

3 Планирование проекта

Строй реалистичные планы

Ах, дорогая, мой проект так хорошо распланирован, что я буду каждый день возвращаться домой вовремя.



Любое хорошее ПО начинается с хорошего плана.

В этой главе ты узнаешь, как построить план; как работать с клиентом, чтобы определить приоритеты его требований. Ты организуешь итерации, основываясь на которых, будет идти разработка. И наконец, ты построишь реалистичный план, который твоя команда сможет реализовать. Когда ты прочитаешь эту главу, ты будешь точно знать, как перейти от требований к разработке.

выпускай, что можешь, когда это нужно

Заказчику нужно ПО сейчас!

Клиенты хотят, чтобы разработка их ПО была завершена тогда, *когда оно им нужно*, и ни секундой позже. Ты уловил идеи клиента при помощи мозгового штурма, у тебя есть набор требований заказчика, описывающий все, что ему может потребоваться от ПО, ты даже добавил оценки к каждой опции, которые помогли тебе выяснить, сколько времени уйдет на проект. Проблема в том, что на разработку *всего*, что нужно заказчику, уйдет *слишком много времени...*

Наша оценка

489 дней

Сумма оценок требований заказчика

То, чего хочет заказчик

90 дней!



Совершенно очевидно, что ты не сможешь сделать все, чего хочет клиент, за 90 дней. Почему бы не обрезать проект в соответствии с приоритетами?



Упражнение

Orion's Orbits все еще хочет модернизировать систему бронирования, но не хочет ждать завершения проекта два года. Используй эти обрывки требований заказчика вместе с оценками и выбери те, которые нужно реализовать, чтобы объем работ **не превышал 90 дней**.

Название: <u>заказ перелета</u> Оценка: <u>15 дней</u>	Название: <u>оплата кредиткой</u> Оценка: <u>15 дней</u>	Название: <u>отзыв о полете</u> Оценка: <u>13 дней</u>
Название: <u>заказ еды на борту</u> Оценка: <u>13 дней</u>	Название: <u>заказ DVD полета</u> Оценка: <u>12 дней</u>	Название: <u>заказ сигары</u> Оценка: <u>15 дней</u>
Название: <u>просмотр косм. миль</u> Оценка: <u>14 дней</u>	Название: <u>выбор мест</u> Оценка: <u>12 дней</u>	Название: <u>просмотр карты астронавта</u> Оценка: <u>14 дней</u>
Название: <u>заказ места для животного</u> Оценка: <u>12 дней</u>		Название: <u>заказ спец. предложений</u> Оценка: <u>13 дней</u>

Общая оценка:

← Общая оценка всех требований заказчика.

Проблемы?

.....

.....

.....

.....

← Возникли проблемы? Запиши их сюда.

Предположения?

.....

.....

.....

.....

Запиши сюда все предположения, которые возникли по ходу работы.



Решение упражнений

Orion's Orbits все еще хочет модернизировать систему бронирования, но не хочет ждать завершения проекта два года. Тебе нужно было использовать обрывки требований заказчика вместе с оценками со с. 101 и выбрать те, которые нужно реализовать. Вот что получилось у нас:

Это требования заказчика, которые выбрали мы.

Название: заказ перелета
Оценка: ...15 дней...

Название: оплата кредиткой
Оценка: ...15 дней...

Название: заказ места для животного
Оценка: ...12 дней...

Название: заказ сигары
Оценка: ...15 дней...

По-моему, это требование звучит клёво!

Вот сколько уйдет на разработку всех опций.

Мы смогли включить и эту опцию, не превышая лимит.

Название: просмотр косм. миль
Оценка: ...14 дней...

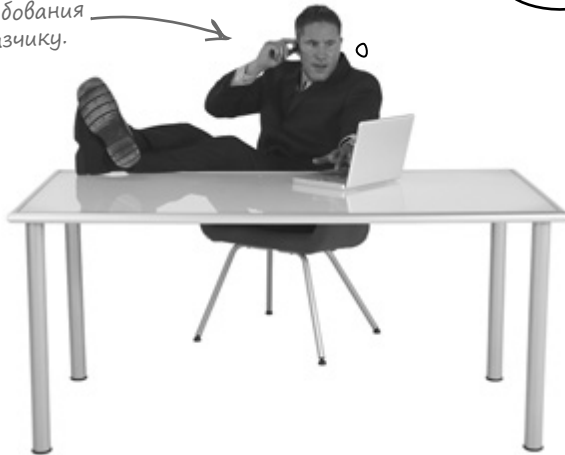
Название: заказ спец. предложений
Оценка: ...13 дней...

