**Agile. Оценка и планирование проектов**

**Автор: Майк Кон**

Не гибкий план, а гибкое планирование

Методология agile (гибкая методология разработки ПО) появилась около 20 лет назад. Она была создана в соответствии с международными стандартами проектного менеджмента специально для проектов по разработке программного обеспечения.

Традиционный подход к планированию проекта состоит в поэтапном построении плана, с указанием стадий осуществления проекта и дедлайнов. Метод agile отличается от этого подхода тем, что ориентирован не на финальный согласованный план, а на сам процесс планирования.

Все, кто имеет опыт работы в проектном бизнесе, знает, как тяжело на протяжении всей жизни проекта придерживаться плана, созданного в самом его начале. Иногда это просто невозможно. Методология agile учитывает эту особенность проектной работы и настраивает менеджеров на то, что изменения в плане с учетом меняющейся обстановки — это не недостаток плана, а особенность, присущая любому проекту. Задача менеджера — научиться создавать планы, которые можно безболезненно изменить, и уметь превращать каждое изменение в плане в полезный опыт. Книга Майка Кона построена на примерах из сферы ПО, но принципы agile отлично подходят для любых проектов.

Что такое agile-планирование

Поскольку каждый проект — это по определению комплекс действий, выполняемых в течение долгого времени командой людей, очевидно, что работу в проекте нужно планировать. Чем сложнее проект, чем больше задач и людей в него вовлечено, тем сложнее процесс планирования.

Поэтому часто люди либо решают не тратить время на планирование и действуют вообще без плана, что неизбежно приводит к хаосу, срыву сроков и неисполнению обязательств, либо, наоборот, стараются предусмотреть каждую деталь, тратят огромное количество времени и ресурсов на детализацию плана, а потом испытывают стресс и сожалеют о потраченном времени, когда в ходе проекта что-то начинает идти иначе.

Цель планирования — не расписать до мельчайших деталей весь список задач и назначить четкое время начала и окончания каждой. Это в принципе нереализуемо.

Планирование нужно, чтобы:

• оценить общий объем задач в проекте, предвидеть основные риски и попытаться их минимизировать;

• более четко сформулировать конечную цель всего проекта;

• наладить коммуникации между всеми участниками проекта, в том числе и заказчиками. План должен объяснять всей проектной команде основные задачи проекта, приблизительные сроки, служить инструментом установления взаимного доверия между заказчиками и исполнителями. В этом смысле план — одновременно техническое задание и список обязательств. Поэтому особенно важно, чтобы он содержал понятную всем информацию, но в то же время давал пространство для маневра.

По-настоящему гибкий план — это план, который легко изменить. При этом каждое изменений в таком плане — это не катастрофа, а новый инсайт, который остается в проектной команде, несмотря на все изменения.

Именно поэтому планирование более важно, чем план. Гибкое планирование, а не гибкий план — это основа методологии agile.

Основные черты гибкого планирования:

• фокусируется на процессе планирования, а не на плане как документе;

• позволяет вносить изменения в план и поощряет их;

• приводит к созданию планов, которые легко изменить;

• распространяется на весь проект, а не на его отдельные части.

Недостатки традиционного планирования

Типичные планы ориентированы на список действий, а не на список предоставляемых результатов. Широко распространенная диаграмма Гантта, которую используют для построения большинства планов проектов, расписывает конкретные действия членов проектной команды с привязкой к календарю. Но на самом деле заказчик ожидает от нас не выполнения определенных действий, а предоставления ему определенных результатов.

Диаграмма Гантта и другие традиционные инструменты имеют четкую привязку именно к датам. При этом действия, которые должна выполнить команда, взаимозависимы. То есть такой план оказывается негибким: если где-то происходит задержка, часть людей простаивает, потому что им не передали вовремя работу, а сдвиг на одном участке приводит к сдвигу по цепочке во всем плане.

Традиционное сетевое планирование не позволяет приоритизировать результаты. В итоге, если сроки сдвигаются и проект затягивается, команда отказывается от тех целей, которые она не успевает выполнить, хотя именно они, возможно, имеют больший приоритет для заказчика, чем те, которые уже выполнены.

