

Одним из основных факторов успешного выполнения проекта является составление хорошего графика работ, который обладает следующими характеристиками:

- *основывается на детальной декомпозиции.* Он учитывает все виды деятельности, которые вы запланировали, на которые вы разбили ваш проект;
- *содержит все задачи в правильном порядке.* Почти всегда задачи в проекте взаимосвязаны друг с другом, поэтому их необходимо выстроить в правильной последовательности;
- *учитывает сторонние ограничения, не зависящие от команды*, т.е. учитывает разного рода события, которые происходят извне;
- *может быть завершен вовремя при наличии нужных ресурсов.* Если у вас есть график, в котором все подробно и детально расписано, но у него точка завершения по времени через месяц после дедлайна, то это не очень хороший график;
- *направлен на достижение целей проекта.*

Построение такого плана включает в себя ряд шагов, которые мы и разберём в этой лекции.

Планирование проекта

Планирование проекта предполагает множество взаимосвязанных итераций, итогом которых выступает единый сводный план. Рассмотрим основные задачи планирования (первые три из них мы разбирали в прошлый раз).

1. *Определение сути проекта.* В первую очередь необходимо понять, «что вы будете производить», прежде чем решать, «как это производить». Вы уточняете и детализируете цели и результаты проекта.
2. *Разработка стратегии управления рисками.* На этом этапе вы выявляете препятствия, которые могут возникнуть в ходе реализации проекта, оцениваете их, а затем определяете варианты действий по их устраниению.
3. *Декомпозиция проекта.* Невозможно анализировать сложную систему, рассматривая ее как единое целое. Поэтому, как правило, требуется сначала разбить ее на части (подсистемы) и лишь потом проводить анализ.
4. *Выявление зависимостей между задачами.* При планировании задач необходимо учитывать взаимосвязи между ними. Например, вы не можете начать писать документацию, пока не будут разработаны компоненты системы. В свою очередь проект не будет принят заказчиком, пока не будет написана документация. Поэтому необходимо определить, какие связи есть между задачами, в каком порядке их нужно выстроить.
5. *Оценка задач.* Для каждой задачи определяются объем работы и ее длительность, которая измеряется в часах/днях/неделях.
6. *Создание и оценка плана работ.* После того, как вы разбили ваш проект на задачи, установили последовательность их выполнения и оценили каждую из них, вы составляете план. Здесь уже вычисляются сроки выполнения всего проекта, указываются запланированные даты начала и завершения. Но следует учитывать, что первоначальный план, скорее всего, не будет соответствовать

имеющимся ограничениям по производственным ресурсам, времени или каким-нибудь другим параметрам. Поэтому есть еще пункт 7.

7. *Распределение и оптимизация ресурсов.* Планирование -- это итеративный процесс. С первого раза его очень сложно сделать правильно, поэтому эта стадия предполагает серьезную корректировку всего графика, который был построен до этого. Задачи перепланируются с целью оптимизации распределения людей и ресурсов, используемых в проекте. Вы получаете новую информацию, основываясь на которой можете более точно спланировать вашу деятельность.

Таким образом, эти шаги генерируют довольно много информации, необходимой для понимания того, как будет выполняться проект. Эта работа сильно итеративная, с первого раза вряд ли сразу получится получить хороший план.

Матрица зависимостей

Операция	Непосредственно предшествующие операции	Длительность
A. Установка компьютеров	-	1
B. Протяжка сети	-	2
C. Настройка сети	A, B	3
D. Установка программного обеспечения	C	1
E. Разработка регламента использования ПО	-	4
F. Обучение пользователей	D, E	3

После того, как вы определили задачи проекта, необходимо определить связи между ними. Зависимости между задачами влияют на даты планируемого начала или завершения задач. Наиболее распространенный тип связи -- это когда работа В не может начаться раньше, чем закончится работа А. Например, задача "интегрировать модуль X" не может начаться, пока не завершится задача "реализовать модуль X".

Матрица зависимостей представляет собой простой, но эффективный метод обнаружения связей между задачами. Вы выписываете все задачи, нумеруете их каким-нибудь уникальным идентификатором и пишите, за какой задачей должна выполняться конкретная задача.

Но следует учитывать, что когда у вас есть большое количество задач и между ними сложные зависимости, то такая форма представления может оказаться ненаглядной. Чаще всего используют так называемый сетевой график.