Наша общая оценка:

84

Но это совсем не то, чего я хотел!

Мы показали отобранные требования заказчику.



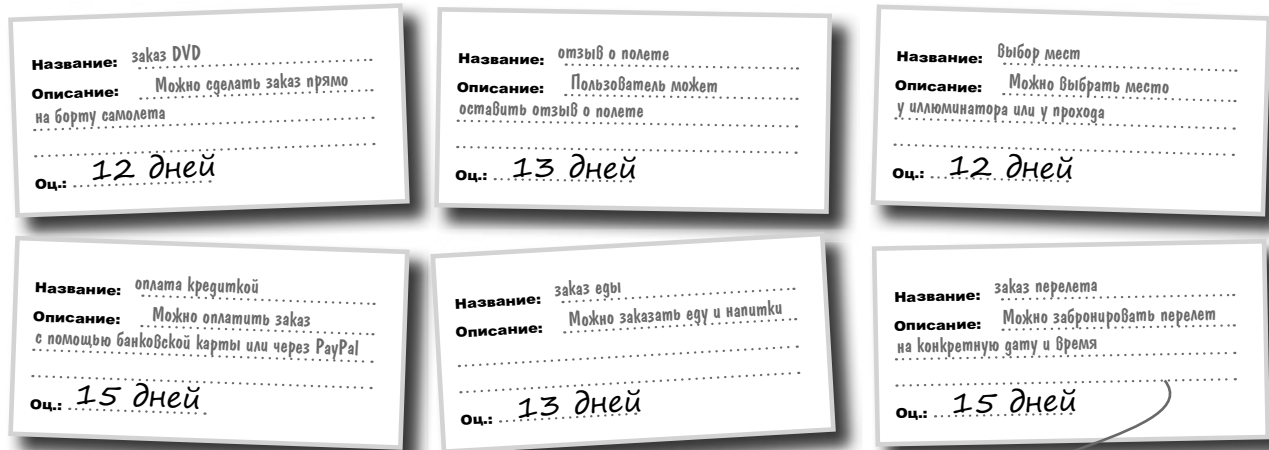
Приоритеты расставляет клиент

Похоже, директор Orion's Orbits недоволен, но можно ли его винить? После долгого выяснения того, что же ему нужно, мы полностью игнорировали его, решая, какие требования заказчика имеют высокий приоритет в нашем проекте.

Когда определяешь приоритеты, нужно *основываться на мнении заказчика*. Только он знает, что на самом деле необходимо. Так что когда пора *решить, что включить, а что оставить за кадром*, ты можешь советовать, но *выбор должен делать именно клиент*.

Пусть заказчик проставит приоритеты

Решать, каким требованиям какой проставить приоритет, *должен заказчик*. Чтобы помочь ему принять это решение, перемешай и разложи перед ним карточки с опциями. Попроси клиента разложить карточки в соответствии с приоритетом требований (начиная с наиболее важных), а потом выбери набор опций, которые нужно реализовать в Milestone 1.0.



Что такое Milestone 1.0?

Milestone 1.0 — это твой *первый и самый важный релиз* ПО. В отличие от итераций, в конце которых ты показываешь заказчику части ПО, чтобы узнать его мнение, на этот раз ты на самом деле *выпустишь ПО* (и получишь деньги за его разработку). Используй эти рекомендации при планировании Milestone 1.0.

Разложи все карточки перед заказчиком и попроси его упорядочить их в соответствии с приоритетом.

Обязательно... найди баланс между функциональностью и нетерпением заказчика.

Помоги заказчику понять, *что может быть сделано в указанный срок*. Все требования заказчика, которые не попадут в Milestone 1.0, не забыты, они просто отложены до релиза Milestone 2 или 3...

Це... старайся охватить всё.