Сетевое планирование не дает пространства для изменений. Это очень важно. У заказчика нет возможности изменить список целей проекта, не разрушив весь план. У команды тоже нет возможности поменять порядок работы. В итоге такой план, созданный еще до начала проекта, превращается в обязательство, а его изменения становятся нарушениями.

Чем планирование agile отличается от традиционного метода

Agile, прежде всего, базируется на другом наборе ценностей: на людях, а не на процессах, на сотрудничестве, а не на торге, на результатах, а не на плане.

Команда, работающая по принципам agile, подходит к проекту как к общей работе, а не как к списку отдельных, не связанных друг с другом задач. Такая команда стремится к достижению общего результата, а не к выполнению только своего участка.

Гибкое планирование отказывается от плана как единого списка одноуровневых задач или шагов. Цель agile-планирования — более общая, она состоит в том, чтобы к окончанию проекта добиться достижения тех условий, которые ставит заказчик. Эти условия состоят в сочетании свойств конечного продукта, сроков, стоимости и качества.

Для того чтобы эти принципы можно было осуществить, методология agile вводит несколько новых понятий.

• Уровни планирования. Гибкое планирование, в отличие от традиционного, происходит на трех уровнях: уровень всего проекта, уровень итерации и уровень рабочего дня. На первом уровне формулируется общая цель всего проекта. На втором — более подробные задачи для данного отрезка проекта. На третьем — задачи на конкретный день.

• Итерация — это отрезок проекта, на котором команда должна полностью выполнить определенный кусок работы. Причем ключевой момент здесь именно в полном выполнении — результат итерации можно показать заказчику. Если речь идет о ПО, то результат итерации — это функционирующий и уже протестированный фрагмент кода.

• Пользовательская история — это короткое (в одну строчку) предложение, описывающее одно из требований заказчика. Обычно история выглядит следующим образом: «Как <роль, например, новый пользователь>, я хочу <цель, например, иметь возможность зарегистрироваться на сайте>, чтобы <деловая выгода для заказчика, например, иметь возможность совершать покупки на сайте>». Одна итерация, в зависимости от величины, может включать одну или несколько историй.

• Эпопея — это сочетание нескольких пользовательских историй, объединенных логически.

• Баллы — это единица измерения, которой можно выразить размер истории.

• Идеальные дни — еще один способ измерения историй. Это количество чистого времени, которое команда тратит на проект.

Оценка размера и задач проекта

В первом разделе мы рассмотрели, чем гибкое планирование отличается от традиционного. Теперь посмотрим, как гибкое планирование работает.

Первый шаг планирования — оценка объема работы, который предстоит выполнить по проекту. Обычно люди оценивают работу по временной шкале: проект (итерация) займет 3 месяца (2 недели).

Краеугольный камень гибкой методологии состоит в том, что размер и длительность проекта — это принципиально разные вещи. При agile-планировании мы всегда оцениваем размер проекта и вычисляем его длительность.

Процесс выглядит следующим образом. Мы разбиваем весь проект на истории. Каждой истории присваивается определенный размер в баллах.

Перед вами 10 собак разной породы. Чтобы описать их размер, мы можем, нап-ример, выделить самую маленькую из них (допустим, йоркширского терьера), взять ее размер за единицу, и описывать остальных собак относительно нее: пудель будет в три раза больше, так что его размер будет 3 балла, овчарка — в 7 раз больше, так что ее размер — 7 баллов и так далее. Можно пойти другим путем: взять среднюю по размеру собаку (сеттера) и присвоить ей размер 5, а остальных сравнивать с ней. Так же можно оценивать и пользовательские истории.

Второй шаг — оценка, с какой скоростью команда будет работать. Очевидно, что если историю размером в 5 баллов команда сможет выполнить за 10 дней, то на историю в 3 балла ей понадобится 6 дней. После этого мы можем высчитать длительность всего проекта.

Скорость всегда позволит нам скорректировать любые ошибки, допущенные при оценке. То есть если, например, скорость команды на деле оказалась не 1 балл в два дня, как мы планировали, а 1 балл в 3 дня, то мы можем быстро пересчитать продолжительность всего проекта, не разрушив план.