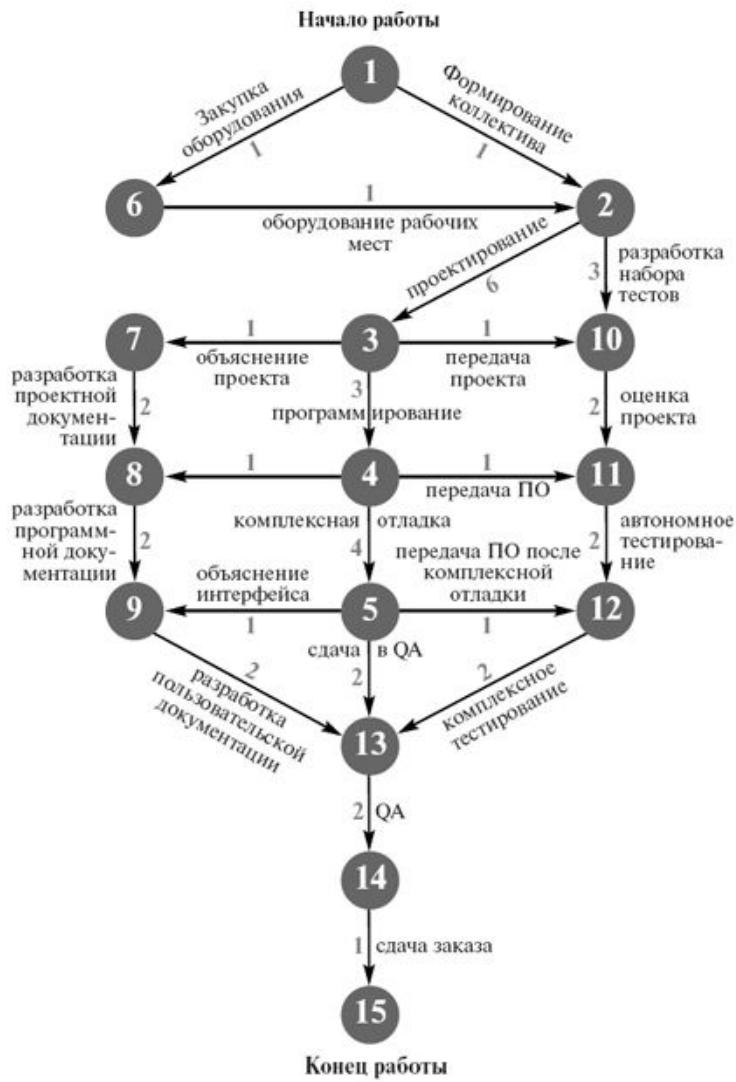
Сетевой график

Сетевой график -- это модель, отражающая зависимость и последовательность выполнения задач. Этот график передает ту же самую информацию, что и матрица зависимостей, но он более наглядный -- на нём гораздо более явны потенциальные проблемы и возможности для их решения. Например, вначале часто бывает

построить таблицу, а потом по ней уже график. Основными параметрами сетевого графика являются работа и событие.

Под работой подразумевается любой процесс, требующий затраты времени. Событие -- это промежуточный или окончательный результат одной или нескольких работ, необходимый для начала других работ. Событие совершается после выполнения всех работ, входящих в него.

Работу на сетевом графике изображают одной сплошной стрелкой. Продолжительность работы в единицах времени (часы/дни/недели) и наименование работ обычно пишут рядом со стрелкой. Каждое событие изображается кружком и нумеруется.



Здесь есть два важных момента.

1. Задачи, которые попадают в этот график -- это те самые задачи, которые вы получили при декомпозиции, которые вы потом будете “делать руками”. Т.е. тут не пишутся задачи, которые находятся в промежуточных узлах и обобщают задачи ниже уровня. Сюда заносятся только те задачи, которые вы реально будете делать, которые вы будете оценивать, из которых будет строиться все состояние.

2. Сетевой график должен отражать только зависимости между задачами. Не надо строить этот график, учитывая те ресурсы, которые у вас есть. Просто забудьте о них. Представьте, что у вас есть неограниченное число людей, денег и т.д. Изменение сетевых графиков из-за ограничений ресурсов является наиболее распространенной ошибкой при построении такого рода диаграмм. Тот факт, что не хватает людей или других ресурсов для одновременного выполнения нескольких задач, не имеет значения. Независимо от ресурсов, задачи все равно должны выполняться в том же порядке.

Основные правила разработки сетевого графика

При разработке сетевого графика целесообразно придерживаться следующих правил:

- Ни одна операция не может быть начата, пока все предшествующие связанные с ней операции не будут выполнены.
- Стрелки в сетевом графике отображают отношения предшествования и следования. На рисунке стрелки могут пересекаться.
- Каждая операция должна иметь свой собственный номер.
- Номер последующей операции должен быть больше номера любой предшествующей операции.
- Образование петель недопустимо (другими словами, не должно происходить зацикливания хода выполнения установленного набора операций).
- Условные переходы от одной операции к другой не допускаются (имеется в виду определение последовательности хода выполнения операций условиями типа: "Если будет достигнут успех, сделайте то-то...; если нет -- ничего не предпринимайте").

При включении любой операции в сетевой график необходимо определить для нее три отношения. Эти отношения могут быть определены в результате ответов на следующие три вопроса:

- Какие операции должны быть завершены непосредственно перед этой операцией?
- Какие операции должны следовать непосредственно за этой операцией?
- Какие операции могут выполняться во время выполнения этой операции? Какие операции можно назвать параллельными данной?

Внешние события и майлстоуны

Часто на сетевом графике бывает удобно указывать также внешние для проекта события и ключевые точки (майлстоуны) проекта. Они не требуют никакого времени и не влияют на продолжительность путей в сетевом графике, однако часто удобны для синхронизации хода работ. К ним можно отнести точки начала и завершения работ, даты выпуска версий ПО, события получения денег от заказчика или что-то ещё.

Оценка задач

После того, как вы нарисовали такой график, выяснили, что от чего зависит, нужно каждую задачу аккуратно оценить. Оценивают их обычно по времени. Только необходимо понимать, что это не фактическое время выполнение задачи, а время от

начала реализации этой задачи до получения конечного результата. Т.е. если вы сделали задачу за 20 мин, а потом запустили тесты, чтобы проверить, и они, например, идут два дня, то время выполнения данной задачи будет не 20 мин, как это ошибочно можно полагать, а 2 дня и 20 мин. Это очень важный момент, поэтому вы должны четко представлять, что является конечным результатом конкретной задачи.

Есть еще такое понятие, как объем работы, и это именно то, в чём скорее всего будут оценивать задачи технические специалисты. Для того, чтобы конвертировать объём работы в календарное время, вам нужно знать продуктивность людей, которые будут делать эту задачу. Есть очень простая формула:

$$\text{ДЛИТЕЛЬНОСТЬ РАБОТ} = \frac{\text{ОБЪЕМ РАБОТ}}{\text{ПРОИЗВОДИТЕЛЬНОСТЬ}}$$

Но эту формулу очень сложно применять, поскольку, как правило, вы не знаете ни числитель, ни знаменатель. Вы, конечно, можете примерно прикинуть и то, и другое, но с погрешностью, и эта погрешность может оказаться довольно большой.

