

Agile

Оценка
и планирование
проектов

Майк Кон

Если вы руководитель проекта, исполнитель проекта, разработчик или директор, то найдете здесь инструменты, необходимые для оценки, планирования и управления agile-проектами практически любого масштаба.

альпина
ПАБЛИШЕР
| бизнес

Майк Кон

**Agile: оценка и
планирование проектов**

«Альпина Диджитал»

2006

Кон М.

Agile: оценка и планирование проектов / М. Кон — «Альпина Диджитал», 2006

ISBN 978-5-9614-5208-2

Оценка и планирование критически важны для успеха любого проекта. Однако процесс планирования сложен, и наши планы часто оказываются далекими от реальности. На помощь приходит agile-подход. Благодаря agile вы научитесь создавать реалистичные планы, которые сможете корректировать по ходу работы, при этом выполняя проекты в срок и в рамках бюджета. Майк Кон, гуру в области agile, дает инструменты, необходимые для оценки, планирования и управления agile-проектами любого масштаба. В книге нет теоретических рассуждений, она полна конкретных примеров, методов, графиков, рецептов, а главное – аргументированных рекомендаций.

ISBN 978-5-9614-5208-2

© Кон М., 2006

© Альпина Диджитал, 2006

Содержание

Об авторе	8
Предисловие	9
Предисловие	10
Предисловие	13
Благодарности	15
Введение	17
Часть I	19
Глава 1	20
Зачем это нужно	21
Что делает план хорошим	24
Что делает планирование гибким	25
Резюме	25
Вопросы для обсуждения	26
Глава 2	27
Планирование ориентировано на деятельность, а не на функцию	27
Многозадачность приводит к дальнейшим задержкам	30
Функции не разрабатываются в соответствии с их приоритетом	32
Конец ознакомительного фрагмента.	33

Майк Кон

Agile: оценка и планирование проектов

Майк Кон

AGILE

ОЦЕНКА И ПЛАНИРОВАНИЕ ПРОЕКТОВ

Перевод с английского



альпина
ПАБЛИШЕР

Москва
2018

Переводчик *В. Ионов*
Главный редактор *С. Турко*
Руководитель проекта *А. Василенко*
Корректоры *Е. Аксёнова, О. Улантимова*
Дизайн обложки *Ю. Буга*
Компьютерная верстка *А. Абрамов*

© Authorized translation from the English language edition, published by Pearson Education, Inc.; publishing as Prentice Hall. Copyright © 2006 by Pearson Education, Inc. All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

© Издание на русском языке, перевод, оформление. ООО «Альпина Пабlishер», 2018 год

Все права защищены. Произведение предназначено исключительно для частного использования. Никакая часть электронного экземпляра данной книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, включая размещение в сети Интернет и в корпоративных сетях, для публичного или коллективного использования без письменного разрешения владельца авторских прав. За нарушение авторских прав законодательством предусмотрена выплата компенсации правообладателя в размере до 5 млн. рублей (ст. 49 ЗОАП), а также уголовная ответственность в виде лишения свободы на срок до 6 лет (ст. 146 УК РФ).

* * *

Посвящается Лоре, без всяких сомнений

Об авторе

Майк Кон – основатель Mountain Goat Software, фирмы, занимающейся консалтингом в сфере управления процессами и проектами. Майк специализируется на помощи компаниям в применении agile-подхода с целью повышения эффективности. Он также является автором книги «Пользовательские истории: Гибкая разработка программного обеспечения» и книг по языкам программирования Java и C++. За спиной у Майка более чем 20-летний опыт работы руководителем в организациях разного размера, от стартапа до компании из списка Fortune 40. Его статьи можно найти в таких изданиях, как *Better Software*, *IEEE Computer*, *Cutter IT Journal*, *Software Test and Quality Engineering*, *Agile Times* и *C/C++ Users Journal*. Он часто выступает на отраслевых конференциях, является соучредителем организации Agile Alliance и входит в ее совет директоров. Майк – сертифицированный Scrum-мастер и тренер, член Компьютерного общества IEEE и Ассоциации компьютерной техники.

Для получения дополнительной информации обращайтесь на сайт www.mountaingoatsoftware.com или отправьте запрос по адресу mike@mountaingoatsoftware.com.

Предисловие

Куда бы я ни попадал в agile-мире, везде слышал одни и те же вопросы:

- Как подходить к планированию, когда работает большая команда?
- Каким должен быть размер итерации?
- Как представлять руководству данные о прогрессе в разработке?
- Как приоритизировать истории?
- Как получить представление о проекте в целом?

Настоящая книга дает ясные ответы не только на эти, но и на многие другие вопросы. Если вы руководитель проекта, исполнитель проекта, разработчик или директор, то найдете здесь инструменты, необходимые для оценки, планирования и управления agile-проектами практически любого масштаба.

Я знаю Майка Кона уже пять лет. Мы познакомились вскоре после подписания Agile-манифеста. Майк привнес в организацию Agile Alliance заряд энтузиазма и энергии. Любой проект, за который бы он ни взялся, выполнялся полностью и на хорошем уровне. Его вклад был видимым и ценным. Майк очень быстро стал незаменимым членом этой молодой организации.

К созданию данной книги он подошел так же профессионально, тщательно и энергично. Этого нельзя не заметить, это сквозит в каждой строчке.

Это видно по тому, что рекомендации в книге носят исключительно практический характер. Здесь нет теоретических абстракций. Автор не заставляет читателя витать в облаках, созерцая проблемы с высоты 10 000 метров. Майк приводит конкретные примеры, методы, инструменты, графики, рецепты, а главное – аргументированные рекомендации. Эта книга – практическое руководство по оценке и планированию.

Она одобрена описанием случаев из практики Майка, связанных с применением рассматриваемых методов и инструментов. Он рассказывает, когда они давали результат, а когда нет, говорит о том, где можно оступиться, а где все будет нормально. Вы найдете в книге не обещания, панацеи и гарантии, а в буквальном смысле кладесь добытого с большим трудом опыта.

Гибкий подход к оценке и планированию затрагивают многие работы, однако таких, где этот подход является главной темой, единицы. Равных же этой книге по глубине и полезности вообще нет. В ней данная тема рассматривается настолько полно, настолько практично, что, на мой взгляд, ее можно считать фундаментальным трудом.

Возможно, я преувеличиваю, но она меня зацепила. Зацепила тем, что в ней даны квалифицированные ответы на многие давно ожидающие решения вопросы. Она стала инструментом, который я могу предложить клиентам, когда они ставят сложные проблемы. Меня радует, что эта книга вышла в свет и вот-вот попадет к читателю. Мне очень приятно, что она вошла в курируемую мной серию. Думаю, это бестселлер.

Роберт Мартин, редактор серии

Предисловие

Читая книгу или рукопись в первый раз, я всегда задаю себе вопрос: «Что нового автор привносит в эту область?» В случае Майка у меня два ответа: его книга расширяет наше представление о том, «как» подходить к оценке и планированию, а также «почему» важны определенные методы.

Agile-подход к планированию обманчив. Может показаться, что он довольно легкий: создайте карточки историй, присвойте им приоритет, распределите их по итерациям, а затем добавьте детали – и получите план следующей итерации. Основы планирования можно объяснить команде за пару часов, и ей потребуется всего несколько часов, чтобы составить сносный план (для небольшого проекта). Книга Майка здорово помогает перейти от сносных планов к составлению очень хороших планов. В этом месте я предельно аккуратно подбираю слова. Я не говорю «превосходных планов», поскольку, как подчеркивает Майк в своей книге, выигрыш от превосходного плана по сравнению с (довольно) хорошим планом чаще всего оказывается не настолько значительным, чтобы тратить на него дополнительные силы.

Поначалу мои мысли о книге Майка крутились вокруг концепции agile-подхода к планированию. Меня всегда удивляло, а порой расстраивало массовое непонимание сути agile-подхода к планированию. То и дело слышишь саркастические замечания о том, что «команды agile-проектов не занимаются планированием» или «agile-команды не придерживаются ни определенных сроков, ни определенных требований». Даже Барри Боэм и Ричард Тернер не совсем правы в своей книге «Баланс гибкости и дисциплины: Руководство для сбивших с толку» (Balancing Agile and Discipline: A Guide for the Perplexed. Addison-Wesley, 2004), когда сравнивают традиционные методы планирования с agile-методами. Фактически Боэм и Тернер правильно понимают идею, но используют неверную терминологию. У них под термином «методы планирования» понимается «тщательно взвешенное сочетание прогноза и адаптивного приближения к прогнозу», а по отношению к термину «agile-методы» взвешивание является антонимом. Таким образом, противопоставление «традиционных методов планирования» и «agile-методов» несет совершенно неправильный посыл, что agile-команды не занимаются планированием. Нет ничего более далекого от реальной практики. Книга Майка дает правильную установку – планирование является неотъемлемой частью любого agile-проекта. В ней очень много говорится о том, почему планирование так важно, и о том, как сделать его эффективным.