Кроме того, если это вычисление покажет такую продолжительность, которая не устроит заказчика, то, благодаря тому, что проект разбит на истории, мы сможем предложить ему выбрать истории, от которых ему нужно будет отказаться, чтобы его пожелания по продолжительности проекта были соб-людены. Этот подход дает прозрачность во взаимоотношениях с закачиком, позволяет более точно предсказать для него конечный результат проекта, а также позволяет изъять из проекта или добавить в него определенное количество итераций, не перестраивая весь план целиком.

Кроме баллов, истории можно оценивать в идеальных днях. На первый взгляд это кажется проще, поэтому начинающим командам рекомендуют следовать этому методу. Суть его в том, что из 8 рабочих часов каждый сот-рудник проводит на проекте, допустим, 4 часа чистого времени. Остальное время у него занято звонками, написанием электронных писем, участием в совещаниях и т. д. Соответственно, два календарных дня — это один идеальный день. Данная методология требует оценки скорости работы каждого участника команды отдельно.

При этом нельзя забывать, что разница между идеальными и календарными днями может быть очень значительной, если члены команды задействованы, к примеру, на нескольких проектах.

Команды, работающие по методологии гибкого планирования, планируют вместе. Очень важно, чтобы пользовательские истории оценивал не человек со стороны, а именно испольнитель. При этом все члены команды понимают, что гибкое планирование по сути своей подлежит изменениям. Поэтому они уделяют время и внимание процессу планирования, но не позволяют этому процессу отнимать слишком много усилий. Как правило, наиболее подробно планируются ближайшие 2 итерации. Остальные, более удаленные во времени, планируются менее детально.

Основные методы оценки пользовательских историй:

• Экспертное мнение (спрашиваем тех, кто уже работал в похожих проектах, имеет опыт).

• Аналогия (пользуемся собственным опытом в похожих проектах).

• Разделение истории на составные части и оценка по частям.

При оценке историй можно играть в так называемый «оценочный покер» — просить каждого члена команды оценить историю, а затем озвучить и сравнить все оценки.

Если в процессе работы выявилось, что оценка размера при планировании была произведена неверно, нужно произвести переоценку этой истории, а также других, размер которых выводили на ее основании. Только так можно получать и накапливать опыт в команде.

При выборе метода оценки (баллы или идеальные дни) следует учитывать следующие факторы:

• Оценка в баллах лучше способствует сплочению команды, поскольку вся команда участвует в оценке. Когда речь идет об идеальных днях, каждый оценивает только свой участок работы.

• Оценка в баллах более объективна, поскольку не должна учитывать скорость работы и загруженность каждого отдельного сотрудника.

• Оценка в баллах более абстрактна и не привязывается к датам. Психологически человек склонен приравнивать оценку в идеальных днях к календарным дням.

• Оценку в идеальных днях проще объяснить людям вне команды.

Третий шаг — помимо оценки объема предстоящей работы, при планировании команде необходимо оценить ее суть. Прежде всего необходимо приоритизировать задачи проекта и выделить из них те, которые наиболее важны для заказчика.

Основные факторы приоритизации задач:

• Финансовая ценность задачи — то есть сколько прибыли принесет заказчику ее внедрение.

• Стоимость разработки — необходимо понимать, что разработка этой задачи не станет слишком дорогой для заказчика, чтобы в результате он смог окупить затраты на разработку.

• Ценность новой информации, которую получит заказчик в результате внедрения задачи.

• Размер рисков, которых удастся избежать в результате внедрения задачи.

Как правило, основная цель заказчика — получение прибыли. Он заказывает нам новый продукт, чтобы либо привлечь новых клиентов, либо предоставить новые платные услуги уже существующим клиентам, либо сэкономить деньги за счет оптимизации процессов. Здесь для приоритизации, скорее всего, придется прибегнуть к финансовому моделированию. Построив финансовую модель с учетом внедрения результатов проекта, можно будет точно посчитать, какую прибыль принесет заказчику разработка той или иной пользовательской истории — и какая из них, соответственно, более важная.

В процессе переговоров с заказчиком, особенно если речь идет о большом проекте со значительным количеством историй, всегда важно выделить более и менее важные истории.

