

По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru-Книги России» единственный легальный способ получения данного файла с книгой «FreeBSD. Подробное руководство» (ISBN 5-93286-066-9) – 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» (www.symbol.ru), где именно Вы получили данный файл.

The Peopleware Papers

Notes on the Human Side of Software

Larry L. Constantine



PRENTICE HALL

Человеческий фактор *в программировании*

Ларри Л. Константин



Санкт-Петербург — Москва
2004

Серия «Профессионально»

Ларри Л. Константин

Человеческий фактор в программировании

Перевод Ю. Асотова

Главный редактор	<i>А. Галунов</i>
Зав. редакцией	<i>Н. Макарова</i>
Научный редактор	<i>А. Михайлов</i>
Редактор	<i>Ю. Кунивер</i>
Верстка	<i>Н. Макарова</i>
Корректор	<i>С. Беляева</i>

Константин Л.

Человеческий фактор в программировании. – Пер. с англ. – СПб: Символ-Плюс, 2004. – 384 с.
ISBN 5-93286-044-8

Хорошее программное обеспечение создается людьми. Так же как и плохое. Именно поэтому основная тема этой книги – не аппаратное и не программное обеспечение, а человеческий фактор в программировании (peopleware). Первое издание «Constantine on Peopleware» признано классическим трудом в области информационных технологий. Новая книга Ларри Константина включает все 52 легендарные статьи из предыдущей книги и 25 новых эссе.

Peopleware охватывает все аспекты, связанные с ролью людей в разработке программного обеспечения. Это качество и продуктивность, модели и методы, динамика поведения коллектива, руководство проектами, разработка интерфейсов и взаимодействие между человеком и компьютером, психология и процессы мышления. В данное издание включены два новых раздела, посвященных организационной культуре и юзабилити программных продуктов.

ISBN 5-93286-044-8

ISBN 0-13-060123-3 (англ)

© Издательство Символ-Плюс, 2004

Original English language title: The Peopleware Papers by Larry L. Constantine, Copyright © 2001, All Rights Reserved. Published by arrangement with the original publisher, Pearson Education, Inc., publishing as PRENTICE HALL.

Название оригинала на английском языке: The Peopleware Papers by Larry L. Constantine, Copyright © 2001, 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 04.11.2003. Формат 70x100^{1/16}. Печать офсетная.

Объем 24 печ. л. Тираж 2000 экз. Заказ N

Отпечатано с диапозитивов в Академической типографии «Наука» РАН
199034, Санкт-Петербург, 9 линия, 12.

Оглавление

Предисловие	10
Предисловие к первому изданию	13
Часть I. Групповая разработка	17
1. Решения, решения	21
2. Консенсус и компромисс	25
3. Достижение консенсуса.	29
4. Скромный и высокопоставленный писарь	34
5. Официальное пространство	40
6. Раздражающие прерывания	45
Часть II. Ковбои и ковгерлы	49
7. Кодеры-ковбои.	53
8. Возвращение блудного ковбоя.	58
9. Единство в разнообразии.	63
10. Кодеры-ковбои и программисты-мудрецы	68
Часть III. Организация работы	77
11. Традиционная тактика	81
12. Методы хаоса	86
13. Открытая архитектура	92
14. Синхронное плавание	97
15. Командная политика	101
16. Все сразу	105
17. Заговор упрямцев	109

Часть IV. Инструменты, модели и методы	113
18. CASE и познание	117
19. Вопросы моделирования	122
20. Свет мой, зеркальце	126
21. Методичное сумасшествие	130
22. Говоря по существу	134
23. Будущие формы	139
24. Цели программного обеспечения	144
25. Шито белыми нитками	149
Часть V. Совершенствование процесса	157
26. Преимущества видимости	161
27. Повторение и вознаграждение	166
28. Суперобучение	172
29. Вверх по водопаду	176
30. Своевременная поставка	180
31. Под давлением	184
32. Re: Архитектура	188
33. Пошаговое улучшение качества	192
Часть VI. Юзабилити программного обеспечения	203
34. Согласованность и условности	207
35. Сложность и прогрессирующий функционализм	212
36. Назад к истокам	217
37. Цветной язык	222
38. Совершенствующиеся середнячки	227
39. Пригодны ли вы	232
40. Редактирование интерфейсов	237
41. Сервис	241
Часть VII. Удобные объекты	245
42. Объекты, которые раздражают	249
43. Глубокое понимание	256
44. Абстрактные объекты	264