Во-первых, agile-команды планируют очень многое, но этот процесс более равномерно распределен по всему проекту. Во-вторых, agile-команды прямо учитывают тот критический фактор, который упускают из виду многие не использующие agile-подход команды, – неопределенность. Важно ли планирование? Несомненно, важно. Важна ли корректировка плана по мере накопления знаний и уменьшения неопределенности? Исключительно важна. Мне известно множество случаев, когда организации, которые на начальном этапе принимали нереальные обязательства, а потом не могли их выполнить, оказывались подходящими для заказчика, в то время как на тех, которые старались быть реалистичными (и понимали неопределенность), навешивали ярлык «неспособные соблюдать программу» или «некомандные игроки». Похоже, срыв поставки продукта считается приемлемым, а отказ принять обязательства (даже когда они очевидно нелепы) – нет. Agile-подход в мастерском представлении Майка сфокусирован на поставке ценного для пользователя продукта, а не на составлении ничем не оправданных и невыполнимых планов и принятии обязательств. Agile-разработчики, по существу, говорят: «Мы предоставим вам план на основе того, что нам известно в настоящий момент; мы будем адаптировать этот план так, чтобы реализовать вашу наиболее важную цель; мы будем адаптировать проект и наши планы по мере продвижения вперед и получения новой инфор-

мации; мы ожидаем, что вы понимаете, о чем просите нас. Иными словами, гибкость, допускающая адаптацию к меняющимся условиям, и жесткое соблюдение первоначальных планов являются взаимоисключающими целями. В настоящей книге разбираются все эти утверждения.

Возвращаясь к критически важному вопросу управления неопределенностью, Майк превосходно показывает, как agile-подход к процессу разработки снижает одновременно и неопределенность целей (что мы реально хотим создать), и неопределенность средств их достижения (как мы будем создавать это). Многие сторонники традиционного планирования не понимают ключевого момента: планирование не устраняет неопределенность. Планы строятся на основе того, что мы знаем в данный момент. Неопределенность – это способ представления того, что нам неизвестно относительно целей и средств их реализации. Для большинства неопределенностей (отсутствия знания) единственным путем их снижения и приобретения знания является действие – выполнение каких-либо работ, создание чего-либо, моделирование чего-либо – и получение обратной связи. Подход многих руководителей проектов можно представить как «планирование, планирование, планирование – выполнение». Agile-подход – это «планирование – выполнение – адаптация», «планирование – выполнение – адаптация». Чем выше неопределенности проекта, тем важнее применение agile-подхода для успеха.

Я бы хотел проиллюстрировать «как» и «почему» из книги Майка на примере глав 4 и 5, где детально показано, как оценивать пользовательские истории в пунктах или идеальных днях, а также приведены все за и против для каждого из этих подходов. Я практиковал оба подхода при работе с клиентами, но слова Майка помогли кристаллизироваться моим представлениям об оценке историй в пунктах и позволили понять, что пункты являются частью эволюции – эволюции в направлении простоты. Организации, занимающиеся разработкой программного обеспечения, давно ищут ответ на вопрос «насколько велик данный элемент программного обеспечения?». Строитель способен дать довольно обоснованную оценку, имея данные о площади здания. Оценки разных строителей могут варьировать, но размер фиксирован (хотя отделочные работы, требования к материалам и т. п. также влияют на оценку) и остается постоянным. Разработчики программного обеспечения давно хотят иметь подобный показатель.

В сфере разработки программного обеспечения для измерения размера продукта поначалу использовали количество строк программы (этот показатель до сих пор не вышел из употребления). В текущем планировании, однако, количество строк программы находит ограниченное применение по целому ряду причин, включая трудозатраты на их подсчет. Затем на сцену вышли функциональные точки (и несколько аналогичных идей). Функциональные точки устраняли некоторые проблемы показателя количества строк, но по-прежнему требовали значительных трудозатрат для подсчета (нужно было оценивать входные данные, выходные данные, файлы и т. п.). Впрочем, на пути широкого использования функциональных точек встали не трудозатраты, а их сложность. По моему мнению, именно увеличение сложности подсчета – беглый просмотр веб-сайта International Function Point User Group (IFPUG) дает хорошее представление об уровне этой сложности – привело к сокращению использования этого показателя.

Так или иначе, потребность в оценке «размера» программного проекта никуда не исчезла. Проблема с обоими историческими показателями имеет две стороны – их сложно определять и они основаны на каскадном подходе к разработке. Нам же требуется показатель размера, который просто определить и можно применять без необходимости проходить все требования и фазы проектирования.

Два критически важных отличия пунктов от количества строк программы и функциональных точек заключаются в том, что они проще в расчете и могут определяться на значительно более раннем этапе. Почему они проще? Да потому, что характеризуют относительный размер, а не абсолютный. Почему их можно определять на более раннем этапе? Да потому, что они относятся в большей мере к относительному размеру, чем к абсолютному.

Как подчеркивает Майк, оценка с использованием пунктов – это обсуждение историй за круглым столом (обмен знаниями) и предположительная оценка относительного размера истории. Оценка относительного размера, в отличие от оценки абсолютного, осуществляется на удивление быстро. К тому же после нескольких итераций оценки размера и поставки продукта точность предположительных оценок, даваемых командой, значительно возрастает. Описание Майком «как» и «почему» при сравнении оценки в пунктах и идеальных днях позволяет глубоко понять эту критически важную тему.

Еще одним примером тщательности изложения материалов Майком служат главы 9–11, посвященные приоритизации историй. Майк не ограничивается советом братья в первую очередь за истории с наивысшей стоимостью, а раскрывает ключевые аспекты стоимости: финансовые выгоды, затраты, инновации/знание и риск. Он дает четкие разъяснения по каждому из этих аспектов (включая общие представления о чистой приведенной стоимости, внутренней ставке доходности и других инструментах финансового анализа), а затем приводит ряд схем (с разной степенью упрощения) принятия решений по весам на основе рассмотренных аспектов стоимости.

Нередко новички в области agile-разработки полагают, что если применить определенную методологию 12, 19 или 8 раз, то это и будет Agile, Extreme, Cristal Clear или еще что-нибудь подобное. Однако на самом деле вы применяете методологию Agile, Extreme и т. п. только тогда, когда знаете достаточно, чтобы адаптировать ее к своей конкретной ситуации. Суть agile-разработки – непрерывное обучение и адаптация. Что Майк отлично делает в этой книге, так это знакомит нас с идеями и практикой, которые помогают вывести наши agile-оценку и agile-планирование на следующий уровень развития. Майк детально разъясняет, «как» подойти к делу, например провести оценку в пунктах и идеальных днях, и «почему» нужно подходить именно так, представляя плюсы и минусы показателей «пункты» и «идеальные дни». Хотя он обычно дает рекомендации (сам он предпочитает пункты), в книге достаточно информации, чтобы мы уверенно могли адаптировать методику к своей ситуации.

Именно в этом и заключается вклад Майка в данную область – сначала он помогает нам получать новые знания и опыт через оценку и планирование («как»), а затем принимать решения об использовании нового знания для адаптирования оценок и планов к новым, уникальным или просто конкретным ситуациям («почему»). Из полудесятка книг, которые я постоянно рекомендую своим клиентам, две принадлежат Майку. «Agile-подход к оценке и планированию» входит в мой список обязательных для изучения материалов для тех, кто хочет понять последние веяния в сфере agile-управления проектами.

*Джим Хайсмит,
директор по agile-практике в компании Cutter Consortium,
Флагстафф, Аризона, август 2005 г.*

Предисловие

«Agile-подход к разработке не годится для Yahoo! за исключением, быть может, небольших оперативных проектов, поскольку команды совершенно не занимаются планированием, а члены команд не могут оценить проделанную работу. Им приходится давать указания, что делать».

