

# Предисловие

Как принять решение о том, что собой должна представлять программная система? А как добиться того, чтобы принятое решение отражало интересы самых разных людей, непосредственно или косвенно связанных с проектом? Рассмотрению этой сложнейшей проблемы и посвящена данная книга. Основные трудности обусловлены необходимостью согласования множества различных целей и потребностей всех участников проекта. Менеджер проекта стремится контролировать ход выполнения работ. Программисты хотят реализовать систему. Менеджер по продукту требует гибкости программного обеспечения. Тестировщикам нужны модули, обладающие измеримыми характеристиками. Пользователи заинтересованы в получении работоспособного и эффективного приложения. Возникновение конфликта между столь разными ожиданиями от проекта, поиск единого коллективного решения, одобренного всеми сторонами, и сохранение достигнутого баланса интересов на протяжении месяцев или даже лет — все вместе это представляет серьезную проблему.

Подход, предлагаемый Майком Коном в данной книге, внешне мало чем отличается от уже известных решений — спецификаций требований, вариантов использования и сценариев. Казалось бы, что тут сложного? Вы просто записываете, что надо сделать, а потом делаете это. Однако при любой попытке передачи решений между участниками проекта сразу же выясняется, что проблема не так проста, как кажется. Все зависит от того, что именно записывать и когда именно это делать.

Рассматриваемая в книге технология процесса проектирования начинается с составления пользовательских историй, содержащих следующую информацию: описание пользовательских целей, достижение которых должна обеспечить реализуемая система, и примерная оценка связанных с этим трудозатрат. Такая история уместается буквально в нескольких коротких предложениях и предоставляет информацию, которая не обеспечивается при других подходах. Действуя по принципу “последний момент решающий”, команда откладывает детализацию функциональных возможностей до того времени, когда это становится действительно необходимым для их реализации.

Такой простой перенос планирования части работ на более поздние этапы разработки приводит к двум важным последствиям. Во-первых, команде не составляет труда начать реализацию наиболее “лакомых” функциональных возможностей на ранних этапах, когда о других возможностях достаточно иметь лишь самые общие представления. Автоматизированные тесты, документирующие детали отдельных функциональных возможностей, гарантируют сохранение работоспособности продукта и его соответствие ожиданиям заказчика на протяжении всего цикла разработки, начиная с самых ранних ее стадий. Во-вторых, раннее определение стоимости разработки каждой функциональной возможности заставляет устанавливать приоритеты историй уже с самого начала, что исключает лихорадочный пересмотр проекта на завершающем этапе, когда команда лезет из кожи вон, лишь бы уложиться в срок.

Благодаря большому опыту работы Майка с пользовательскими историями книга насыщена множеством практических рекомендаций, которые позволят вашей команде разработчиков получить максимальную отдачу от применения пользовательских историй в своих проектах.

Кент Бек  
Three Rivers Institute

# Благодарности

Работая над книгой я учитывал замечания множества рецензентов. В частности, хочу поблагодарить Марко Абиса, Дейва Астелза, Стива Баннермана, Стива Берчука, Лин Бейн, Дэна Брауна, Лору Кон, Рона Крокера, Уорда Каннингема, Рейчел Дейвис, Роберта Элсуорта, Дорис Форд, Джона Гилмана, Свена Гортса, Дебору Хартманн, Криса Лесли, Цзинь Кьун Лина, Кита Рэя, Мишель Слигер, Джеффа Тейтельмана, Анко Теймана, Тронда Вингорда, Джейсона Ипа и многих других анонимных рецензентов.

Выражаю искреннюю признательность официальным рецензентам: Рону Джеффрису, Тому Поппендику и Биллу Уэйку. Рон помогал мне оставаться честным и гибким. Том открыл для меня многие идеи, которые я раньше не замечал. Билл не давал мне сбиться с курса и поделился со мной своим акронимом INVEST. Советы этих прекрасных людей, сотрудничеством с которыми я горжусь, помогли значительно улучшить мою книгу.

Я также благодарен Лайзе Криспин, автору книги *Testing Extreme Programming*, которая вдохновила меня на написание книги своими рассказами о приятном опыте сотрудничества с издательством Addison-Wesley. Без ее поддержки я никогда не взялся бы за этот проект.

Большую часть того, что знаю, я обсуждал с Тодом Голдингом на протяжении последних девяти лет. Чаще всего мы сходились с Томом во мнении, что оба в достаточной степени владеем предметом, но я всегда черпал в наших беседах что-то новое для себя. Я благодарен Тому за все, чему он научил меня за многие годы. Содержание моих бесед с ним во многом обогатило эту книгу.

Я благодарю Алекса Виджио и всех коллег из XP Denver, с которыми я имел возможность поделиться многими идеями, вошедшими в книгу. Говорю спасибо Марку Мошолдеру и Дж. Б. Рейнсбергеру, которые рассказали мне о том, как они используют программное обеспечение вместо бумажных карточек. Я благодарен Кенни Рубину, написавшему в соавторстве с Адель Гольдберг книгу *Succeeding With Objects*. Его блистательное мастерство, которым наполнена эта книга, побудило меня вернуться к писательской деятельности после нескольких лет перерыва.