45. Новая среда	270
46. Полезные ситуации	277
47. Эффективные объекты	284
48. Связанные объекты	291
Часть VIII. Это превосходное новое программное обеспечение . . .	297
49. Высокомерное программирование	301
50. Интерфейсы разнообразные	306
51. Мастеры	310
52. Образы будущего	314
Часть IX. Культура и качество	319
53. Культурное изменение	323
54. Агенты изменения	327
55. Встроено самое лучшее	331
56. Заметки из итальянского ресторана	336
57. Наставничество	341
58. На обучение	345
59. Одаренные программисты	350
60. Иконы отрасли	354
61. Импресарио	359
Приложение. Зарегистрированный Peopleware	363
Библиография	367
Алфавитный указатель	372

Посвящение

*Памяти Денниса Дэви и Кена МакКензи –
и всех, кто открыл путь*

Благодарности

Ни одна книга не является результатом труда одного человека, а эта, с ее упором на сотрудничество и совместную инициативу, с самого начала создавалась совместными усилиями. В ее основе лежат написанные мною статьи, которые появились благодаря настойчивости и дальновидности моих редакторов, Ларри О'Брайэна (Larry O'Brien) и Мэри Лензи (Marie Lenzie). Ларри заманил меня в мир постоянных крайних сроков, предложив лучшее место на страницах журнала *Computer Language Magazine* для публикации статей обо всем, что меня интересует. Мэри привела меня в журнал *Object Magazine*, где среди знатоков процессов и технологий я был чужаком, ориентированным на человеческий фактор. И Ларри, и Мэри были не только хорошими редакторами, но и моими хорошими друзьями в течение многих лет. Усилия Ларри поддержала группа талантливых и очень терпеливых людей из Miller Freeman Publications, включая Гретхен Бэй (Gretchen Bay), Николь Фримэн (Nicole Freeman), Мишель Гэхи (Michele Gehee) и Николь Кларо (Nicole Claro), которые справились с причудами моего стиля и смогли снести мои порой невыносимые лекции по электронной почте.

Выражаю свою благодарность Полу Петралии (Paul Petralia), редактору издательства Prentice Hall, за энтузиазм, с которым он вдохнул новую жизнь в старый проект. За последние штрихи профессионализма в рукописи я благодарен Мэри Судул (Mary Sudul), которая была усердным выпускающим редактором этой книги, и Хэриет Телем (Harriet Tellem), внесшей существенный вклад в первое издание.

В список редакторов, которые помогали мне в подготовке книги, я добавил бы имя моего лучшего критика и советчика, Люси Локвуд (Lucy Lockwood). Она мой партнер в жизни и в работе и обычно первая читает то, что я пишу, и первая говорит о моих писательских взлетах и падениях. Спасибо, Люси. Спасибо всем.

Предисловие

Другая сторона программного обеспечения

Эта книга о другой стороне программного обеспечения – той, что смотрит во внешний мир. Эта сторона компьютеров касается людей – технарей, как вы и я, и обычных людей, как вы и я. В собранных здесь заметках исследуются многие разнообразные аспекты человеческого фактора в программировании (peopleware), которые обеспечивают интерфейс между программным обеспечением и его разработчиками, а также между программным обеспечением и его пользователями.

Мои редакторы, как в журналах, в которых этот материал появился впервые, так и в издательстве Prentice Hall, разрешили мне затронуть большой круг вопросов. Тема человеческого фактора в программировании не объятна, и это позволило мне писать почти обо всем, о чем я хотел написать – начиная от организационной культуры и организации проектов, хаоса и дисциплины в кодировании, инструментов и методов программирования и заканчивая пользователями, юзабилити и пользовательскими интерфейсами. Эта широкая область охватывает особый промежуточный мир, в котором сливаются границы между техническими и социальными вопросами. Здесь психология встречается с кибернетикой, а теория и практика смешиваются друг с другом. Все это отражает мой давний личный и профессиональный интерес как к людям, так и к программному обеспечению для компьютеров.