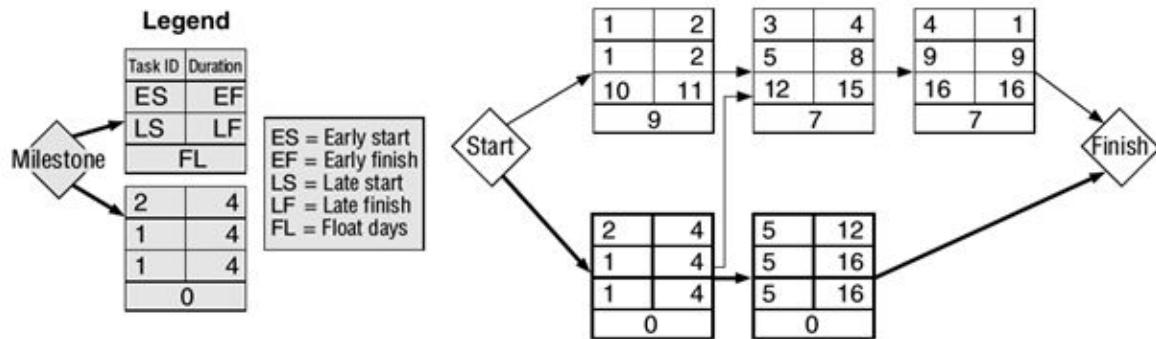
Все понимают, что разные программисты пишут код по-разному, с различной скоростью. И здесь нельзя просто взять и посчитать среднюю производительность. Представьте, что у вас есть 4 программиста, у первых двух высокая продуктивность, у вторых двух – низкая. Вы посчитали среднюю производительность, и затем задача досталась одному из них. Если она попала первым двум программистам, то они сделают ее быстрее, а если она попала другим двум, они просто не успеют. Ну и не стоит забывать, что один и тот же человек работает с различной производительностью в разные дни. Это сильно усложняет точную оценку, но в некотором приближении пользоваться этим всё равно приходится. Тут многое зависит от того, насколько хорошо вы знаете свою команду, а команда знает проект, который вы планируете.

Ещё при конвертации объёма работ во время не стоит рассчитывать, что каждый разработчик будет стабильно выдавать вам по 8 часов продуктивной работы каждый день. Продуктивно писать код по 8 часов в день довольно сложно, разработчики так или иначе оказываются вовлечены в разные совещания, планирование и оценку, проектирование, да и просто могут помогать коллегам с их задачами. Правильные коэффициенты тут тоже выясняются обычно опытным путём и сильно зависят от конкретных людей и принятых процессов.

Отдельная сложность при планировании работы людей, занятых на нескольких проектах сразу. Сложно совмещения проектов заключается в переключении контекстов. Разработка ПО -- крайне творческая задача, часто требующая серьёзного погружения в предметную область, имеющийся код и особенности задачи. Если человека постоянно дёргать туда-сюда между проектами и задачами, он просто большую часть времени будет тратить на переключение между ними, а на собственно решение задач будет оставаться гораздо меньше 8 часов в день. Ну и в чём то схожая ситуация с людьми, занятыми не на полную ставку (разного рода стажёры и т.п.), им может приходиться тратить какое-то время на синхронизацию с тем, что было сделано другими членами команды в их отсутствие.

Оценка графика работ

Предположим, что вы как-то оценили все ваши задачи. Теперь вы хотите построить общий график и вычислить сроки выполнения всего проекта. Делается это на основе того же самого сетевого графика, только теперь он немного видоизменяется.



По сути, справа нарисован сетевой график. Табличка слева -- это легенда, которая просто показывает, что эти кубики значат. Здесь используются следующие понятия.

- *Раннее начало (ER)* -- это самая ранняя дата, с которой может начаться задача. Т.е. эта величина показывает время, раньше которого задача не может быть начата.
- *Раннее окончание (EF)* определяется суммой раннего начала и продолжительности рассматриваемой задачи.
- *Позднее начало (LS)* -- это самая поздняя дата, когда задача может быть начата без задержки завершения проекта. Т.е. эта величина показывает время, позже которого задача не может быть начата без увеличения продолжительности всего проекта.
- *Позднее окончание (LF)* вычисляется как сумма позднего начала и продолжительности рассматриваемой задачи.

Первая строкка элемента сетевого графика: слева идентификатор (номер задачи), а справа -- длительность. Т.е. глядя на первый прямоугольник, вы понимаете: задача 1 длительностью в 2 единицы. Во второй/третьей строкке записываются раннее/позднее начало и раннее/позднее окончание.

Теперь о том, как это все вычислить. В самом начале у вас все ячейки пустые, кроме первой строки. Идентификатор задач и их длительность вы знаете.

Шаг 1. Прямой проход

Он называется так потому, что вычисления начинаются с исходного события и продолжаются до тех пор, пока не будет достигнуто завершающее событие. Прямой проход помогает вам определить раннее начало и раннее окончание для каждой задачи. Пойдем по порядку:

1. Первая задача. Она занимает два дня. Значит, это первый и второй день. Следовательно, early start = 1, early finish = 2. Идем дальше в задачу 2.

2. Вторая задача. Она ни от чего не зависит. Значит, ее можно начать выполнять сразу же (в первый день). Длительность = 4, поэтому early finish = 4.
3. Третья задача. Она не может начаться раньше, чем завершится задача 2. Поэтому у нее early start = 5, early finish = 8.

И так далее вы заполняете эти оптимистичные оценки.

Шаг 2. Обратный проход

Обратный проход определяет позднее начало и позднее окончание. Цель обратного прохода -- пройтись в обратном направлении от даты окончания проекта, чтобы определить, насколько поздно любая задача может начаться и завершиться. Т.е. вы берете последнюю дату и смотрите: "если мы в этот момент все закончим, то когда должна начать выполняться последняя задача?" И отнимаете от нее, сколько она длится. Затем смотрите, от кого она зависит. И таким образом идете в самое начало.