В Milestone 1.0 должно быть *разработано то, что необходимо*, то есть набор опций, отражающих самые важные требования заказчика.

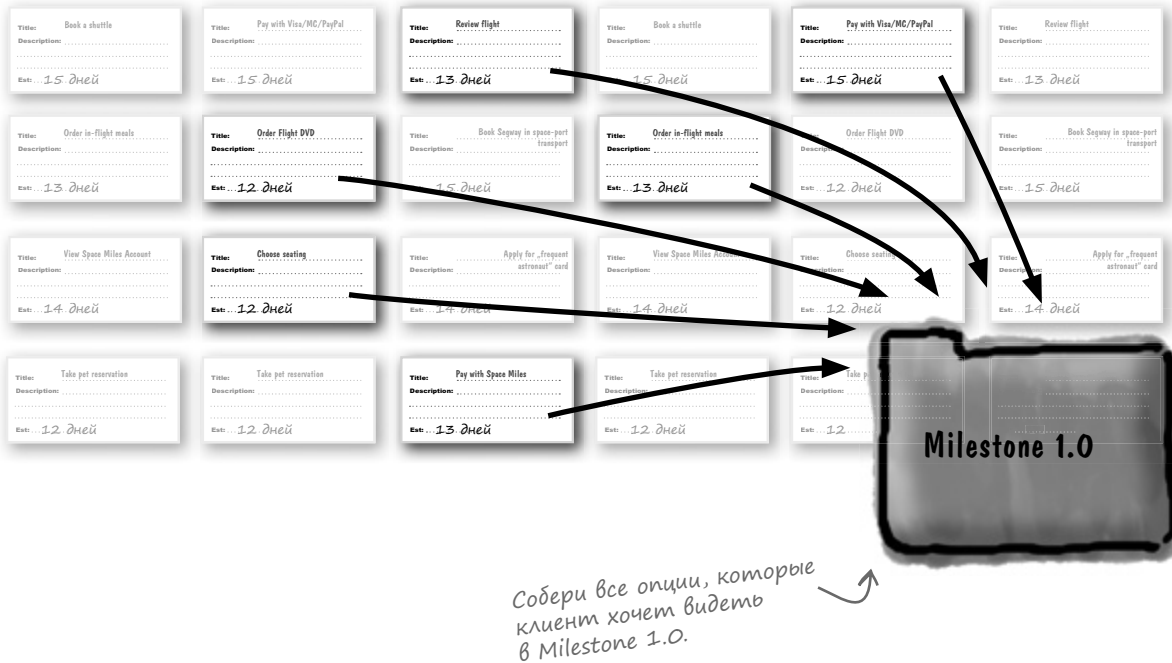
Це... принимай в расчет длительность (пока).

На этом этапе ты просто выясняешь у заказчика, какие из его требований самые важные. *Не заикливайся на том, сколько времени* уйдет на их разработку. Ты просто пытаешься разобраться в приоритетах клиента.

Не беспокойся, мы не забыли об оценке. Мы к ней еще вернемся.

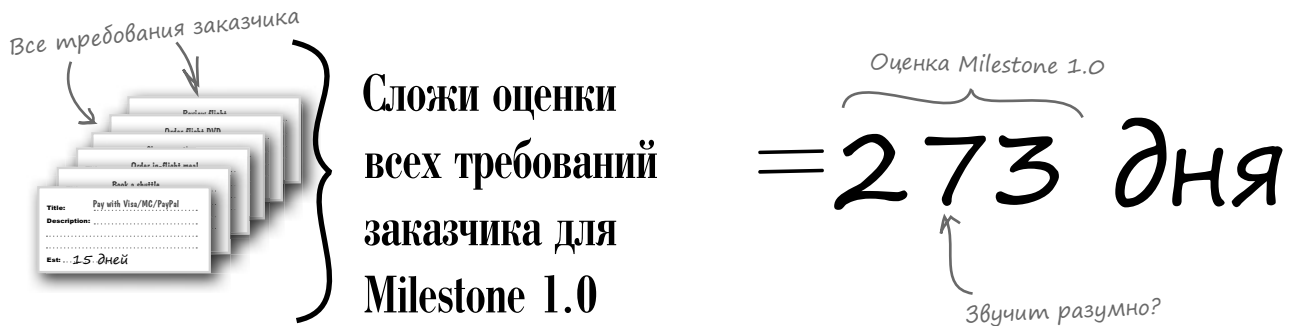
Мы знаем, что вошло в Milestone 1.0 (ну, наверное)

Из всех требований заказчика, записанных на основе его идей и распределенных во времени в соответствии с их приоритетами, заказчик выбирает те, которые должны войти в проект Milestone 1.0...