Эта книга является пересмотренным, расширенным и обновленным переизданием книги «Constantine on Peopleware», Prentice Hall, 1995 [26]. Книга слишком радикально пересмотрена, чтобы называться вторым изданием, но в то же время она тесно связана со своей предшественницей. Читатели смогут найти здесь много новых материалов, освещающих данную тематику. К главам из первоначального издания добавлены 25 новых статей, которые впервые публикуются в виде книги. Статьи включают в себя все 52 заметки, которые изначально публиковались в *Computer Language Magazine* и *Software Development* под рубрикой «Peopleware»,

включая «потерянную заметку», появившуюся в самом конце этой серии (см. приложение). Кроме того, для удобства читателя я добавил еще семь близких по содержанию статей из журнала *Object Magazine*. Они особенно важны с точки зрения общего представления о том, что такое дизайн, ориентированный на использование. Описание этого подхода было улучшено и расширено в книге Люси Локвуд (Lucy Lockwood) «Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design», Addison-Wesley, 1999 [30], которая получила всеобщее признание.

В том, чтобы писать о людях, имеет одно большое преимущество – люди изменяются намного медленнее, чем технологии. При перечитывании и редактировании материалов для этой книги я часто поражался тому, как мало изменилось за все эти годы, в течение которых я пишу о человеческой стороне технологии. Проекты все еще превышают установленные для них бюджеты, продукты все еще создаются с бессовестным запаздыванием, ресурсы все еще безнадежно недостаточны для решения задач, а руководители проектов все еще бьются над тем, как лучше всего раскрыть в подчиненных творческий потенциал и направить его в нужное русло. Разработчики, со своей стороны, все еще раздражаются из-за ограничений, которые создают схемы проекта, инструменты моделирования и жесткая дисциплина при создании программного обеспечения. В свою очередь, пользователи все еще вынуждены стараться изо всех сил, чтобы понять программное обеспечение, которое хорошо говорит с компьютерами, но плохо – с обычными человеческими существами.

Тем не менее, если люди меняются мало, то в технологии произошли настолько радикальные изменения, что некоторые примеры и ссылки из ранних статей могут показаться чуть ли не архаичными. Например, статья о переходе от монохромного монитора к полноцветному выглядит возвратом к предыстории разработки программного обеспечения – описанием, предназначенным для новичков. Однако центральная тема этой статьи – правильное и неправильное применение цвета – до сих пор остается актуальной для Интернета. Поэтому с учетом изначальной идеи и цели книги я обновил содержание статей там, где это понадобилось с точки зрения текущей ситуации и корректности.

Данный расширенный сборник разбит на девять тематических разделов. Каждая заметка помещена в соответствующий раздел с тем, чтобы в какой-то мере сохранить последовательность в изложении. Были добавлены новые разделы, посвященные организационной культуре, а также юзабилити объектов программного обеспечения. Читатель быстро поймет, что каждая статья, будучи взаимосвязанной с другими, является, тем не менее, самостоятельным материалом. Думаю, что отчасти именно это вызывало интерес к моим журнальным заметкам и к первому сборни-

ку, поскольку каждая глава может быть прочитана независимо от других в такси или в перерыве между встречами.

Кроме того, новые главы размещены блоками для того, чтобы читатели первой книги смогли их быстро найти. Новые главы: 22–25, 31–32, 40–41, 43–49, 53–61, а также приложение.

Данный материал собран из разных мест. Я надеюсь, что эта книга еще долгое время будет оставаться ценным источником и продолжит раздражать, вдохновлять и просвещать этих удивительных людей, которые работают в индустрии программного обеспечения. В первую очередь я писал и продолжаю писать именно для них – дизайнеров, разработчиков и руководителей, которые изобретают и внедряют программное обеспечение.

Предисловие к первому изданию

Аппаратное и программное обеспечение и человеческий фактор в программировании

Хорошее программное обеспечение берет начало не в применении CASE¹-инструментов, методов визуального программирования, ускоренной разработки программ или объектно-ориентированной технологии. Хорошее программное обеспечение создается людьми. Так же, как и плохое. В 1992 году я начал вести колонку в журнале, исходя из простого принципа: поскольку программное обеспечение создается и применяется людьми, то наилучшее понимание людей – как они работают, каким образом выполняют свою работу, как взаимодействуют между собой – является основой для усовершенствования программного обеспечения и процесса его разработки. Таким образом, основным предметом заметок в этой колонке было не аппаратное (hardware) и не программное (software) обеспечение, а человеческий фактор в программировании (peopleware).