Это реальное высказывание, которое я неоднократно слышал, когда начинал руководить внедрением agile-подхода в Yahoo!. Люди, которые не понимают концепцию agile-разработки, полагают, что это просто отказ от документации и планирования, лицензия на свободное плавание. На самом деле такое представление предельно далеко от истины.

«Команды совершенно не занимаются планированием». Тот, кто говорит такое, забывает, что agile-команда раз в две недели посвящает полдня составлению перечня задач, которые необходимо выполнить, чтобы представить определенную полезную для пользователя функциональность через две недели. То, что команды распределяют процесс планирования на весь период реализации проекта, воспринимается как отсутствие планирования. Это не так, и agile-команды в Yahoo! создают продукты, которые нравятся нашим менеджерам по продуктам намного больше, чем продукты традиционных команд.

«Члены команд не могут оценить проделанную работу. Им приходится давать указания, что делать». Это классическое заблуждение. Упование на магическую способность менеджера по продукту или руководителя проекта предвидеть, что кто-то другой, эксперт в своем деле, может реально создать, самоубийственно для заказчика. Нередко такой подход на деле оборачивается простым соглашательством с нереалистичными целями заказчика. Члены команды при этом вынуждены работать круглые сутки и халтурить. А потом мы удивляемся, почему в нашей отрасли люди так быстро выгорают и почему так низок моральный дух.

Оценка и планирование – темы, по которым мне задают больше всего вопросов, особенно во вновь созданных командах. Простой подход к планированию не только итерации, но и проекта в целом – вещь, прямо скажем, неоценимая. Менеджеров по продукту волнует достижение целевых показателей по выручке и предсказуемое планирование релизов. Команды могут использовать гибкий подход и изменять направление разработки при необходимости, однако очень важно иметь стратегический план, которому они будут следовать. Кому нужно быстрое движение, если вы идете не в том направлении? Освоение искусства оценки сделанного и планирования дальнейших действий – важнейшее условие успеха для тех, кто хочет внедрить agile-подход в своей организации.

Учебный курс Майка по оценке и планированию пользуется наибольшей популярностью при освоении agile-подхода в Yahoo!. Он дает командам профессиональные знания и инструменты, необходимые, чтобы заниматься планированием именно в таком объеме, который требуется для оптимизации результатов. Это и в самом деле работает, если следовать рекомендациям Майка? Да. Эффект от применения agile-подхода в Yahoo! грандиозен. Команды, прошедшие курс Майка, сразу начинают использовать его рекомендации на практике. Мы стали выводить продукты на рынок быстрее, а команды буквально в восторге от agile-подхода.

Почему agile-подход к оценке и планированию более эффективен, чем традиционные методы? Он сконцентрирован на создании стоимости и формировании доверительных отношений между заказчиком и проектными командами. Обеспечение полной прозрачности и информирование об изменениях по мере их появления позволяют заказчику принимать наилучшие решения. На предыдущем месте работы я воочию наблюдал процесс перехода из вечного хаоса в состояние предсказуемости, когда мы действительно стали браться за проекты, которые могли выполнить. Заказчикам не каждый раз нравились наши ответы (им ведь все-

гда хочется получить продукт уже завтра), однако они как минимум верили, что мы говорим правду, и не чувствовали, что их постоянно водят за нос.

Эта книга предельно честна. В ней не говорится о том, как достичь абсолютной точности оценок. Это была бы напрасная трата сил в стремлении достичь недостижимого. Майк не пытается навязать вам удобные шаблоны, в которых остается только заполнить пробелы. Вместо этого он заставляет вас думать и учиться тому, как подходить к проблемам и успешно решать их. Одинаковых проектов и организаций не бывает, поэтому намного важнее научиться мыслить определенным образом и применять принципы. Майк перенес свой огромный опыт, полученный из реальной практики, и индивидуальный подход на страницы этой книги. Она живая и честная. Она определенно должна стоять в первых строках вашего списка учебной литературы.

*Габриэль Бенефилд,
директор Agile Product Development Yahoo!*

Благодарности

Я выражаю глубокую признательность официальным рецензентам этой книги. Том Поппендик, Стив Токи, Пол Ходжеттс, Майк Сирфос и Престон Смит дали мне полезные отзывы и рекомендации. Благодаря их вкладу книга стала намного лучше. В частности, я хочу поблагодарить Стива и Тома за то, что они не ограничились формальными обязанностями. Стив отметил целый ряд идей и концепций, которые я упустил, и подсказал несколько полезных источников информации. Но главное, он натолкнул меня на то, что стало впоследствии моей мантрой при проведении занятий по оценке и планированию: оценивай размер, определяй срок. Том, пожалуй, потратил на эту книгу больше сил, чем я. Он неустанно подчеркивал важность того, чтобы книга была ориентирована на команду в целом, а не только на руководителя проекта. Именно в разговорах с Томом я понял, что книга по планированию должна быть шире, чем просто ответ на вопрос «Когда мы уже закончим?». В широком контексте создания стоимости для наших организаций дать ответ на этот вопрос нетрудно.

Мой поклон Джону Гудсену из RADSoft. Первоначально мы с Джоном собирались писать эту книгу вместе. Увы, наши календарные графики не позволили сделать этого, однако я благодарен Джону за обсуждения набросков книги.

Одно из величайших благ интернета – возможность показывать книгу другим в процессе работы над ней. Рукопись этой книги висела на моем веб-сайте в течение 20 месяцев, и комментарии и замечания читателей здорово помогли довести ее до ума. Я особенно благодарен Брайану Амброджиано, Кену Ауэру, Саймону Бейкеру, Рэю Боэму, Лесли Боррелл, Кларку Чингу, Лайзе Кристин, Рейчел Дейвис, Майку Дуайеру, Хакану Эрдогмусу, Джону Фаваро, Крису Гарднеру, Джону Джилману, Свену Гортсу, Полу Грю, Сридхару Джайяраману, Анджело Кастроулису, Лайзе Катценмейер, Лассе Коскеле, Митчу Лейси, Патрику Логану, Кенту Макдональду, Эрику Петерсену, Майку Поулену, Дж. Рейзенбергу, Биллу Рамосу, Мэтту Риду, Джорджу Рейлли, Крису Риммеру, Оуэну Роджерсу, Кевину Резерфорду, Дику Шилу, Джеймсу Шилу, Кену Скотту, Карлу Скотланду, Алану Шаллоуэю, Джагадишу Шринивасавадхани, Мишель Слайджер, Карен Смайли, Хьюберту Смитсу, Виктору Сзалвею, Чарли Трейнору, Раджу Вагхрею, Рудигеру Вульффу, Скотту Уорли и Джейсону Йипу.

Я также хотел бы поблагодарить всех, кто принимал участие в проведении моего учебного курса по agile-подходу к оценке и планированию в последние два года, как в компаниях, так и на конференциях. Моя благодарность адресована также всем клиентам, особенно тем, у которых я проводил занятия по оценке и планированию и которые пользуются идеями из этой книги, в том числе таким компаниям, как Farm Credit Systems of America, Fast401k, High Moon Studios, Nielsen Media Research, Sun Microsystems, Ultimate Software, Vision Pace, Yahoo! и Webroot.

Как всегда, с командой издательства Prentice Hall было очень приятно работать. Пол Петралиа и Мишель Хаусли участвовали в проекте от начала до конца. Тиррелл Албо помогла преодолеть некоторые трудности с использованием системы FrameMaker. Я попросил прикрепить ко мне придирчивого редактора, чтобы книга получилась предельно хорошей. Этим редактором оказалась Кэти Симпсон, которая сделала именно то, что мне хотелось. Наконец, Лара Уайсон тщательно контролировала весь процесс превращения рукописи в книгу, которую вы держите в руках. Она без усталости отвечала на мои бесконечные вопросы и электронные письма.

Благодарю Боба Мартина за включение этой книги в свою серию. Дядя Боб – один из моих любимых писателей еще с той поры, когда он был редактором журнала *C++ Report*. Боб сделал для распространения идей agile-подхода в сообществе разработчиков программного обеспечения столько, что попасть в его серию книг – большая честь. Также я хочу поблагода-

рить Джима Хайсмита из Cutter Consortium и Габриэля Бенефилда из Yahoo! за предисловия. Работа с ними была удовольствием.