Проверка Milestone 1.0

Теперь ты знаешь, какие опции должны войти в проект Milestone 1.0. Пора выяснить, успеаешь ли ты завершить разработку вовремя, если включишь все важные опции...



Если не успеваешь разработать все опции, используй приоритеты

У тебя есть 273 дня работы над Milestone 1.0, а Orion's Orbits хочет, чтобы проект был завершен за **90 дней**. Пусть тебя это не пугает. Это стандартное положение дел. Заказчик обычно хочет от тебя больше, чем ты можешь дать. Тебе нужно поговорить с ним и перераспределить приоритеты так, чтобы справиться с работой вовремя.

Чтобы перераспределить приоритеты Milestone 1.0 с заказчиком...

1 Убери ненужные ОПЦИИ.

Первое, что стоит сделать, чтобы сократить время работы над Milestone 1.0, — это обрезать опции, убрав те требования заказчика, которые не являются совершенно необходимыми для работы ПО.

Когда ты объяснишь, в каких временных рамках находишься, большинство клиентов признают, что на самом деле им нужно не все, что они просили изначально.

2 Установи контрольную точку пораньше.

Постарайся разработать значительную часть ПО пораньше. Так заказчик будет меньше на тебя давить, и твоя команда сможет сконцентрироваться на дедлайне, до которого не так уж и много времени.

Не позволяй заказчику увеличить. Чем ближе дедлайн, тем больше ты и твоя команда на нем сконцентрированы.

3 Сосредоточься на **ОСНОВНЫХ** опциях.

Milestone 1.0 — это проект, включающий *только* те опции, которые необходимы для работ ПО. Другие могут быть реализованы в дальнейших версиях.

Часть Задаваемые Вопросы

В: В чем разница между контрольной точкой и версией?

О: Разница небольшая. На самом деле, ты можешь назвать первую контрольную точку «Версия 1», если тебе так больше нравится. Основное различие между ними в том, что контрольная точка отмечает момент, когда ты разрабатываешь значительную часть ПО и получаешь гонорар, в то время как термином «версия» описывается определенный релиз твоего ПО.

Разница не очевидна. Это просто способ понять, что «версия» — это метка, и больше ничего, а «контрольная точка» — момент, когда ты разрабатываешь значительную часть и получаешь оплату. При этом обе точки могут совпадать.

В: Какие опции моего ПО являются основными?

О: Основные опции твоего ПО — это минимальный набор, который необходимо реализовать, чтобы и твой клиент, и пользователи продукта могли им пользоваться.

Например, основные опции электронной обработки текста — это загрузка, редактирование и сохранение текста в файл. Все остальное второстепенно. Не важно, насколько полезны будут другие опции. Без возможности загружать, редактировать и сохранять документ текстовый редактор просто **нельзя использовать**.

Как правило, если ты можешь обойтись без какой-то опции, она не является основной, и при нехватке времени ее реализация может быть перенесена в следующие версии.

В: Я провел подсчеты и выяснил, что, как ни обрежай опции, я не смогу разработать то, чего хочет заказчик, вовремя. Что делать?

О: К сожалению, пора признать поражение. Если ты действительно не можешь разработать необходимое ПО вовремя и заказчик не желает сокращать набор требований, пора отойти от проекта и, по крайней мере, остаться честным с клиентом.

Другой вариант — привлечь к проекту нескольких новых разработчиков, чтобы справиться за более короткий промежуток времени. Однако пополнение команды существенно увеличит расходы и может не дать тебе тех преимуществ, которые тебе нужны.



Эй?! А нельзя привлечь еще несколько человек, чтобы сократить время? Добавим двух разработчиков и закончим в три раза быстрее, да?

Если у тебя работы на 273 дня, добавив двух людей, ты сократишь время разработки втрое, верно?