Градация историй по степени важности:

• Обязательные истории — те, без которых заказчик не примет проект и не будет считать его исполненным.

• Линейные истории — такие, которые заказчик тоже хотел бы видеть в итоге. Чем больше их будет, тем лучше. Они дополняют обязательные.

• Бонусы — неожиданные для заказчика приятные истории, которые не оговаривались изначально, но которые команда может выполнить и представить заказчику, если останется время и будет техническая возможность.

Истории могут в процессе проекта кочевать из одной категории в другую, особенно это касается историй из первой и второй групп, поэтому очень важно держать постоянную связь с заказчиком и регулярно информировать его о ходе проекта.

Построение графика проекта

Как уже говорилось выше, методология agile предполагает планирование работы на нескольких уровнях.

На уровне проекта нужно определить его конечную цель. Для заказчика эта цель будет показывать, как быстро он сможет начать зарабатывать на внед-ренном продукте. В зависимости от этого он сможет вести свое стратегичес-кое планирование.

Для проектной команды конечная цель — тот маяк, к которому она движется. Важно, чтобы команда знала эту цель и не теряла ее из вида.

Чтобы сформулировать цель на уровне проекта, нужно:

• определить условия удовлетворенности заказчика — сроки, объем выполненной работы, использованные ресурсы;

• выбрать оптимальную для данного проекта длину итерации;

• оценить скорость работы команды;

• приоритизировать пользовательские истории;

• представить заказчику список историй, которые команда обязуется выполнить, и приблизительную дату завершения проекта.

Нужно регулярно проверять и сверять с заказчиком актуальность конечной цели проекта.

Планирование на уровне итерации становится более детальным, и его нужно проводить перед началом каждой итерации. При этом в ходе планирования важно избежать ситуации, когда каждому участнику проекта просто раздаются его задачи. Это полностью разрушает командный дух. Нужно планировать всю итерацию полностью, а не для каждого участника индивидуально. Итерации обычно длятся от двух до четырех недель, поэтому они всегда планируются в идеальных часах. В конце каждой итерации следует также проводить встречу для оценки выполненной работы и извлечения уроков на будущее.

Планирование итерации может основываться либо на скорости (вы до начала итерации выбираете такое количество историй, которое соответствует скорости работы команды и выбранной длине итерации), либо на выполнении обязательств (вы по ходу работы выбираете истории по одной, в зависимости от того, сколько у вас осталось времени до конца итерации).

Чтобы определить длину итерации, нужно учитывать:

• Общую продолжительность проекта. Чем он длиннее, тем длиннее может быть длина итерации (но не больше 4 недель).

• Уровень неуверенности заказчика, нечеткости конечной цели. Чем она выше, тем короче должны быть итерации, чтобы заказчик как можно чаще получал информацию о том, что уже выполнено (но не короче одной недели).

• Частота получения фидбека на сделанную работу. Чем чаще, тем лучше. Но если, например, заказчик может встречаться с командой только раз в три недели — соответственно, нужно планировать итерации так, чтобы к моменту встречи предоставлять очередной результат.

• Комфорт команды. Команда в процессе работы не должна быть ни слишком расслабленной, ни слишком загруженной.

Когда длина итерации выбрана — держитесь ее и старайтесь соблюдать до окончания проекта. Единственное исключение — постарайтесь, чтобы конец очередной итерации не выпадал на конец квартала — обычно это очень нервное время в любой организации, и у команды будет дополнительная нагрузка, а также сложно будет получить фидбек, потому что у всех будут другие дела.

Поскольку гибкое планирование не ставит перед собой задачу планировать ход проекта с точностью до дня, необходимо закладывать в план буферы. Буферы особенно полезны для тех проектов, где сложно сформулировать конечную цель или крайне высок риск серьезных негативных последствий в случае ошибок при планировании.

Буферы бывают двух видов:

• По задачам. Чтобы создать буфер по задачам, нужно приоритизировать истории, как объяснялось выше, и решить, какие нужно выполнить обязательно, а какие — необязательно. Необязательные истории — это и есть ваш буфер.