Сердечная благодарность Марку и Дэну Гутрич, основателям компании Fast401k, которые всей душой восприняли идеи пользовательских историй и Scrum. Наряду с ними хочу поблагодарить также всех моих коллег в Fast401k, где мы успели далеко продвинуться на пути к нашей заветной цели — стать одной из лучших команд в Колорадо.

Никакими словами не передать, насколько я благодарен своей семье за то, что она дала мне возможность работать над книгой в течение длительного времени, ни на что не отвлекаясь. Спасибо вам, мои очаровательные дочери и принцессы Саванна и Дилейни. Особая благодарность моей прекрасной и очаровательной жене Лоре за то многое, что она сделала для меня, тогда как я для нее сделал так мало.

Я в большом долгу перед коллективом Addison-Wesley. Пол Петралия сделал весь процесс подготовки книги к печати приятным от начала до конца. Микеле Винченти заставлял меня не стоять на месте. Лайза Ярковски оказала мне неоценимую помощь при работе с программой FrameMaker. Гейл Кокер придала моим иллюстрациям приличный вид, а Ник Радхубер свел напоследок все воедино.

И наконец, не могу не поблагодарить Кента Бека за его глубокомысленные советы, за проведенное в обществе с ним время и за включение моей книги в его серию *Signature Series*.

# Введение

Примерно с середины 90-х годов прошлого столетия я начал испытывать чувство вины. Я работал на компанию, которая ежегодно покупала другие компании, и каждый раз меня назначали руководить вновь приобретенной группой разработки программного обеспечения. Каждая группа приносила с собой великолепно оформленные, объемные, тщательно документированные требования. Я постоянно чувствовал себя виноватым, потому что в моих собственных группах никогда не готовились настолько замечательные спецификации требований. И тем не менее в *производстве программного обеспечения* мои группы всегда были более успешными, чем те, которые приобретались.

Я понимал: то, что мы делаем, работает. Но я никак не мог избавиться от мысли, что если бы мы научились выкраивать время для подготовки развернутых, обширных спецификаций, то смогли бы стать еще более успешными. В конце концов, именно об этом писали во всех книгах и статьях, которые я тогда читал. Если все успешные команды разработчиков не жалели времени для написания великолепно подготовленных документальных требований, то, казалось бы, точно так же должны поступать и мы. Но на это нам никогда не хватало времени. Наши проекты всегда были слишком важными, а их разработка всегда была срочной, поэтому любые задержки перед началом проекта были просто непозволительны.

Из-за вечной нехватки времени для написания прилично документированных требований мы остановились на способе их установления, который предполагал простые беседы с пользователями. Вместо того чтобы все записывать, каждый раз просматривать сделанные записи и вести переговоры, постоянно прислушиваясь к неумолимому тиканью часов, мы разговаривали. Мы рисовали эскизы экранов на бумаге, иногда создавали прототипы, часто писали немного кода, а затем показывали предполагаемым пользователям, что у нас получилось. Не реже одного раза в месяц мы собирали представительную группу пользователей и показывали им все, что к тому времени было получено в виде работоспособного кода. Сохраняя постоянный контакт с пользователями и демонстрируя им достигнутые результаты небольшими порциями, мы нашли способ быть успешными, обходясь без формального документирования требований.

И все же я испытывал вину за то, что мы поступаем не так, как должны были бы.

В 1999 году вышла небольшая книга Кента Бека под названием *Extreme Programming Explained: Embrace Change*, которая перевернула мои взгляды. Хватило всего одной ночи, чтобы мое чувство вины улетучилось. Наконец-то появился тот, кто сказал, что беседы разработчиков с заказчиками, вместо писанины, переговоров и повторной писанины, — это совершенно нормальная вещь. Кент многое разъяснил для меня и открыл передо мной множество путей. Но что самое главное, он дал разумное обоснование того, чему я научился на собственном опыте.

Основательная работа по сбору и документированию требований еще до начала каких бы то ни было работ во многих отношениях способна погубить проект на корню. Одна из наиболее распространенных опасностей состоит в том, что документирование требований становится самоцелью. Документальное оформление требований необходимо применять лишь в тех случаях, когда оно способствует достижению реальной цели поставки программного обеспечения.

Вторая опасность, которая может подстергать каждого при работе с документированными требованиями, связана с ограниченными возможностями передачи мыслей средствами письма. Помню, как много лет назад мне рассказали одну историю о ребенке, которого собирались купать в ванне. Отец наполнил ванну и начал помогать ребенку войти в воду. Девочка, двух-трех лет от роду, едва окунув ногу в воду, быстро вытянула ее обратно и попросила отца “сделать воду теплее”. Отец попробовал рукой воду и очень удивился, поскольку она оказалась не слишком холодной, как он подумал, а даже теплее, чем обычно.