Что бы я ни сказал, этого будет мало, чтобы выразить признательность моим близким, создавшим условия для работы над этой книгой. Предполагалось, что значительную часть подготовки рукописи я проделаю во время своих поездок. Однако все сложилось иначе. Лоре, моей жене и партнеру во всех начинаниях, пришлось не покладая рук заниматься моими делами, делами нашей компании и этой книгой. Она многократно читала и перечитывала каждую главу. Без ее помощи я бы не сделал и десятой части того, что мы сделали вместе. Мои чудесные дочери Саванна и Делани настолько привыкли к постоянному уединению своего отца в кабинете, что мое появление стало казаться им странным. Я благодарю их за нежные объятия и поцелуи и за то, что они такие, какие есть.

Введение

Эта книга в основном посвящена планированию, под которым я понимаю ответ на вопрос «Что необходимо сделать и к какому сроку?». Однако ответ на него требует ответа на вопросы, связанные с оценкой («Насколько это объемно?») и составлением календарного графика («Когда это должно быть сделано?» и «Сколько будет готово к этому моменту?»).

Книга состоит из семи частей, в ней 23 главы. Каждая глава завершается обобщением ключевых моментов и вопросами для обсуждения. Поскольку оценка и планирование – это виды деятельности, которыми занимается команда в целом, я предполагаю, что книгу будут читать все ее члены, а потом собираться (возможно, раз в неделю) для обсуждения прочитанного и вопросов в конце каждой главы. Учитывая, что agile-подход к разработке программного обеспечения популярен во всем мире, я старался избежать чрезмерной концентрации внимания на США. С этой целью в книге используется универсальное обозначение валюты и денежные суммы выглядят как \$500, а не \$500, €500 и т. п.

В части I рассказывается, почему планирование так важно, рассматриваются проблемы, которые часто встречаются на практике, и цели agile-подхода. В главе 1 говорится о цели планирования, о том, что нужно для составления хорошего плана и что превращает планирование в гибкий процесс. Основные причины, по которым традиционные подходы к оценке и планированию дают неудовлетворительные результаты, рассматриваются в главе 2. Наконец, глава 3 содержит краткий перечень особенностей гибких методов и описание обобщенных принципов agile-подхода к оценке и планированию, которому посвящаются последующие части книги.

В части II представлено основное правило оценки, требующее, чтобы оценки размера и срока никогда не смешивались. Главы 4 и 5 знакомят читателя с пунктами и идеальными днями – двумя показателями, подходящими для оценки размера разрабатываемых компонентов. В главе 6 описываются методы оценки в пунктах и идеальных днях и дается представление о покере планирования. Глава 7 посвящена тому, когда и как проводить переоценку, а в главе 8 приводятся рекомендации, что лучше выбрать – пункты или идеальные дни.

Часть III «Планирование на основе стоимости» содержит рекомендации для проектной команды относительно того, как создать наилучший конечный продукт. В главе 9 перечислены факторы, которые необходимо учитывать в процессе приоритизации функций. В главе 10 представлен подход к моделированию финансовой отдачи от отдельной функции или набора функций, а также методы сравнения финансовой отдачи, с тем чтобы в первую очередь разрабатывались наиболее ценные функции. В главу 11 включены рекомендации по оценке и последующей приоритизации желательности функций для пользователей продукта. Глава 12 завершает раздел обсуждением вопроса о том, как разбивать крупные функции на более мелкие части, которыми легче управлять.

В части IV мы переключаем внимание на вопросы, связанные с составлением календарных графиков для проекта. Глава 13 открывается обзором этапов составления календарных графиков для сравнительно простого, выполняемого одной командой проекта. В следующей главе (14) рассматривается вопрос планирования итераций. Главы 15 и 16 посвящены выбору длины итераций для проекта и оценке первоначального темпа продвижения команды. В главе 17 детально разбирается составление календарных графиков для проекта с высоким уровнем неопределенности или с высокой чувствительностью к некорректности графика. Раздел завершается главой 18, в которой описываются дополнительные этапы при оценке и планировании проекта, осуществляемого несколькими командами.

После составления плана необходимо проинформировать о нем всю организацию и использовать его для мониторинга прогресса разработки. Именно этим вопросам посвящены три главы части V. В главе 19 рассматривается мониторинг плана релиза, а в главе 20 – плана

итераций. В последней главе этого раздела (21) разбираются вопросы информирования о плане и процессе его выполнения.

Часть VI состоит всего из одной главы – 22, в которой рассматривается вопрос, почему работает agile-подход к оценке и планированию. Эта глава является своего рода дополнением к главе 2, где говорится о том, почему традиционные подходы нередко дают неудовлетворительные результаты.

Заключительная часть (VII) также включает в себя только одну главу. Глава 23 представляет собой расширенный анализ примера, который повторяет основные моменты этой книги применительно к гипотетической ситуации.

Часть I

Проблема и цель

Чтобы уяснить суть agile-подхода к оценке и планированию, необходимо ясно понимать цель планирования. Именно этому вопросу посвящена глава 1 данного раздела. В главе 2 рассматриваются основные причины, по которым традиционные подходы к оценке и планированию дают неудовлетворительные результаты. Заключительная глава этого раздела содержит описание обобщенных принципов agile-подхода к оценке и планированию, которому посвящаются последующие части книги.

Глава 1

Цель планирования

*Планирование – это все. Планы – ничто.
Фельдмаршал Хельмут фон Мольтке*

Оценка и планирование критически важны для успеха проекта по разработке программного обеспечения любого размера и значимости. Планы определяют наши инвестиционные решения: мы можем взяться за проект, на выполнение которого, по нашим оценкам, потребуется полгода и \$1 млн¹, и отказаться от этого же проекта, если на него потребуется два года и \$4 млн. Планы помогают нам понять, кого нужно привлечь к работам по проекту в течение определенного периода. Планы помогают нам понять, как продвигается создание функциональности, которая нужна пользователям и получения которой они ожидают. Без планов мы открываем ворота для целого ряда проблем.

Процесс планирования, однако, сложен, а планы нередко получаются далекими от реальности. Как результат, команды зачастую впадают в одну из двух крайностей: они либо полностью отказываются от планирования, либо тратят столько сил на составление планов, что начинают верить в их правильность. Команды, которые отказываются от планирования, не могут ответить на такие фундаментальные вопросы, как «Когда это должно быть выполнено?» и «Можем ли мы ожидать выпуск продукта в июне?». Команда, затратившая слишком много сил на планирование, обольщает себя уверенностью в том, что план в принципе может быть «правильным». Ее план может быть тщательно проработанным, но вовсе не обязательно отличаться высокой точностью или полезностью.

Тот факт, что оценка и планирование – дело непростое, не является открытием. Это известно давным-давно. В 1981 г. Барри Боэм построил первую версию того, что Стив Макконнелл (Steve McConnell, 1998) позднее назвал «конус неопределенности». На рис. 1.1 показаны первоначальные диапазоны неопределенности Боэма в разных точках в процессе последовательного развития («каскадный процесс»). Конус неопределенности говорит о том, что на этапе оценки осуществимости проекта оценка обычно отклоняется от истины на 60–160 %. Иначе говоря, на проект, который, как ожидается, должен занять 20 недель, может потребоваться от 12 до 32 недель. После формулирования требований в письменном виде оценка может отклоняться на $\pm 15\%$ в любом направлении, т. е. плановый срок 20 недель может сократиться до 17 недель или вырасти до 23 недель.

¹ Не забывайте, что \$ – это универсальный, обобщенный символ для обозначения валюты.