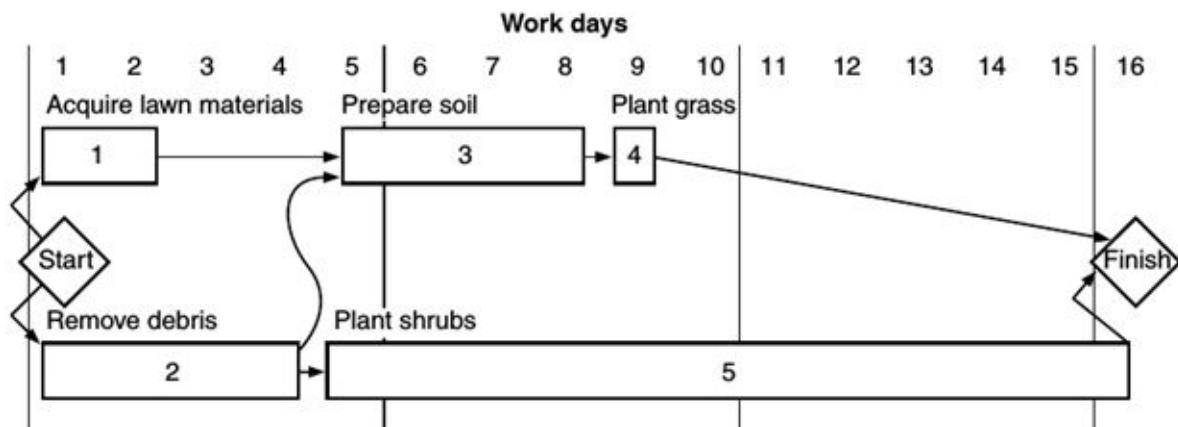
Шаг 3. Вычисление резервов

Резерв (float) вычисляется как разница между поздним и ранним началом. Он показывает, какой у вас есть запас времени между этими двумя величинами. Значение резерва вполне может быть отрицательным значением: это значит, что вы уже не успеваете к обозначенной дате завершения проекта.

Критический путь

Одной из ключевых особенностей построенного графика является критический путь. Это такой путь, у которого все задачи в ячейке резерва имеют ноль или отрицательные значения. Другими словами, критический путь -- путь от исходного до завершающего события, имеющий наибольшую длину (продолжительность) из всех путей. Критический путь является одним из показателей жизнеспособности графика. Это объясняется тем, что он демонстрирует минимальное время, которое займет проект. Поэтому увеличение длительности задач, лежащих на критическом пути, увеличивает общую продолжительность проекта. Соответственно, сокращение этих работ приводит к общему сокращению срока выполнения проекта. Заметим, что в сетевом графике может быть несколько критических путей. На графике выше критический путь выделен утолщенными линиями.

Другой формат представления сетевого графика



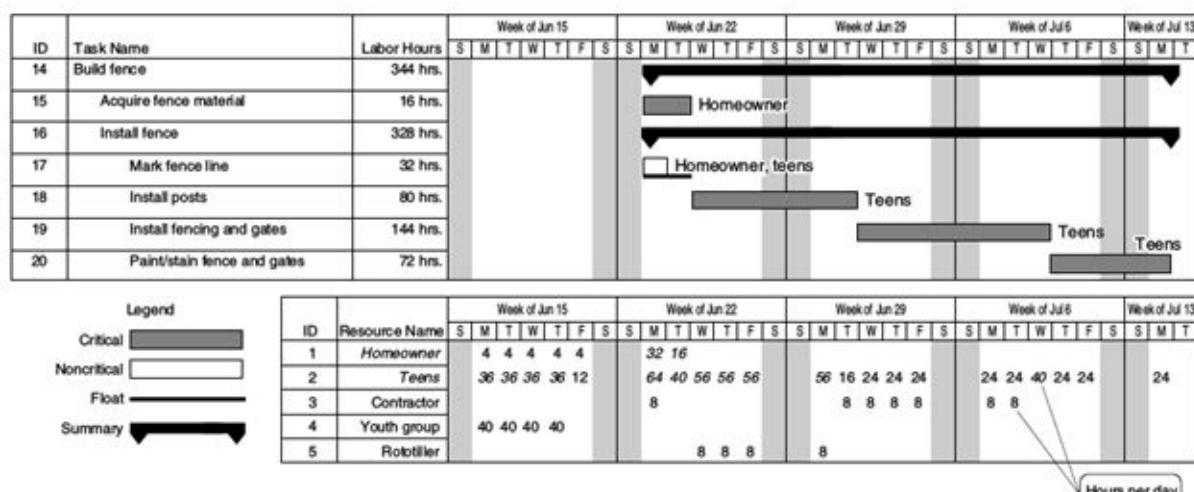
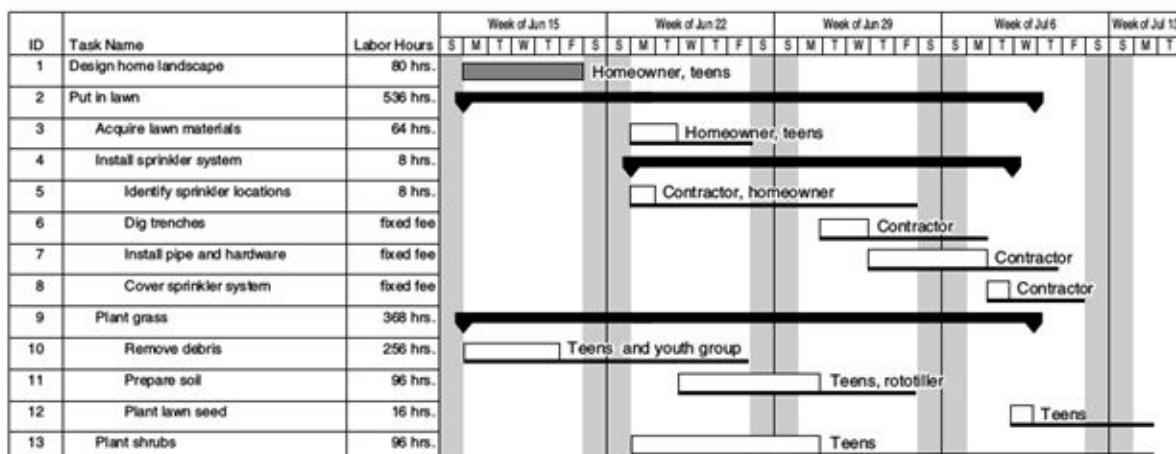
Такой график еще более наглядный, но его сложно строить “вручную” и он занимает много места. Поэтому если у вас много задач, для построения таких графиков пользуйтесь специализированными тулами.