В области, которая изобилует неологизмами, термин «peopleware» – один из немногих, которые действительно стоило выдумать. По-видимому, в печатном издании этот термин первым применил Питер Г. Ньюман (Peter G. Newmann), который, наверное, более всего известен по своим регулярным статьям о риске потребителей и опасностях компьютеров и компьютерных программ для человека. В 1976 году его статья «Peopleware in Systems» была опубликована в книге, получившей свое название от этой статьи. Судя по всему, этот термин придумал Меилир Пейдж-Джонс (Meilir Page-Jones), который использовал его в 1980 году в своей книге «Practical Guide to Structured Systems Design» (Практическое введение в проектирование структурированных систем) – книге, в конце концов сделавшей мою работу по структурному проектированию более понятной для обыч-

¹ CASE (computer-aided software engineering) – автоматизированное проектирование и создание программ. – *Примеч. науч. ред.*

ного программиста. Но постоянное место в лексиконе нашей отрасли этот термин обрел, скорее всего, после того как в 1987 году под таким заголовком вышла небольшая, но великолепная книга Тома ДеМарко (Tom DeMarco) и Тима Листера (Tim Lister) [33].¹ Таким образом, можно сказать, что название для своей колонки, «Peopleware», я позаимствовал из самых лучших источников.

На самом деле «человеческий фактор в программировании» (peopleware) является третьей волной компьютерной революции. Сначала произошел кризис, связанный с аппаратным обеспечением. Одно время мы думали, что наши проблемы возникают из-за аппаратного обеспечения. Нам казалось, что если бы только у нас были более быстрые и мощные компьютеры с большими объемами памяти и более совершенными периферийными устройствами, мы смогли бы создавать более совершенные системы и решать наши задачи. И вот мы получили лучшие компьютеры. Год от года аппаратное обеспечение становилось все быстрее, память – больше, а периферийные устройства – более разнообразными и эргономичными. Однако наши проблемы не исчезли. Мы все еще продолжали создавать системы, трудные для применения. Мы все также опаздывали с завершением своих проектов и превышали запланированный для них бюджет. Поэтому мы решили, что на самом деле проблема состоит в программном обеспечении, после чего линия фронта в компьютерной революции переместилась к рубежу, который многие стали называть «кризисом программного обеспечения». Если бы только у нас были более совершенные инструменты, языки более высокого уровня, более мощные библиотеки компонентов и программы для создания программ, мы смогли бы решать наши задачи и создавать хорошие системы вовремя и в пределах бюджета. Языки третьего поколения стали еще более сложными и породили языки четвертого поколения (4GL). Компиляторы становились все быстрее и эффективнее. Библиотеки компонентов многократного использования расширялись, редакторы стали контекстно-зависимыми, а разнообразные инструменты автоматизированной разработки программного обеспечения появлялись, как грибы после дождя. Вслед за структурной революцией, давшей нам структурное проектирование и анализ, стало развиваться и набирать популярность объектно-ориентированное направление. Однако графики работ по-прежнему не выполнялись, бюджеты превышались, а количество ошибок в программах упрямо не желало становиться меньше.

В конце концов, подобно Пого и его легендарным друзьям из Окифиноки, на своем опыте мы выяснили, в чем тут дело. Как мудро сказал этот ма-

¹ Т. ДеМарко и Т. Листер «Человеческий фактор: эффективные проекты и команды». – Пер. с англ. – СПб: Символ-Плюс, I кв. 2004 г.

ленький опоссум: «Мы встретили врага – это мы сами». И это действительно так. Все сводится к человеческому фактору в программировании. Мы являемся проблемой, и мы же являемся ее решением. Как удобно.

Человеческий фактор в программировании охватывает довольно широкую область. Сюда входит все, что связано с ролью людей в процессе разработки программного обеспечения и приложений. В заметках и в книге затрагиваются разнообразные темы: качество и продуктивность, работа в команде, динамика поведения коллектива, личность и программирование, руководство проектом и организационные вопросы, разработка интерфейса и взаимодействие между человеком и машиной, познавательная деятельность, психология, процессы мышления.