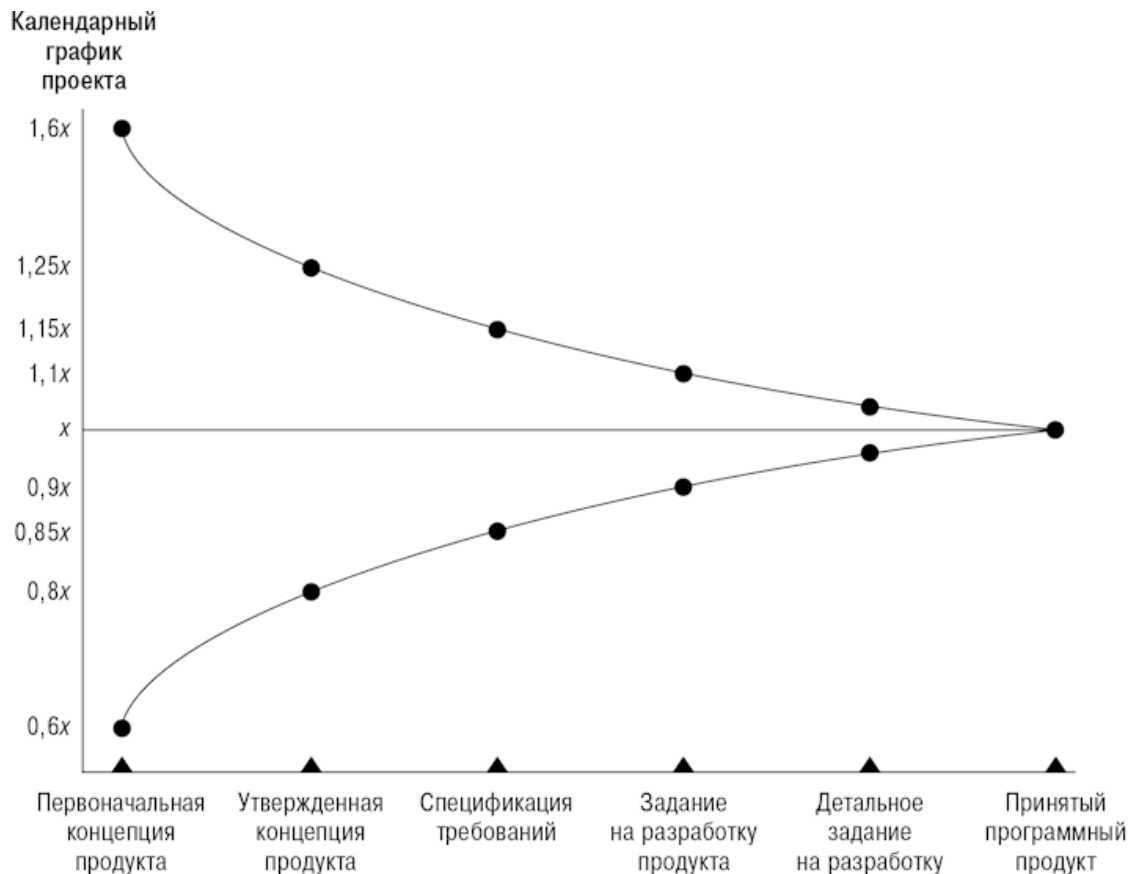


Рис. 1.1. Конус неопределенности сужается по мере выполнения проекта

Институт управления проектами (Project Management Institute – PMI) имеет сходную точку зрения на постепенное повышение точности оценок, однако он считает, что конус неопределенности должен быть асимметричным. PMI предлагает принимать начальный *уровень отклонений оценки* в диапазоне от +75 % до –25 %. Следующий этап – *бюджетные предположения* – предполагает диапазон отклонений от +25 % до –10 %, за ним следует этап *окончательной бюджетной оценки* с диапазоном отклонений от +10 % до –5 %.

Зачем это нужно

Если оценка и планирование настолько трудны и если точную оценку невозможно получить вплоть до последней фазы выполнения проекта, то зачем этим заниматься? Конечно, очевидной причиной является то, что в организациях, где мы работаем, от нас нередко требуют предоставления оценок. Планы и графики могут требоваться для таких вполне понятных целей, как планирование маркетинговых кампаний, планирование релизов продукта и обучение внутренних пользователей. Это очень важные потребности, и трудность оценки проекта не может служить основанием для отказа от составления плана или графика, который организация может использовать для их удовлетворения. Вместе с тем помимо этих номинальных потребностей существует значительно более фундаментальная причина не жалеть сил на оценку и планирование.

Оценка и планирование – это не просто определение сроков или календарных графиков. Планирование, особенно непрерывное планирование итераций, – это поиск стоимости. Планирование представляет собой попытку найти оптимальное решение всеобъемлющего вопроса

разработки продукта: что мы должны создать? Для ответа на этот вопрос команда анализирует функциональность, ресурсы и сроки. Ответ на данный вопрос нельзя найти одномоментно. Его ищут итерационно, шаг за шагом. В начале проекта мы, например, можем решить, что продукт должен иметь определенный набор функций, а его выпуск должен состояться 31 августа. Однако в июне оказывается, что лучше выпустить продукт немного позднее, но с более полным набором функций. А может наоборот: лучше сократить набор функций, но выпустить продукт чуть раньше.

Хороший процесс планирования поддерживает такой подход, обеспечивая:

- сокращение риска;
- снижение неопределенности;
- создание условий для принятия более качественных решений;
- формирование доверия;
- распространение информации.

Сокращение риска

Планирование повышает вероятность успеха проекта, обеспечивая идентификацию проектных рисков. Одни проекты настолько рискованны, что лучше не браться за них. Другие могут содержать функциональности, риски которых, если к ним подойти должным образом с самого начала, поддаются ограничению.

На обсуждениях, происходящих в процессе оценки, поднимаются вопросы, которые позволяют выявить подводные камни проекта. Допустим, вас просят оценить, сколько времени потребуется на интеграцию нового проекта со старой, построенной на основе мейнфрейма системой, о которой вы ничего не знаете. Это заставляет смотреть на функцию интеграции как на потенциальный риск. Проектная команда может устранить этот риск сразу, потратив определенное время на знакомство со старой системой. Альтернативно риск можно идентифицировать и учесть его в работе как отдельную величину или в виде диапазона и включить в общую неопределенность и риск.

Снижение неопределенности

В процессе реализации проекта команда создает новые функциональные возможности продукта, а также генерирует новые знания о продукте, используемых технологиях и своих собственных квалификациях. Крайне важно идентифицировать эти знания и учитывать их при итеративном планировании, которое должно помогать команде улучшать ее представления о продукте. Самым серьезным риском большинства проектов является риск создания несоответствующего продукта. Этот риск, однако, чаще всего полностью игнорируется. Agile-подход к планированию позволяет кардинально уменьшить (а в идеале устранить) такой риск.

Нередко цитируемые исследования Chaos Report (Standish Group, 2001) определяют успешный проект как такой, который выполнен в срок и в рамках бюджета и имеет все изначально предусмотренные функциональности. Это опасное определение, поскольку оно не учитывает того, что функциональность, казавшаяся хорошей до начала проекта, может оказаться не стоящей вложений, когда команда реально возьмется за дело. Если бы меня попросили дать определение неудачному проекту, то в числе прочего я бы назвал «проект, в котором никто не высказал более удачных идей, чем включенные в исходный перечень требований». Мы приветствуем такие проекты, в которых инвестиции, календарные графики и решения по функциональностям периодически переоцениваются. Проект, имеющий все предусмотренные в первоначальном плане функциональности, не обязательно успешен. Пользователи продукта

и клиент вряд ли будут довольны, если хорошие новые функциональности будут принесены в жертву средненьким просто потому, что те заложены в первоначальный план.

Условия для принятия более качественных решений

Оценки и планы помогают нам принимать решения. Как организации определить, стоит ли браться за тот или иной проект, не имея оценки стоимости и затрат по проекту? Помимо поддержки решений относительно принятия проектов оценки позволяют гарантировать, что мы работаем над самыми ценными проектами. Предположим, что организация рассматривает два проекта, один из которых оценивается в \$1 млн, а другой в \$2 млн. Во-первых, календарные графики и оценки затрат нужны организации для того, чтобы определить, есть ли смысл заниматься этими проектами; не окажутся ли они настолько продолжительными, что не уложатся в рыночное окно? Не окажутся ли они настолько затратными, что потеряют смысл? Во-вторых, оценки и план нужны организации для того, чтобы решить, за какой проект взяться. Может оказаться, что у организации есть возможности реализовать один проект, оба проекта или ни один из них, если затраты слишком высоки.

Оценки необходимы организации для принятия и других решений, помимо решения о том, браться за проект или нет. В некоторых случаях штат исполнителей проекта более важен, чем календарный график. Так, реализация проекта может потерять смысл, если она требует участия ведущего разработчика организации, который полностью занят в другом проекте. Вместе с тем если удастся составить план, показывающий, как обойтись без участия ведущего разработчика, то за реализацию проекта, возможно, стоит взяться.