Диаграмма Ганнта

Диаграмма Ганнта -- это популярный тип столбчатых диаграмм, который используется для иллюстрации плана, графика работ по какому-либо проекту. Она представляет собой отрезки, размещенные на горизонтальной шкале времени. Задачи, составляющие план, размещаются по вертикали. Каждый отрезок соответствует отдельной задаче. Начало, конец и длина отрезка на шкале времени соответствуют началу, завершению и длительности задачи.

Важно то, что здесь тоже все делается без учета ресурсов, т.е. вы показываете только зависимости между задачами. Эта диаграмма очень часто используется, поскольку она крайне наглядная и ее можно рисовать где угодно. Например, часто для этого используют Excel/Google Spreadsheets. (Единственная проблема, которая здесь будет -- это перемещение задач. Т.е. если вы поняли, что вы сначала сделали задачу 4, а потом задачу 3, то просто поменять строки местами не получится. Вам еще потом придется сдвигать закрашенные ячейки.)

Диаграмма Ганнта с указанием ресурсов



На этой диаграмме еще указывается дополнительная информация. Например:

-- составная задача проекта, т.е. задача, которая разбивается на несколько других (промежуточный узел дерева декомпозиции).

-- резерв.

Также на обычной диаграмме Ганнта нет информации о людях вообще. Т.е. не указывается, кто делает какую задачу. А здесь эти сведения добавляют. Такую диаграмму можно достаточно быстро читать и в ней закладывается довольно много информации.

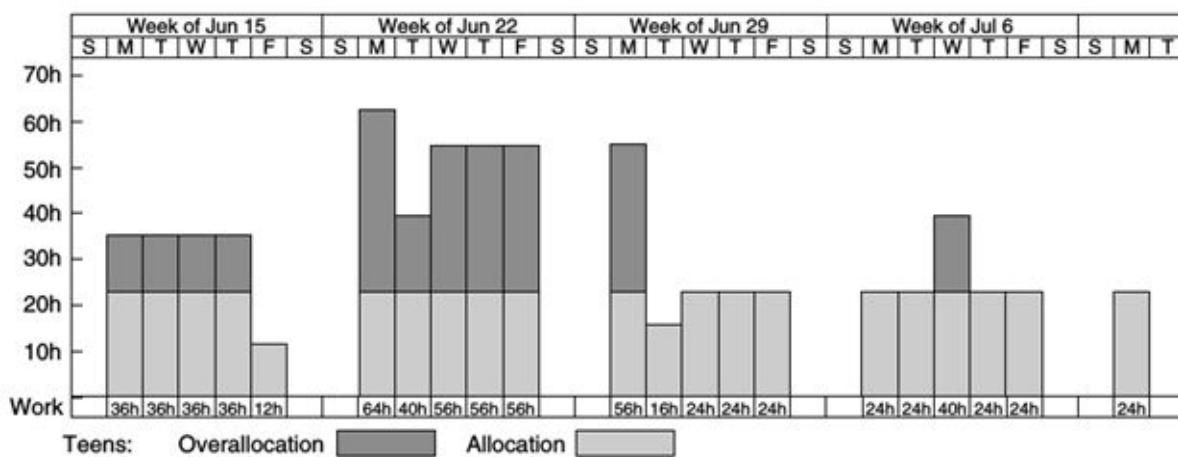
Оптимизация ресурсов

Итак, вы построили начальный план. Он потому и начальный, что он в принципе не подразумевает никакого распределения ресурсов. Начальный план показывает то, как задача была бы сделана в идеальном случае, но такого почти никогда не бывает. Поэтому этот план нужно оптимизировать в соответствии с ресурсами, которыми вы располагаете. Ресурсы -- это люди, оборудование, материалы и т.п.

Здесь есть два понятия: перегруженность (overallocation) и недозагруженность (underallocation). Overallocation означает, что вы пытаетесь перегрузить ваши ресурсы работой. Например, у вас есть один программист и вы ему нарисовали диаграмму, где он должен параллельно заниматься пятью задачами. Разумеется, он за один день не сделает 5 задач, длительность каждой из которых составляет один день. Он будет делать их 5 дней, и ваш график поедет. Overallocation -- это плохо ещё и потому, что когда вы перегружаете человека, он быстро устает и переутомляется, у него падает производительность.

Другая сторона проблемы -- это underallocation: ситуация, при которой есть люди, которые ничем не заняты. Понятно, что это тоже плохо, поскольку вы просто теряете деньги + это почти всегда оказывается негативно на большинстве людей. Чаще всего программисты работают не за зарплату. Программисты хотят решать интересные задачи, и в отсутствии достойной работы им реально становится скучно. Хорошие программисты долго на таких местах обычно не задерживаются.

Смотря на самую последнюю версию диаграммы Гантта, вы можете построить следующий график:



Видно, что в первый день вам нужно 36 человеко-часов, во второй -- 36 ч/ч, а в пятый -- 12 ч/ч. Человек в день должен работать по 8 часов, не более. В предыдущей таблице было 3 разработчика. Поэтому здесь allocation отмечено как 24 ч/ч -- сколько они в сумме могут отработать за один день. Это пример несбалансированного графика работ.