Все эти предметы интересуют и увлекают меня. Я получил ученую степень по менеджменту отчасти потому, что это позволило мне соединить компьютеры и теорию систем с психологией. Моя диссертация была посвящена психологии программирования на компьютере. В течение нескольких лет я знакомил тысячи студентов и десятки коллег с работами психолога Джорджа Миллера (George Miller) и его магическим числом (конечно, я имею в виду 7 ± 2^1). Структура схем была тщательно продумана таким образом, чтобы облегчить формирование визуальных понятий и решение задач. Связывание и сцепление, эти устоявшиеся метрики, лежащие в основе структурного проектирования, в действительности имеют отношение к восприятию людьми компьютерных программ. Сложными или простыми программы делает именно то, что кажется простым или сложным для программистов, которые эти программы пишут, сопровождают и модифицируют.

В известном смысле я могу избегать тему человека не больше, чем тему компьютеров. Казалось, мне удалось уйти от этих вопросов, когда в июле 1976 года я попрощался с компьютерной областью и объявил о своей независимости (Америка как раз отмечала двухсотлетие своего суверенитета). Будучи по образованию семейным врачом, я больше десяти лет потратил на частную практику и работу в медицинской организации, помогая супружеским парам, семьям или просто больным людям. Но силам вселенной было угодно снова вернуть меня в пределы технологии.

Человеческий фактор является перекрестком в этих пределах, пересечением, где сходятся магистрали из моих разных миров. Управление, организационное развитие, личность, моделирование, инструменты, методы, процессы, взаимодействие между компьютером и человеком и т. д. Во

¹ Число, выведенное в 1956 году психологом Джорджем Миллером, относится к возможности человека обрабатывать в кратковременной памяти только от пяти до девяти элементов. – *Примеч. науч. ред.*

всех этих областях в то или иное время я либо писал, либо работал, либо преподавал. Колонка в журнале дала мне повод еще раз окинуть взглядом панораму отрасли. Подобно Чарльзу Куралту (Charles Kuralt), я мог останавливаться для исследования интересной идеи, решать возникавшие задачи и проходить по главным дорогам и магистралям в области разработки программного обеспечения и приложений.

Эта книга является сборником путевых заметок, начатых еще в *Computer Language Magazine* и продолженных в переименованном *Software Development*. Мои заметки выходили под рубрикой с простым названием «Reopleware». В этой книге собраны заметки из этой колонки, а также несколько других, тесно связанных с этой темой и опубликованных в других местах. Эти статьи и эссе были отредактированы в интересах последовательности изложения. Восстановлены некоторые материалы, которые были сокращены при публикации статей в журналах. Таким образом, данная подборка является «редакцией режиссера», в которой статьи размещены в квазилогических разделах, чтобы создать определенную иллюзию организации материала. Однако это не энциклопедия или учебник, и даже не карта огромной территории под названием «человеческий фактор в программировании», а только путевой журнал одного странника.

Путешествие продолжается.

I

Групповая разработка

Введение

Микеланджело, этот гений-индивидуалист, не работал в одиночку. «Сикстинская часовня» не является творением одного художника. Это произведение, созданное группой мастеров под руководством Микеланджело. Очевидно, что его талант проявлялся не только в искусстве как таковом, но также и в искусстве управления командой. Своих художников и подмастерьев он называл по именам и прозвищам. Он поощрял инициативу и взаимовыручку, вселял чувство гордости за мастерство. На собрании, посвященном повышению качества программного обеспечения, он чувствовал бы себя как дома.

Разработка программного обеспечения – это групповой процесс. Независимо от того, идет ли речь об искусстве или о технике, большинство программных продуктов рождается в результате коллективных усилий, когда разработчики трудятся в согласии или разобщенно. Качество программ определяется продуктивностью обсуждений в группе, принимаемыми решениями и отклонениями от них. Создать хорошее программное обеспечение легче, когда разработчики трудятся слаженно, когда группа успешно справляется с конфликтами и использует свои ресурсы рационально и эффективно.