Многие решения, принимаемые в процессе планирования проекта, являются компромиссными. Например, в любом проекте неизбежно приходится искать компромисс между временем разработки и затратами. Нередко наименее затратный путь разработки системы – это нанять одного хорошего программиста и позволить ему работать над созданием продукта 10 или 20 лет с возможностью отвлекаться на освоение соответствующей профессиональной сферы, совершенствование в области администрирования баз данных и т. п. Очевидно, однако, что возможность ждать появления готового продукта 20 лет редко когда выпадает, поэтому мы поручаем работу команде. Команде из 30 человек, возможно, потребуется год (30 человеко-лет) на разработку, с которой один программист мог бы справиться за 20 лет. Стоимость разработки при этом возрастает, однако стоимость, создаваемая при получении продукта на 19 лет раньше, покрывает увеличение затрат.

Нам постоянно приходится принимать компромиссные решения в отношении функциональности, трудозатрат, издержек и времени. Стоит ли из-за той или иной функции откладывать выпуск версии продукта? Следует ли нам привлечь к проекту еще одного разработчика, чтобы включить конкретную функцию в ближайший релиз? Следует ли выпустить версию в июне или стоит отложить выпуск до августа и включить в нее дополнительную функцию? Следует ли нам купить данное средство разработки? Для принятия решений нам необходимы оценки как затрат, так и выгод.

Доверие

Частая надежная поставка обещанной функциональности рождает доверие между разработчиками и заказчиками продукта. Достоверные оценки обеспечивают надежность поставки. Оценки нужны клиенту для распределения приоритетов и принятия компромиссных решений. Оценки также помогают клиенту решить, сколько функций разрабатывать. Вместо того, чтобы потратить 20 дней и получить все, может быть, лучше ограничиться 10 днями и получить 80 %

выгод. Клиенты с неохотой идут на принятие подобных компромиссных решений на начальной стадии осуществления проекта, если оценки разработчиков не внушают доверия.

Достоверные оценки позволяют разработчикам двигаться в стабильном темпе. Результатом является высококачественная программа и снижение количества ошибок. Это, в свою очередь, повышает достоверность оценок, поскольку на такую во многом непредсказуемую работу, как устранение ошибок, приходится тратить меньше времени.

Распространение информации

План дает представление об ожиданиях и показывает, что может произойти в процессе выполнения проекта. План не гарантирует получения точного набора функций в точно определенную дату по заданной стоимости. Он содержит информацию и устанавливает набор базовых ожиданий. Планы, к сожалению, зачастую сводятся к определению конкретной даты, а все допущения и ожидания, которые привели к появлению на свет этой даты, забываются.

Допустим, вы спрашиваете меня, когда будет завершен проект. Я говорю вам, что через семь месяцев, однако не объясняю, каким образом у меня получился именно такой срок. Вы, конечно, скептически отнесетесь к моей оценке. Без дополнительной информации вам не удастся определить, хорошо ли я продумал этот вопрос и реалистична ли моя оценка.

Теперь представьте, что я представляю вам план, который оценивает срок завершения работ в семь – девять месяцев, показывает, какая работа будет выполнена в первые один или два месяца, содержит перечень ключевых допущений и обрисовывает наш совместный подход к оценке прогресса. В этом случае вы можете проанализировать мой план и понять, насколько ему можно доверять.

Что делает план хорошим

Хорошим считается такой план, который, по мнению заинтересованных сторон, является достаточно надежным для того, чтобы на его основе принимать решения. На начальном этапе осуществления проекта это может быть указание на то, что продукт будет выпущен скорее в третьем квартале, а не во втором и что он будет иметь примерно обрисованный набор функций. На более позднем этапе работ этот план, чтобы оставаться по-прежнему полезным для принятия решений, должен быть более точным.

Предположим, вы оцениваете и планируете новый релиз флагманского продукта компании. По вашим расчетам, новая версия будет готова к выпуску через шесть месяцев. Вы составляете план с описанием набора функций, которые определено будут иметься в новой версии продукта, и еще одного набора функций, которые могут быть включены в продукт в зависимости от успешности процесса разработки.

Другие сотрудники компании могут использовать этот план для принятия решений. Они могут готовить маркетинговые материалы, планировать рекламную кампанию, выделять ресурсы на переобучение ключевых клиентов и т. п. Этот план полезен до тех пор, пока он реально предсказывает, что будет происходить в процессе работы над проектом. Если разработка займет 12 месяцев вместо запланированных шести, то план нельзя назвать хорошим.

Вместе с тем если проект займет семь месяцев вместо шести, то план в определенной мере полезен. Да, он неточен и мог привести к принятию не совсем правильных решений. Однако поставка продукта через семь месяцев при осуществлении проекта с расчетным сроком шесть месяцев вовсе не конец света и определено укладывается в пределы допустимой погрешности РМІ для бюджетных оценок. План, несмотря на его неточность, был бы еще полезнее, если бы он регулярно корректировался по мере выполнения проекта. В этом случае задержка поставки на один месяц не стала бы неожиданностью ни для кого.

Что делает планирование гибким

Эта книга посвящена agile-подходу к планированию, а не гибким планам. Планы – это документы и цифры, статическое описание наших представлений о развитии проекта в неопределенном будущем. Планирование – это вид деятельности. Agile-подход предполагает перенос акцента с планов на процесс планирования.

Agile-подход позволяет сбалансировать вкладываемые в планирование усилия и трудозатраты с учетом того, что план будет пересматриваться в процессе осуществления проекта. Никто не собирается менять план ради изменений, изменения вносятся потому, что мы получаем новую информацию или исправляем ошибки. Мы можем получить информацию, например, о том, что пользователи хотят расширить конкретную функцию или, наоборот, урезать ее, или узнать, что простота использования продукта значительно важнее, чем казалось вначале, или обнаружить, что программирование на новом языке занимает больше времени, чем ожидалось. Финансовые последствия каждого такого изменения можно оценить и, если это целесообразно, изменить план и календарный график.

Вновь выясненные обстоятельства влияют на наши планы. Это означает, что нам нужны такие планы, которые можно легко изменять. Именно поэтому процесс планирования становится более важным, чем сам план. Знания и представления, которые мы получаем в процессе планирования, продолжают существовать и после того, как от старого плана отказываются и заменяют его новым. Таким образом, под гибким понимают такой план, который легко поддается изменениям.

Изменение плана само по себе не означает изменение дат. Мы можем сделать это, а можем и не делать. Однако, если оказывается, что мы заблуждались в отношении определенного аспекта целевого продукта и нужно устранить ошибку, в план необходимо внести изменения. Существуют разные способы корректировки плана без изменения даты. Можно отказаться от функции, можно сократить ее объем, можно увеличить численность работников, занятых в проекте, и т. д.

Поскольку мы признаем, что не можем с абсолютной точностью определить все аспекты проекта с самого начала, нам не нужно пытаться запланировать все и вся на начальном этапе. При agile-подходе планирование осуществляется более или менее равномерно на протяжении всего срока реализации проекта. Вслед за планированием релиза, закладывающим фундамент, выполняется серия раундов планирования, и весь процесс многократно повторяется по мере осуществления проекта.

Итак, при определении agile-подхода к планированию мы установили, что он:

- фокусируется на планировании, а не на плане;
- поощряет изменения;
- приводит к составлению планов, легко поддающихся изменению;
- распределяет процесс планирования по всему сроку осуществления проекта.

Резюме

Оценка и планирование критически важны, однако сложны и подвержены ошибкам. Так или иначе, отказываться от них просто из-за трудности нельзя. Оценки, данные в начале проекта, значительно менее точны, чем оценки, полученные позднее. Графическое представление процесса постепенного повышения точности оценок называют *конусом неопределенности*.

Цель планирования – получение оптимального ответа на глобальный вопрос разработки продукта – вопрос о том, что именно создавать. Ответ включает в себя описание функций, ресурсов, а также календарный график. Ответ на этот вопрос, подкрепленный снижающим

риск процессом планирования, сокращает неопределенность, дает основу для объективного принятия решений, устанавливает доверие и обеспечивает распространение информации.

Хорошим является такой план, который достаточно надежен для того, чтобы на его основе принимать решения относительно продукта и проекта. Agile-подход к планированию сфокусирован больше на планировании, а не на создании плана, поощряет изменения, приводит к составлению планов, легко поддающихся изменению, и распределяет процесс планирования по всему сроку осуществления проекта.

Вопросы для обсуждения

1. Эта глава открывается утверждением о том, что чрезмерное планирование и отсутствие планирования одинаково опасны. Какой объем планирования оптимален для вашего текущего проекта?
2. Какие еще доводы в пользу планирования вы можете привести?
3. Вспомните один или два наиболее успешных проекта, в которых вы участвовали. Какую роль планирование играло в этих проектах?