Глядя на такие диаграммы, вы понимаете, что у вас есть пики (когда вы пытаетесь сделать слишком много), их имеет смысл выравнивать. Чтобы этого добиться, вы просто пытаетесь двигать ваши задачи в соответствии с тем резервом, который у них есть. Если у некоторой задачи резерв равен 9, то вы ее можете двигать в пределах девяти единиц времени. Вы можете отложить ее на 9, но она все равно не

повлияет на конечный дедлайн. Т.е. резерв вам позволяет перемещать эти задачи по времени. В нашем примере, надо сдвигать задачи с первых четырёх дней на пятый. В этом примере это всё равно не позволит загрузить всех равномерно и сделать всё вовремя, но как первый шаг это вполне стоит того. Далее, если время является ключевым фактором, нужно искать возможность добавить людей на критические задачи или как-то иначе выровнять график. Но это тоже не всегда просто сделать: программисты -- не укладчики кирпичей, которых можно без проблем заменять одного на другого. В итоге получается жонглирование большим количеством параметров, к которому нужно подходить очень аккуратно.

Типичные ошибки при оценке проектов

Оценку делали не те люди. Мало опыта, непонимание техник оценивания.

Оценки должны делать нужные люди. Важно, чтобы это были те люди, которые хорошо знают предметную область. Независимо от того, какие методы используются, оценка всегда основывается на понимании работы, которую предстоит выполнить. Также к оценке стоит привлекать людей, которые будут фактически выполнять эту работу. Они имеют лучшее понимание своих способностей + если человек знает, что ему предстоит делать данную задачу, то он ответственнее подойдет к ее оценке. Но при этом, если у вас нет никакого опыта оценивания, то первую оценку, скорее всего, вы сделаете очень плохо. Со временем, если вы будете анализировать и использовать свой опыт, вы научитесь это делать более точно.

Слишком быстрый ответ и оценка в условиях недостаточной информации. Никогда не давайте быстрых ответов. Вы всегда что-нибудь забудете, всегда оцените более оптимистично, чем хотелось бы. Поэтому есть несколько стратегий, как с этим бороться.

- Каким-то образом откажитесь от ответа, но это не всегда оказывается возможным.
- Начните задавать вопросы, уточнять некоторые моменты. Когда вы получите больше информации, вы сможете точнее сделать оценку. Но, скорее всего, информации у того человека, который вас это спрашивает, нет. Поэтому после нескольких вопросов, на которые вам не смогли дать ответ, менеджер сам поймет, что слишком много неизвестных, чтобы дать сейчас точную оценку.
- Попробуйте оценить задачу максимально точно, а потом умножьте ее на два. А затем еще раз на два. Это тот коэффициент, который, скорее всего, будет похож на правду. Т.е. если вы про что-нибудь не знаете или забудете, то благодаря этому коэффициенту вы с большой долей вероятности все туда уложите.

Забыли про риски и прочие буферы. На одной из прошлых лекций мы говорили о том, что противодействие рискам стоит денег. Скорее всего, если вы умножите вашу изначальную оценку на 4, то этот бюджет на риски и прочие буферы туда просто войдут автоматически. Но если вы этого не делаете, то проведите переоценку рисков, продумайте, как реализация и интеграция этой задачи влияет на другие задачи и проект в целом.

Забыли про налоги. У вас есть диаграммы, вы посчитали количество человеко-часов, умножили их на зарплату. Допустим, что вы учли подоходный налог,

но при этом забыли про все остальные налоги -- половину бюджета вы просто выкинули. С человека государство берет 13%, но с компании гораздо больше -- до 54%. Если это не заложить в бюджет, вам придется это платить из собственного кармана.

Забыли про расходы на “административный аппарат”. Еще один пункт, который включается в общую сумму проекта. Вы считаете затраты по задачам, которые делают программисты. Вы учитываете их зарплаты, подсчитываете общий бюджет, свою зарплату вы тоже скорее всего посчитали. Но есть еще уборщица, водитель, отдел кадров, менеджеры над вами, бухгалтеры, ген. директор и т.д. Назовем их “административным аппаратом”, и им тоже нужно платить деньги. Поэтому вы должны добавить к общей сумме проекта еще некий процент на эти издержки. Если у вас компания маленькая (10-20 чел), то процент будет небольшой (5-10%). Если у вас большая компания (300-400 чел), то накрутка может доходить до 30-40%. И заказчики платят эти деньги, потому что крупная компания считается более надёжной, не совершают “детских ошибок” и т.п.

Забыли про отпуск. Вы берете ваш график и рассчитываете для каждого человека: 8ч в день => 40ч в неделю => x часов в год. Но потом оказывается, что работа у вас вовремя не заканчивается, поскольку 1/12 вы куда-то потеряли. Отпуск -- это то время, когда человек освобождается от работы для отдыха. Это нужно учитывать.

Забыли про индексацию зарплат. Уровень жизни с каждым годом растет, цены в магазинах растут, зарплаты, соответственно, тоже понемногу увеличиваются. Если вы планируете долгий проект, который длится несколько лет, то необходимо учитывать, что зарплаты людей не будут стоять на месте.

Забыли про закупки. Не учли деньги на приобретение разного рода товаров: компьютеры, столы, стулья, еда... Это, конечно, не такой большой бюджет, но про них не стоит забывать.

Политика vs здравый смысл. Очень часто на менеджера происходит давление со стороны заказчика или собственного руководства, которые пытаются заставить его ужать как-нибудь график. Например, просит сделать так, чтобы проект закончился на 2 месяца быстрее или стоил на 10% дешевле. Если вам нужно сделать на 10% дешевле проект, уменьшайте задачи. Никогда нельзя идти на поводу: “давайте вы мне сделаете проект на 10% дешевле, а я вам потом тоже что-нибудь хорошее сделаю” -- так не работает. Вы ведь не просто так придумали этот график и стоимость. Вы так сделали, потому что есть объективные реальности. У вас эта оценка обоснованная, поэтому эти 10% невозможно просто так выкинуть.