Задумавшись над просьбой дочери, отец пришел к выводу, что причина недопонимания крылась в том, что в одни и те же слова они вкладывали разный смысл. Просьба дочери “сделать воду теплее” для отца означала “увеличить температуру воды”. Для девочки же “сделать теплее” означало “сделать температуру воды ближе к привычному ощущению тепла”.

Слова, особенно когда их видишь на письме, представляют собой слишком тонкий носитель, чтобы его можно было использовать для передачи столь сложных требований к программному обеспечению. Учитывая риск неправильного истолкования написанного, мы должны заменить письменные формулировки частыми обсуждениями, в которых участвуют разработчики, заказчики и пользователи. Пользовательские истории не только позволяют записывать лишь то, о чем мы не должны забыть и что сможем оценить и использовать при составлении планов, но и стимулируют устные обсуждения. К тому моменту, когда вы закончите чтение части I, вы уже будете подготовлены к тому, чтобы полностью отказаться от строгой фиксации всех без исключения требований вплоть до мельчайших деталей. После прочтения всей книги вы будете знать все необходимое для того, чтобы реализовать процесс, управляемый историями, в своей среде.

Книга разбита на пять частей: в частях I–IV содержится основной материал, а в части V — два приложения.

- **Часть I. “Приступаем к работе”.** Содержит все необходимое для того, чтобы вы могли приступить к написанию историй *уже сегодня*. Одна из целей написания пользовательских историй — заставить людей разговаривать друг с другом, а не обмениваться письменной документацией. Цель части I — стимулировать вас к тому, чтобы вы приступили к таким обсуждениям как можно раньше. Глава 1 содержит обзор, из которого вы узнаете, что такое пользовательские истории и для чего они предназначены. В следующих главах подробно рассказывается о том, как писать пользовательские истории, как осуществлять их сбор посредством моделирования ролей, как писать истории в отсутствие контакта с конечными пользователями и как тестировать такие истории. Завершает эту часть глава, содержащая рекомендации по улучшению пользовательских историй.
- **Часть II. “Оценка и планирование”.** После того как у вас наберется достаточное количество пользовательских историй, вам в большинстве случаев потребуется хотя бы приблизительно оценить, сколько времени займет разработка. В этой части объясняется, как оценить трудоемкость историй в баллах, как планировать выпуски на периоды времени от трех до шести месяцев, как планировать предстоящую итерацию с учетом большего количества деталей и, наконец, как измерить количественные характеристики хода выполнения работ и оценить, продвигается ли разработка проекта с должной скоростью.
- **Часть III. “Часто обсуждаемые вопросы”.** Эта часть начинается с описания того, чем пользовательские истории отличаются от других возможных видов требова-

ний, таких как варианты использования, спецификации требований и сценарии проектирования взаимодействия. В следующих главах будет рассказано об уникальных преимуществах пользовательских историй, а также о том, как определить, что что-то пошло не так, и как видоизменить процесс Scrum для применения пользовательских историй. В заключительной главе данной части рассматривается ряд второстепенных тем, например, вопрос о том, что лучше — записывать истории на бумажных карточках или сохранять их в электронном виде с помощью каких-либо программ, а также вопрос о том, как обрабатывать нефункциональные требования.

- **Часть IV. “Пример”.** В этой части представлен расширенный пример, иллюстрирующий практическое применение всего изложенного материала. Если утверждается, что пользовательские истории помогают разработчикам лучше понять потребности пользователей, было вполне логично завершить данную книгу рассмотрением обширной истории, позволяющей проиллюстрировать все аспекты пользовательских историй, собранные в одном примере.
- **Часть V. “Приложения”.** Пользовательские истории берут свое начало в экстремальном программировании. Чтобы читать эту книгу, вы не обязаны быть знатоком экстремального программирования. Тем не менее краткое введение в эту дисциплину дано в приложении А. В приложении Б содержатся ответы на контрольные вопросы, которыми завершаются главы.

---

## **Ждем ваших отзывов**

Вы, читатель этой книги, и есть главный ее критик. Мы ценим ваше мнение и хотим знать, что было сделано нами правильно, что можно было сделать лучше и что еще вы хотели бы увидеть изданным нами. Нам интересно услышать и любые другие замечания, которые вам хотелось бы высказать в наш адрес.

Мы ждем ваших комментариев и надеемся на них. Вы можете прислать нам бумажное или электронное письмо либо просто посетить наш Web-сервер и оставить свои замечания там. Одним словом, любым удобным для вас способом дайте нам знать, нравится ли вам эта книга, а также выскажите свое мнение о том, как сделать наши книги более интересными для вас.

Посылая письмо или сообщение, не забудьте указать название книги и ее авторов, а также ваш обратный адрес. Мы внимательно ознакомимся с вашим мнением и обязательно учтем его при отборе и подготовке к изданию последующих книг. Наши координаты:

E-mail: [info@williamspublishing.com](mailto:info@williamspublishing.com)

WWW: <http://www.williamspublishing.com>

Адреса для писем из:

России: 127055, г. Москва, ул. Лесная, д. 43, стр. 1

Украины: 03150, Киев, а/я 152