Глава 2

Почему планирование дает неудовлетворительные результаты

Ни один план не выдерживает реального столкновения с противником.

Фельдмаршал Хельмут фон Мольтке

В предыдущей главе говорилось, что целью планирования является итеративное приближение к получению оптимального ответа на вопрос о разработке совершенно нового продукта – вопрос о том, что именно создавать. Иначе говоря, какими функциональными возможностями должен обладать продукт, за какой срок его необходимо создать и сколько для этого потребуются ресурсов. Мы узнали, что планирование поддерживает этот процесс, обеспечивая снижение риска, уменьшение неопределенности в отношении облика продукта, создание основы для принятия более качественных решений, укрепление доверия и распространение информации.

К сожалению, традиционные подходы к планированию нередко подводят нас. Отвечая на комплексный вопрос об объеме / календарном графике / ресурсах по новому продукту, наши традиционные процессы планирования не всегда дают удовлетворительные ответы и продукты. В подтверждение сказанного приведу следующие данные:

- почти две трети проектов значительно превышают сметы затрат (Lederer and Prasad, 1992);
- 64 % функций, включенных в продукты, используются редко или вообще не используются (Johnson, 2002);
- срок выполнения среднего проекта превышает календарный график на 100 % (Standish, 2001).

В этой главе мы рассмотрим пять причин, по которым планирование дает неудовлетворительные результаты.

Планирование ориентировано на деятельность, а не на функцию

Критическая проблема традиционных подходов к планированию заключается в том, что они сфокусированы на выполнении той или иной деятельности, а не на поставке функциональности. Диаграмма Ганта для традиционно управляемого проекта, или структура распределения работ, идентифицирует виды деятельности, подлежащие выполнению. Именно по ней мы определяем прогресс команды. Первая проблема планирования по видам деятельности связана с тем, что клиенты не получают никакой стоимости от выполнения видов деятельности. Единицей стоимости для клиента является функция. Планирование, таким образом, должно осуществляться на уровне функций, а не видов деятельности.

Вторая проблема возникает при анализе ранее составленного традиционного календарного графика. Когда мы анализируем календарный график, в котором представлены виды деятельности, наше внимание приковано к поиску пропущенных видов деятельности, а не отсутствующих функций.

Дополнительные проблемы объясняются тем, что планы на основе видов деятельности (процессно-ориентированные планы) зачастую ведут к проектам, которые не укладываются в календарные графики. Сталкиваясь с невозможностью выдержать сроки, заложенные в календарный график, некоторые команды пытаются сэкономить время за счет неуместного сниже-

ния качества. Существует также практика принятия политики, которая ограничивает возможности внесения изменений в продукт, в том числе и очень ценных изменений. Вот основные причины, по которым в результате планирования по видам деятельности трудно уложиться в сроки, предусмотренные календарным графиком:

- запланированные работы не завершаются досрочно;
- запаздывание распространяется на последующие этапы календарного графика;
- работы не являются независимыми.

Каждая из этих проблем рассматривается в последующих разделах.

Запланированные работы не завершаются досрочно

Несколько лет назад я занимался двумя крупными проектами, которые отнимали много времени. Мне нужно было запрограммировать ряд интересных функций для продукта, а также подготовить документацию для аудита соответствия требованиям стандарта ISO 9001. Если программирование доставляло мне удовольствие, то подготовка документов – нет. Неудивительно, что я умудрился раздуть объем программирования так, что оно заняло чуть ли не все мое время и практически вытеснило подготовку к аудиту.

Я вовсе не одинок в таком подходе к работе. Если говорить начистоту, то подобное поведение настолько распространено, что у него есть даже свое название – закон Паркинсона (1993 г.). Этот закон гласит:

«Работа растягивается так, чтобы занять все отведенное на нее время».

Паркинсон говорит, что нам требуется столько времени на завершение какого-либо дела, сколько, на наш взгляд, будет позволено. Если на стене висит диаграмма Ганта, из которой следует, что на тот или иной вид деятельности отведено пять дней, то программист, которому поручена эта работа, будет стараться растянуть удовольствие на полные пять дней. Чтобы избежать досрочного завершения, он может, например, добавить в программу какие-нибудь лишние функции (практика, известная как *украшательство*). Или может использовать часть времени на изучение какой-нибудь новой технологии, которая, на его взгляд, полезна для дела. Единственное, на что он редко когда идет, так это досрочное завершение работы. Во многих организациях в случае досрочного завершения работы шеф может обвинить исполнителя в предоставлении раздутой оценки. Или, как вариант, шеф станет рассчитывать на досрочное выполнение и других работ. Зачем рисковать и нарываться на то или другое, когда лучше немного побродить по сети и сдать работу в срок?

Пять дней, отведенные в календарном графике на работу, – это, по существу, разрешение разработчику использовать именно столько на выполнение задания. Человеку свойственно при опережении графика заполнять сэкономленное время другими, более интересными для него занятиями.

Запаздывание распространяется на последующие этапы календарного графика

Поскольку традиционные планы составляются на основе видов деятельности, они по большому счету сфокусированы на взаимозависимости работ. Рассмотрим диаграмму Ганта (рис. 2.1), где представлены четыре вида деятельности и их взаимозависимости.

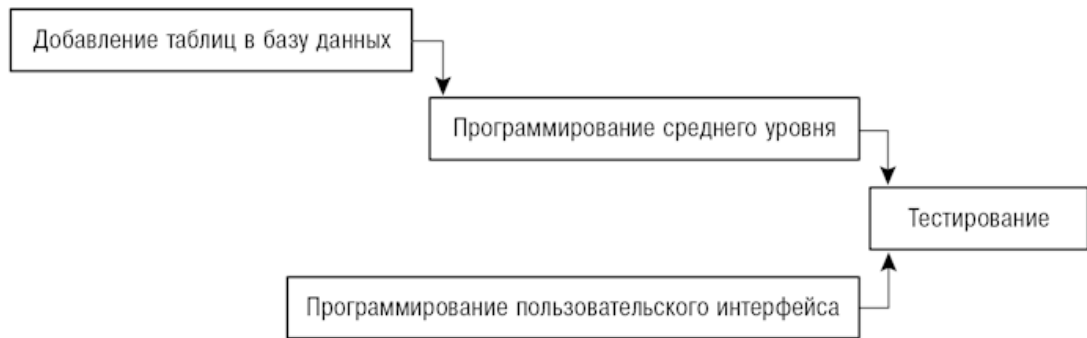


Рис. 2.1. Тестирование начнется с задержкой, если хоть что-нибудь пойдет хуже запланированного; оно начнется досрочно лишь в том случае, если всё без исключения будет лучше запланированного

Для досрочного начала тестирования требуется совпадение следующих событий:

- досрочное завершение программирования среднего уровня, которое зависит от срока завершения добавления таблиц в базу данных;
- досрочное завершение программирования пользовательского интерфейса;
- досрочное высвобождение тестировщика.

Ключевым моментом является то, что даже в этом простом случае досрочное начало тестирования зависит от выполнения всех трех условий. В то же время если для досрочного начала тестирования необходимо выполнение целого ряда условий, то для задержки тестирования достаточно наступления любого из перечисленных ниже событий:

- задержка завершения программирования пользовательского интерфейса;
- программирование среднего уровня требует больше времени, чем планировалось, и завершается позже;
- программирование среднего уровня укладывается в отведенное планом время, но начинается позже из-за задержки добавления таблиц в базу данных;
- недоступность тестировщика.

Другими словами, для досрочного начала необходимо сочетание условий, а для задержки начала достаточно одной причины.

Проблема осложняется тем, что, как мы уже говорили, работы очень редко завершаются досрочно. Это означает, что они обычно начинаются с опозданием и что запаздывание распространяется на последующие этапы календарного графика. Поскольку досрочное завершение – явление редкое, такой вид деятельности, как тестирование на рис. 2.1, начинается досрочно еще реже.

Работы не являются независимыми

Считается, что работы не зависят друг от друга, если сроки исполнения одной из них не влияют на сроки исполнения другой. При строительстве дома время подготовки котлована для фундамента не зависит от времени, необходимого для покраски стен. Когда работы не зависят друг от друга, задержку окончания одной из них можно компенсировать досрочным завершением другой. Многократное подбрасывание монеты – другой пример независимых видов деятельности. Если при первом подбрасывании выпадает орел, то это никак не влияет на вероятность выпадения орла при втором подбрасывании.