Уровни детальности оценки

Конечно, мы всегда хотим как можно более точных оценок, но точность стоит денег. Поэтому имеет смысл использовать различные методы оценки для различных точек принятия решения в проекте. Например, первоначальная оценка идеи проекта не должна занимать столько времени и усилий, сколько детальное планирование, необходимое для официального проекта.

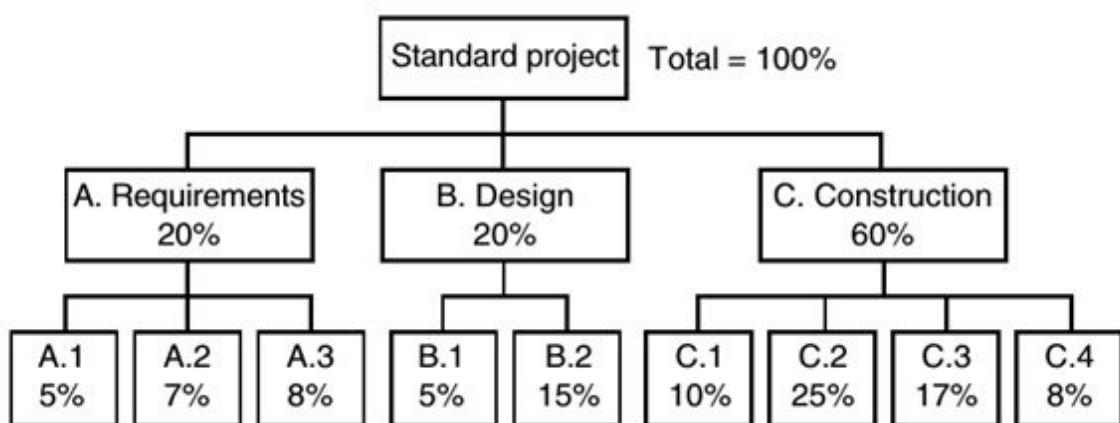
Есть три уровня детальности оценки.

- Неточная оценка, которую вы даете, особо не задумываясь. Она нужна на самом раннем уровне, когда вы просто пытаетесь понять, насколько это большой, масштабный и сложный проект.
- Второй уровень -- это когда вы несколько часов несколько посидели и подумали. Вы не делаете подробную декомпозицию, но вы уже собрали какую-то информацию, что-то порисовали на листочке. Обычно на основе этой второй оценки принимается решение: идти дальше или нет. Т.е. здесь вы уже прикидываете: оправдает ли результат затраченные средства? Осуществим ли проект технически?
- Если вы дважды сказали себе "да", значит, вы готовы продолжить и взяться за проект. Дальше делается детальная оценка. Она включает в себя всю информацию о графике и ресурсах. Это оценка будет использоваться для управления проектом и оценки его успеха. Чтобы сделать подробную декомпозицию большого проекта, вам может потребоваться несколько недель.

В зависимости от стадии проекта, необходимой степени точности, возможных расходов и трудозатрат применяются различные типы оценок стоимости.

Оценка "сверху-вниз"

Метод оценки "сверху вниз" используется для определения затрат на ранних стадиях проекта, когда информации о проекте еще очень мало. Смысл такой оценки в том, что она производится обобщенно и проект оценивается в целом. Сначала дается укрупненная оценка всего пакета работ, а затем она детализируется и декомпозируется на отдельные элементы, которые, в свою очередь, также оцениваются.

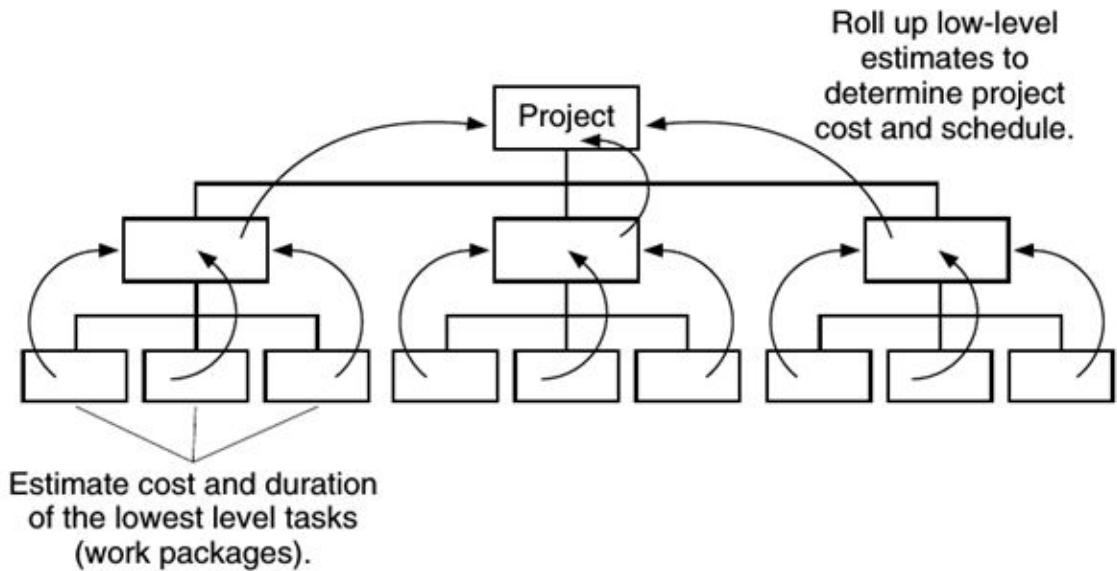


Данная оценка удобна тем, что не требует больших усилий и времени. Недостатком же является не такая высокая точность, какая могла бы быть при более детальной оценке. Зато мы можем быстро получить высокоуронневую картину проекта, и, например, не доводить декомпозицию до самого конца, если уже на промежуточном шаге понимаем, что сроки или бюджет не особо реалистичные.