Дело не только во времени разработки.

Хотя привлечение новых разработчиков выглядит привлекательно, это совсем не просто. И, удвоив команду, на самом деле ты не сократишь время разработки вдвое.

Каждому новому члену команды нужно *включиться в проект*; нужно *разобраться с ПО, техническими решениями, компоновкой опций*, и пока они этим занимаются, *они не могут работать на 100%*.

Затем тебе нужно обеспечить нового члена команды нужным оборудованием. То есть купить новые лицензии и новое железо, но даже если тебе нужно просто загрузить и установить бесплатное ПО, *на это тоже потребуются время*, и это время нужно учитывать при пересмотре оценок.

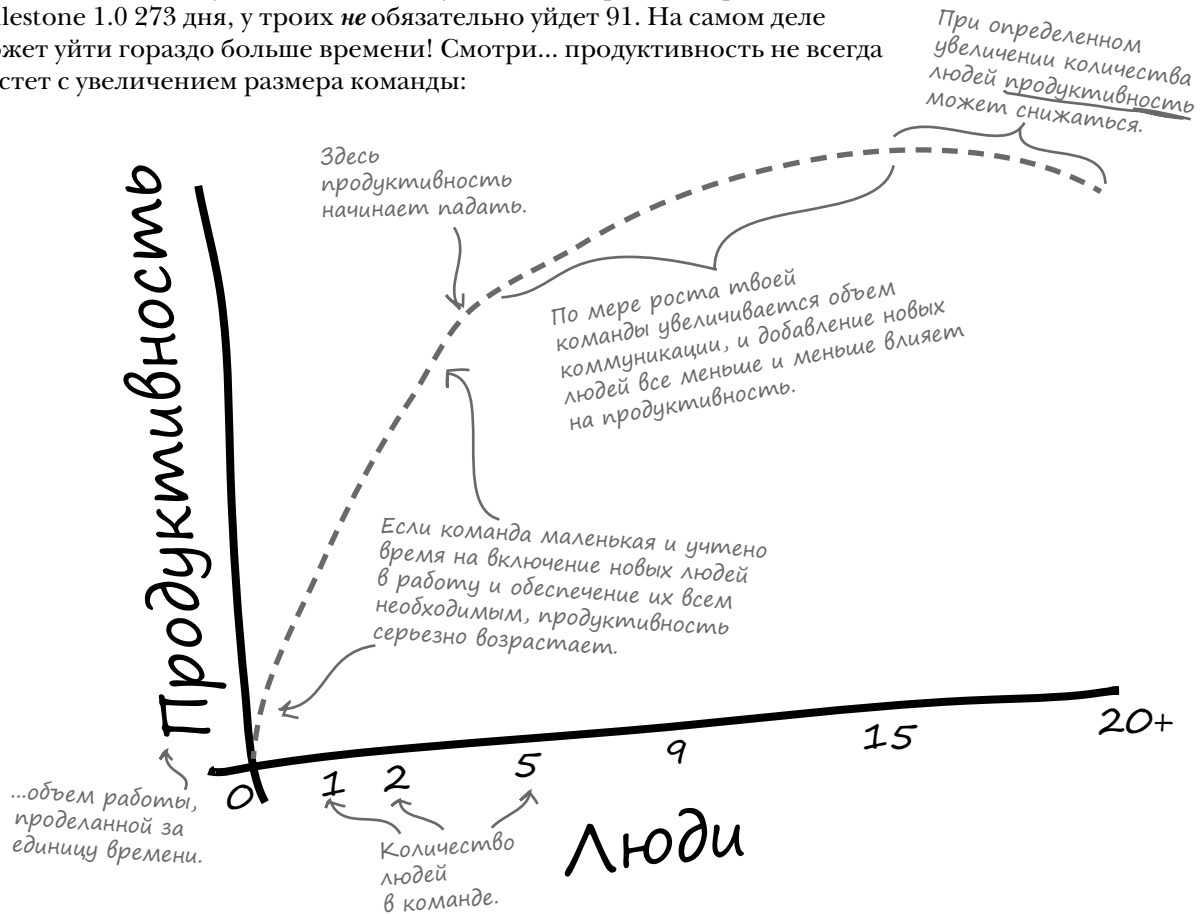
Наконец, новый разработчик усложнит задачу контроля над течением проекта. Управление проектом может занять все твое время. И при увеличении команды сложности коммуникации могут негативно сказаться на продуктивности работы и способности написать качественное ПО.