Групповая разработка связана и с развитием самой группы. Процессы, происходящие в группе, и правила, устанавливаемые в ней, могут превратить простое собрание людей в высокопроизводительную команду. К групповой разработке и построению команды многие программисты относятся с долей скептицизма. Им не нравятся лидеры, они не любят петь хором и принимать участие в групповых играх. Им кажется, что все это – лишь видимость командного духа. Эта искренность и открытость не для них. Они были наняты за искусность в программировании, а не за социальные навыки – в большей степени для кодирования, чем для сотрудничества.

Понимание языка, на котором говорят в группах, и процессов, протекающих в них, может быть настолько же важным, как и понимание языка программирования. Вот почему эта книга начинается с исследования того, как работают группы и как можно повысить качество создаваемого ими программного обеспечения.

1

Решения, решения

Путей есть два и больше. Путей всегда есть два и больше. Всю мою профессиональную жизнь этот простой принцип служил практическим ориентиром, который побуждал меня искать различные альтернативы при разработке программного обеспечения и организации трудового процесса. Однако признание существования различных вариантов влечет за собой необходимость выбора. Создание более совершенного программного обеспечения подразумевает выбор между различными вариантами. Более того, это творческий подход, вбирающий в себя самое лучшее из нескольких методов и потому превосходящий каждый из них. Хорошо организованные команды, в которых принятие решений и преодоление трудностей осуществляется на основе консенсуса, в наибольшей степени способны к принятию качественных решений и выработке такого творческого подхода. Тем не менее таким командам следует знать, как избегать некоторых ловушек, в которые обычно попадают группы. Для этого полезно исследовать секреты командной работы, основанной на согласии.

Я всегда считал, что способность принимать решения является одним из самых важных жизненных навыков. Этому можно научиться только опытным путем. Семьи и компании, достигшие успеха, стараются обеспечить достаточные возможности для такого рода практики. К середине карьеры типичный профессиональный программист уже решил бесчисленное множество задач и, вероятно, принял не одну тысячу решений. Естественно, от профессионалов мы вправе ожидать, что как раз в этом они обладают хорошими способностями. Однако большинство таких решений программист принимает самостоятельно, тогда как принятие решений и преодоление трудностей в группе – дело совсем другое.

Опасность посредственности

В давние времена, когда я только начал изучать менеджмент и теорию поведения групп в школе управления Sloan при Массачусетском технологическом институте, исследователи уделяли много внимания возможным слабым местам, типичным при преодолении трудностей и принятии решений в группах, особенно так называемому «поднятию порога риска», а также обратной тенденции скатывания к посредственному среднему. В ту эпоху консерватизма даже демократически настроенные менеджеры больше беспокоились о том, чтобы не допустить поднятия порога риска, чем о прогрессирующей посредственности. Перевороты 70-х были еще впереди, а «групповое мышление» соответствовало духу времени.

Согласно исследованиям, коллективные решения зачастую сводились к более рискованным вариантам по сравнению с теми, которые сотрудники группы могли бы принять самостоятельно. Если применить эту модель к программированию, то можно было бы ожидать, что группы будут создавать программы, использующие экзотические структуры данных, или необычные алгоритмы, или неявные возможности языка. Однако другие исследования групповой динамики показали, что при выполнении задач и принятии решений в группах имеет место эффект нивелирования, который сводит результаты к своего рода наименьшему общему знаменателю индивидуальных способностей и вкладов. Казалось, что один человек принимает более верные решения..

Очевидно, что такие эффекты зависят главным образом от организации и методов управления в группе. В России, где я работал с консультантами и руководителями новых предприятий, «посредственность» доминировала благодаря старой советской системе. Для многих руководителей, учившихся еще при старом режиме, командная работа означала переход на уровень общей некомпетентности. В советской системе управления работа в команде часто позволяла избежать ответственности. Иногда коллективы умышленно снижали свою производительность. Отстаивание своей точки зрения или выдвижение новой, спорной идеи грозило не только недовольством коллег, но и возложением ответственности и ожиданием подобных инициатив в будущем. Зачем беспокоиться, если производительность никак не вознаграждается, а потерять работу в типичном советском коллективе трудно даже при большом желании?