• Временные. Чтобы вычислить оптимальный временной буфер, нужно оценить размер историй с вероятностью 50% и 90% (например, с вероятностью 50% размер этой истории 5 баллов, но с вероятностью 90% она больше — 7 баллов). Затем разницу между этими оценками для разных историй нужно суммировать и извлечь из суммы квадратный корень — так вы получите количество баллов, которое затем нужно разделить на скорость — так вы получите длительность временного буфера.

Важно уметь объяснить заказчику, как вы пришли к той или иной оценке буфера. Буферы нельзя выбирать случайно, потому что так вы можете потерять доверие заказчика, пошатнуть его уверенность в надежности вашего планирования.

Agile-команда не может быть больше 10–12 человек, иначе будет сложно построить работу. Поэтому, если проект требует привлечения большего числа людей, лучше всего не объединять их в одну команду, а разбить на несколько команд. Чтобы работа в проекте была эффективной, нужно установить правила, которые будут соблюдаться всеми командами.

• Ввести единую систему оценки: все команды в одном проекте должны оценивать истории либо в идеальных днях, либо в баллах.

• Как можно точнее формулировать задачи перед каждой итерацией и коммуницировать их друг другу, чтобы они не пересекались.

• Нужно добавить в планирование буферы, которые возникнут, когда команды будут выполнять взаимозависимые задачи.

Отслеживание и коммуникация

Agile дает возможность отслеживать прогресс команды на протяжении всей работы. Такой проект предполагает открытость коммуникации и построение доверия между командой и заказчиком, чтобы заказчик в любой момент мог получить достоверную информацию о том, на какой стадии проект, и дать фидбек команде.

На уровне проекта отслеживание выглядит как общая информация о том, где находится команда по отношению к дате сдачи проекта. Эта информация выглядит обычно либо как график, на котором показано, сколько еще итераций или историй осталось до завершения проекта, либо как шкала, на которой выполненная часть работы и та, которая команде еще предстоит, выражены в процентах.

На уровне итерации отслеживание прогресса нужно, прежде всего, самой команде. Для коротких итераций (1–2 недели) прогресс удобно отслеживать на доске с задачами. Это либо магнитная, либо пробковая доска, разделенная на две части: выполненные и невыполненные задачи. Каждая задача, которая запланирована в итерации, записывается на отдельном листке бумаги. По мере выполнения задачи переносятся с одной стороны доски на другую.

Если итерация длиннее, чем 2 недели, можно использовать более сложные способы: например, такие же графики, как для отслеживания прогресса на уровне проекта. Но планирование и отслеживание должны, прежде всего, быть простыми и не отнимать много времени.

Что касается коммуникации, важно помнить три правила: коммуникация с заказчиком должна быть частой, честной и двухсторонней. Нужно регулярно информировать заказчика о ходе работы и обо всех изменениях в плане. При этом заказчик должен понимать (и нужно постоянно напоминать ему об этом), что планирование — это не точная наука, что в плане происходят изменения и это —норма. Также заказчики обычно ценят, когда в конце каждой итерации команда по собственной инициативе и без напоминаний предлагает им короткий отчет о проделанной работе.

10 лучших мыслей на одной странице

1. Планирование по методологии agile разделяет понятия величины проекта и его продолжительности. Величину проекта оценивают, а продолжительность вычисляют.

2. Планирование должно основываться на свойствах конечного продукта, которые хочет получить заказчик, а не на задачах, которые должна выполнить команда.

3. Планирование должно происходить на трех уровнях: всего проекта, итерации и одного рабочего дня.

4. Планирование — не точная наука, в ходе проекта план меняется несколько раз, и это необходимо объяснять заказчику.

5. Нужно постоянно отслеживать прогресс и сообщать заказчику, на каком этапе находится команда.

6. Изначальный план нужно постоянно переоценивать и держать заказчика в курсе изменений.

7. При планировании надо расставлять приоритеты.

8. В плане полезно оставлять буферы, не стоит планировать занятость каждого члена команды на 100%.

9. В процесс планирования надо включать всю проектную команду.

10. Планирование — это инструмент обучения. Любое изменение в плане должно превращаться в опыт, который можно использовать в дальнейшем.