На самом деле есть ограничение по количеству людей в твоей команде, при превышении которого будет падать продуктивность, но оно серьезно зависит от проекта, команды и новых разработчиков. Лучше всего отслеживать работу команды, и если ты видишь, что *продуктивность снижается*, несмотря на *большее количество людей*, пора переоценить объем работ и длительность проекта.

Позднее в этой главе ты узнаешь о графике хода работ. Это замечательный способ отслеживать производительность твоей команды.

Привлечение большого количества людей может снизить продуктивность

Привлечение новых разработчиков не всегда срабатывает так, как ты ожидаешь. Если у одного человека уйдет на завершение проекта Milestone 1.0 273 дня, у троих *не* обязательно уйдет 91. На самом деле может уйти гораздо больше времени! Смотри... продуктивность не всегда растет с увеличением размера команды:



Часто задаваемые вопросы

В: Есть ли максимальный размер команды, который нельзя превышать?

О: На самом деле нет. Твой опыт может позволять тебе руководить командой из 20 человек, а вот с 21 уже ничего не получится. Или же ты можешь обнаружить, что продуктивность снижается при привлечении более трех человек. Лучше всего следить за продуктивностью и вносить поправки на основе наблюдений.

МОЗГОВОЙ ШТУРМ

Думаешь, объем проекта влияет на график? Тогда попробуй разбить работу на более мелкие части.

Приди к разумному проекту

Привлечение двух новых людей к проекту Orion's Orbits может оказать позитивное влияние. Посмотрим, как это работает.

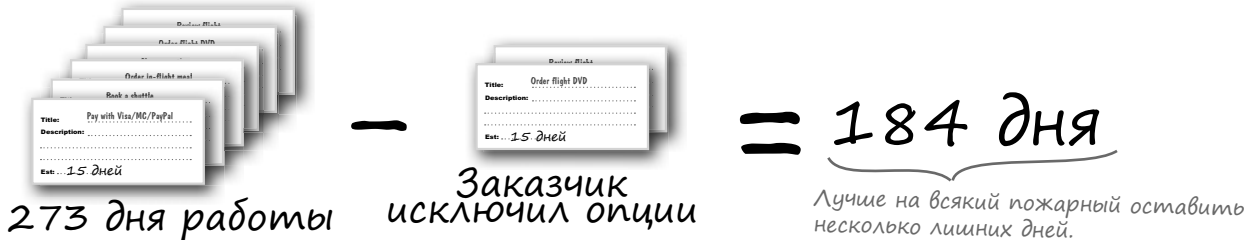
Сначала ты привлекаешь еще двух разработчиков...

Добавление двух людей (теперь вас трое) помогает, но не по мановению волшебной палочки. Два разработчика серьезно сократят время реализации необходимых опций, но не решат всех твоих проблем:



...затем пересмотри приоритеты с заказчиком...

Теперь у тебя есть хороший способ выяснить, что стоит исключить. У нас есть 190 дней рабочего времени и 273 дня работы. Так что нужно поговорить с клиентом и сократить объем работ где-то на 83 дня, перенося реализацию отдельных опций на светлое будущее.



В: Но 190 дней работы — это меньше, чем могут сделать наши разработчики, не стоит ли добавить несколько опций?

О: Общая оценка на самом деле не должна равняться именно 190 дням. Учитывая то, что мы имеем дело с оценками, которые далеко не всегда точны на 100%, и то, что в наших оценках мы обычно несколько оптимистичны, период в 184 дня — довольно близок к нужному. И мы можем быть уверены, что справимся с работой вовремя.

В: Как получилось 190 дней, когда вы добавили еще двух разработчиков?

О: Пока это всего лишь догадка. Мы предположили, что привлечение еще двух разработчиков позволит нам завершить работу, рассчитанную на 190 дней, за 90. Есть способы подтвердить эту догадку некоторыми доказательствами, используя метод под названием «скорость команды», но об этом мы расскажем позднее в этой главе.