Социальный и организационный климат, в котором работает группа, – это как раз то, что в действительности определяет возможность раскрытия потенциала. Для достижения лучших результатов корпоративная культура и само руководство группы должны активно поощрять и поддерживать новаторство и сотрудничество. В известном смысле, команды советского периода работали действительно хорошо, удовлетворяя *реальные* ожида-

ния начальников и политических руководителей, хотя их усилия больше сводились к прикрытию тылов, нежели к достижению результатов. Советские менеджеры говорили мне, что они научились «никогда не оказываться крайними».

Слегка управляя

Если процесс разработки и принятия решений основан на консенсусе, то роль группового лидера становится решающей – не только в создании общего климата сотрудничества, но также и с точки зрения нюансов осуществления руководства. Совместная разработка и принятие решений дает наилучшие результаты, если решение рождается из талантов всех участников команды и отражает их опыт, творческие способности и умение мыслить критически. Речь идет не об отражении «в среднем», а об истинном синтезе, в котором соединяются лучшие качества каждого сотрудника. Какими бы талантливыми и выдающимися ни были лидеры группы, но если они начинают проталкивать собственные решения, качество работы всей команды снижается.

Процесс руководства группой может быть едва заметным. Даже простое выражение своего мнения в неподходящий момент может отклонить группу от правильного пути и привести к худшему результату. Исследования показали, что если руководитель откладывает изложение собственных идей до тех пор, пока все участники группы или их большинство не представят свои, группа приходит к более совершенным решениям. Это означает, что лидер, который всего лишь высказывается слишком рано, возможно, снижает качество работы группы. Самоуверенные руководители, убежденные в том, что они всегда правы и знают все на свете, могут создать наибольшие трудности в работе.

Большинство лидеров проектов и руководителей среднего звена в сфере разработки программного обеспечения являются довольно сильными специалистами. Почти каждый из них поднялся из отделов программирования, системного анализа и проектирования. Они достигли своего положения потому, что разбираются в вопросах разработки программ. Многим из них трудно оставить клавиатуру и позволить кому-либо другому выполнять эту работу и решать технические вопросы.

Теперь мы знаем, что одним из самых важных факторов, способствующих наилучшему решению задач на основе консенсуса, является нейтральность руководства. Позиция ведущего обсуждения довольно сильна, поэтому тот, кто ведет или помогает вести собрание и обсуждение, должен быть нейтральным. Лишь в этом случае проявится наилучшее из того, что может предложить группа.

Такой лидер является другом для каждого и, в то же время, не является чьим-то адвокатом. Такой лидер стремится получить вклад от всех участников группы, никого из них не выделяя. Такой лидер помогает группе прийти к техническому консенсусу, но не оказывает влияния на исход обсуждения и не навязывает свой вариант решения.

Звучит странно, но это означает, что менеджеры проектов и официальные командные лидеры, вероятно, меньше всего подходят для ведения обсуждений или собраний, посвященных преодолению технических трудностей и принятию технических решений. Для них слишком многое стоит на кону. В известном смысле они, наверное, и знают слишком много. Чем сильнее они как лидеры, тем вероятнее то, что они будут сдерживать свободное исследование различных альтернатив и выработку единого технического решения, ведущего к наилучшему результату.

Некоторые руководители избирают подход полного невмешательства и стараются совсем не вдаваться в технические вопросы. Однако это не является полезным ни для команд, которые остаются без опыта и мастерства своих руководителей, ни для самих руководителей, которые упускают много интересного. Лучшие из них передадут ведение собраний или обсуждений нейтральному ассистенту, а сами, оставаясь в заднем ряду, станут учиться тому, как участвовать в работе, не доминируя. Некоторые так никогда и не смогут научиться этому, но многие из тех, с кем мне довелось работать, действительно получают удовольствие от того, что они снова могут быть «в одной связке» и на равных со всеми обсуждать технические вопросы.

Жизнь продолжается. Даже после того, как вас повысили в должности.

Из журнала *Computer Language Magazine*, том 9, № 3, март 1992 г.

По договору между издательством «Символ-Плюс» и Интернет-магазином «Books.Ru-Книги России» единственный легальный способ получения данного файла с книгой «FreeBSD. Подробное руководство» (ISBN 5-93286-066-9) – 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» (www.symbol.ru), где именно Вы получили данный файл.