишком размыты, чтобы их можно было точно реализовать, а некоторые из них меняются настолько часто, что вносят полнейшую неразбериху в системный проект [24], [132]. Но все же проблемы с соответствием требованиям также далеко не новы. Еще в 1969 г. Роберт Фрош (Robert Frosch) заметил, что система «может удовлетворять букве технических условий, но тем не менее не являться целиком и полностью удовлетворительной» [47].

Большинство разработчиков ломают голову, пытаясь поспеть за стремительными изменениями, которые происходят благодаря развитию Интернета. Как угнаться за новыми языками, меняющимися стандартами и предложениями новых программных продуктов? Для тех, кто работает

¹ Это примерная сумма. Оценка основана на опубликованных данных о стоимости разработки Windows NT (\$150 000 000) и совокупных расходах на оплату труда в \$100 000 на человека в год [146].

² Трудозатраты составили примерно 5000 человеко-месяцев [19].