Являются ли работы, производимые в процессе разработки программного обеспечения, независимыми? Могут ли вариации сроков их завершения компенсировать друг друга? К сожалению, нет. Многие виды деятельности, связанные с разработкой программного обеспечения, нельзя считать независимыми. Например, если я пишу клиентскую часть приложения и первый экран отнимает на 50 % больше времени, чем запланировано, высока вероятность того, что каждый из оставшихся экранов также потребует больше времени. Если операции процесса разработки не являются независимыми, то вариации сроков их завершения не компенсируют друг друга.

В типичном плане проекта многие работы не являются независимыми, однако мы снова и снова забываем об этом. Когда кто-то задерживает сдачу первого из нескольких сходных элементов, мы слышим такое оправдание: «Да, я запоздал в этот раз, но дальше отставание будет наверстано». Это следствие надежды на то, что опыт, полученный при выполнении первой работы, позволит завершить оставшиеся работы раньше, чем предусмотрено планом. В реальности же подобная ситуация должна говорить нам, что, если какая-то работа занимает больше времени, чем запланировано, все остальные сходные работы тоже, скорее всего, потребуют больше времени.

Многозадачность приводит к дальнейшим задержкам

Второй причиной неудовлетворительных результатов традиционных подходов к планированию является многозадачность, под которой понимается одновременное выполнение нескольких задач. Многозадачность ужасным образом сказывается на производительности. Кларк и Уилрайт (Clark and Wheelwright, 1993) в своем исследовании эффектов многозадачности пришли к выводу, что время, посвящаемое создающей стоимость работе, быстро сокращается, когда человек занимается более чем двумя задачами. Этот эффект виден на рис. 2.2, где представлены результаты этого исследования.

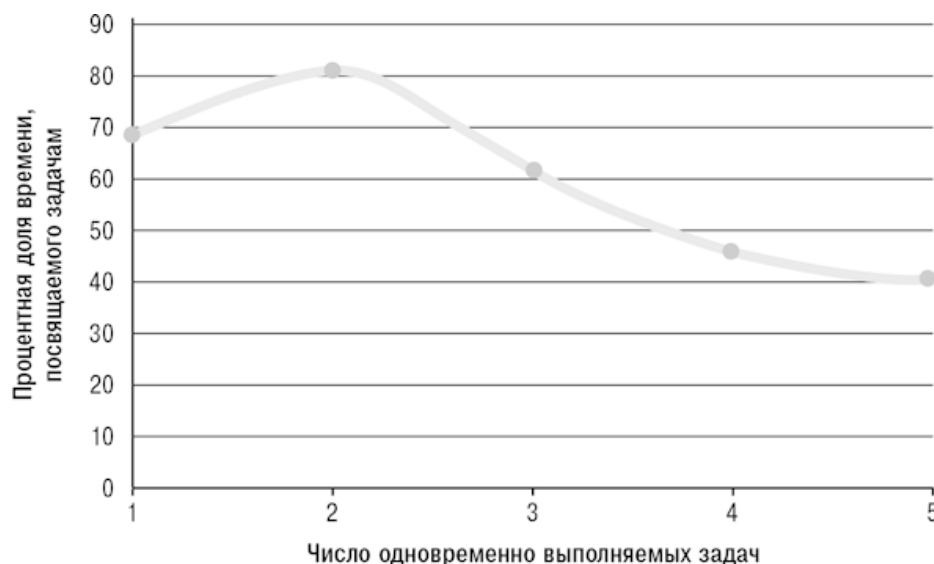


Рис. 2.2. Влияние многозадачности на производительность

По логике следует, что многозадачность помогает, когда вы занимаетесь двумя вещами, — если выполнение одной из них стопорится, вы можете переключиться на другую. Логично и показанное на рис. 2.2 быстрое сокращение времени, посвящаемого создающим стоимость задачам, когда их становится больше двух. Редко когда застопоривается более чем одна задача

за раз, а если мы работаем над тремя и более задачами одновременно, время на переключение с одной из них на другую оборачивается более ощутимыми затратами и бременем.

Многозадачность нередко превращается в проблему, когда какие-либо проектные работы начинают завершаться с запозданием. В этом случае взаимозависимость между видами работ становится критически важной. Разработчик, ожидающий завершения задачи своим коллегой, начинает просить последнего предоставить ему хотя бы сокращенную версию, чтобы можно было продолжить работу. Допустим, мне отведено 10 дней на работу с определенными изменениями базы данных, потом 10 дней на реализацию интерфейса прикладной программы (ИПП) для доступа к базе данных, а затем 10 дней на разработку пользовательского интерфейса. Эта ситуация отражена в верхней части рис. 2.3. Ваша работа не может начаться до тех пор, пока вы не получите ИПП от меня. Вы просите меня сделать необходимый минимум работы по ИПП, чтобы начать выполнение своей задачи. Аналогичным образом тестировщик просит меня сделать минимальную версию пользовательского интерфейса, чтобы он мог начать тестирование. Я соглашаюсь, и мой календарный график приобретает вид, представленный в нижней части рис. 2.3.

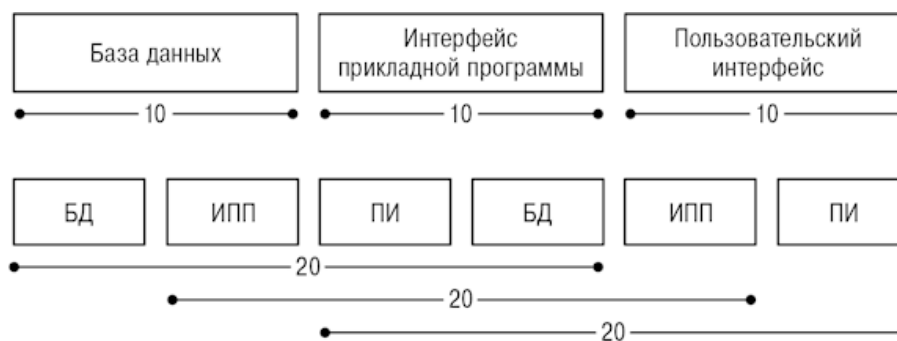


Рис. 2.3. Многозадачность оттягивает срок завершения работы и увеличивает продолжительность существования незавершенных работ

Это зачастую создает иллюзию скорости, однако, как видно на рис. 2.3, моя работа над базой данных и ИПП завершается позже, чем первоначально планировалось. Вряд ли стоит сомневаться в том, что это повлияет на последующие запланированные работы. Кроме того, в нашем примере каждый из затребованных видов работ остается незавершенным в течение 20, а не 10 дней, которые потребовались бы при последовательном выполнении работ.

Ситуацию усугубляет то, что рис. 2.3 не предполагает замедления исполнения работ в результате более частого переключения между ними. Кларк и Уилрайт показывают, что производительность снижается.

Многозадачность превращается в проблему при традиционном планировании проекта по двум основным причинам. Во-первых, работы обычно закладывают в план задолго до их начала, а эффективно распределить работы заранее невозможно. Закрепление работы за конкретным исполнителем, а не за группой углубляет проблему. Во-вторых, многозадачность заставляет фокусироваться на высоком уровне загрузки всех исполнителей в проекте, а не на создании необходимого резерва, позволяющего справиться с неизбежной изменчивостью типичных задач проекта. Загрузка всех на 100 % приводит к такому же результату, как и загрузка скоростного шоссе на 100 %, – движение останавливается и никто не может стронуться с места.

Функции не разрабатываются в соответствии с их приоритетом

Третья причина, по которой традиционное планирование не может обеспечить стабильное создание высокоценных продуктов, заключается в том, что работы, описанные в плане, не приоритизируются по их ценности для пользователей и клиента. Многие традиционные планы составляются в предположении, что все без исключения идентифицированные виды работ должны быть выполнены. Это означает, что работы обычно ранжируются и выстраиваются в определенной последовательности для удобства команды разработчиков.

В соответствии с традиционным мышлением если выполнению подлежат все виды работ, то для клиентов проекта не имеет значения, в какой последовательности они выполняются. Такой подход приводит к тому, что команда разработчиков занимается созданием функций в случайном, с точки зрения клиента, порядке. Затем в конце проекта команда, пытаясь уложиться в календарный график, начинает сокращать набор функций. Поскольку никто не старается выстроить работу над функциями в зависимости от их приоритетности, среди отброшенных функций оказываются такие, которые имеют более значительную ценность, чем функции, включенные в продукт.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.