Оценка "снизу-вверх"

Метод оценки "снизу-вверх" нужен для выработки согласованной базовой цены проекта или окончательной стоимостной оценки проекта. Название метода отражает

способ расчета стоимостной оценки -- данный подход предусматривает оценку затрат на детальных уровнях проекта с последующим суммированием элементов на более высоких уровнях.



Преимущество этого метода состоит в хорошей точности получаемых результатов, которая в свою очередь зависит от уровня детализации при оценке затрат на низких уровнях рассмотрения. Тогда возникает вопрос: если оценка “снизу-вверх” настолько точна, почему она не всегда используется? Все дело в том, что на ранних стадиях проекта просто недостаточно информации для детальной оценки “снизу-вверх” для жизненного цикла проекта. Ну и полноценная декомпозиция -- весьма длительный и затратный процесс.

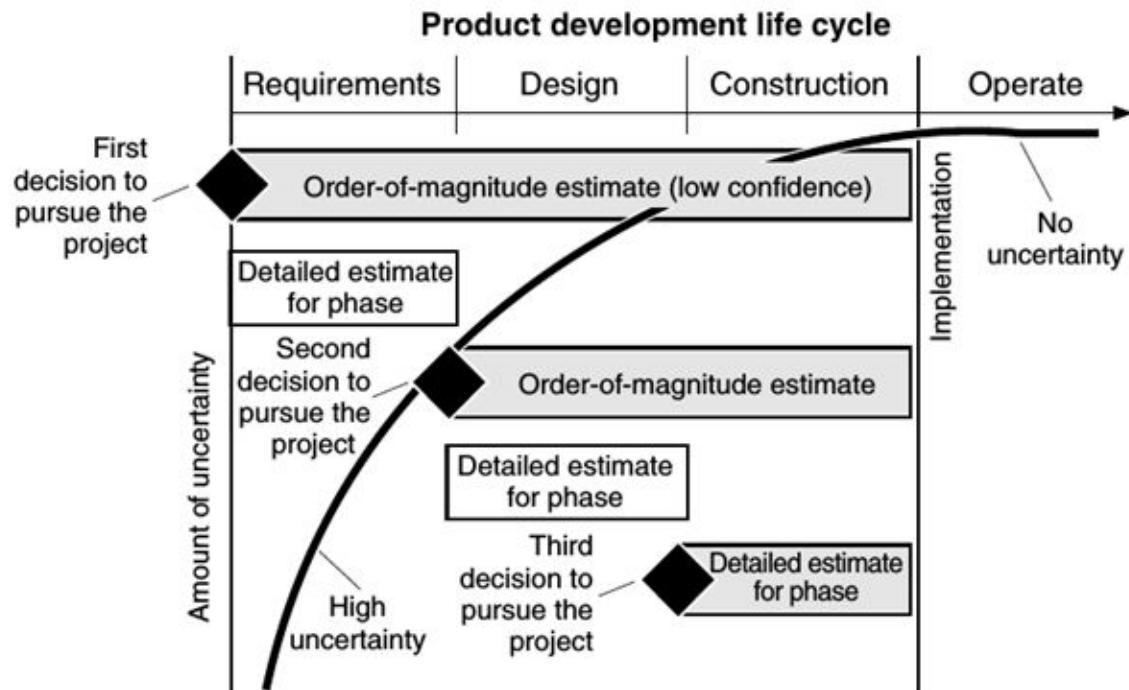
Разбиение проекта на фазы

Поэтапная оценка является одной из самых удобных для менеджеров проектов оценок, поскольку она требует вычисления затрат и обязательств по графику для каждой фазы проекта по отдельности. На начальной стадии любого проекта вы не располагаете достаточным количеством информации и поэтому нецелесообразно требовать полную оценку всего проекта в целом. Вместо этого проект разбивается на фазы, каждая из которых рассматривается как отдельный проект. Каждая фаза четко определена, поэтому проектная команда твердо знает, что ей нужно сделать и какие критерии будут использоваться для принятия решения о переходе к следующему этапу разработки.

Этот процесс выглядит следующим образом:

- Первая фаза начинается с приблизительной оценки полного жизненного цикла проекта в сочетании с подробной оценкой первой фазы. Эта подробная оценка считается обязательством проектной команды. По завершении этой фазы обычно осуществляется анализ как полученных результатов, чтобы с минимальными затратами определить и исправить ошибки, так и общего хода исполнения проекта, чтобы понять, стоит ли переходить к его следующей фазе.
- Вторая фаза начинается лишь после того, как будут одобрены результаты предыдущей. На этой фазе разрабатывается новая приблизительная оценка

для оставшейся части проекта вместе с подробной оценкой второй фазы. Теперь приближенная оценка становится более точной по сравнению с предыдущей, поскольку многое было изучено на первой стадии разработки. Затем этот цикл повторяется снова и снова.



Разбиение проекта на фазы позволяет представить его в виде списка основных результатов и дат, к которым эти результаты должны быть получены. Руководитель проекта осуществляет непосредственный контроль исполнения каждой задачи внутри проекта, сообщая вышестоящему менеджеру о достижении фазовых результатов.

Стоит отметить, что люди, которые это все финансируют, не всегда понимают такое поведение. Заказчикам кажется, что вы ведете себя непрофессионально, так сказать, "сбрасываете с себя ответственность". Им хочется, чтобы вы назвали конкретные сроки проекта и стоимость. И таких людей легко понять: заказчики ведь пытаются оценить перспективы инвестиционного предложения. А в случае разбиение проекта на фазы получается, что заказчик изначально не знает, каков окажется капитал вложения. Он располагает информацией только на первый этап, а на все остальное знает лишь приблизительную оценку, которая будет постоянно меняться.

Однако на самом деле, этой группе заказчиков необходимо понимать, что поэтапная оценка -- это метод снижения риска, который работает также и в их пользу. Если проектная группа должна взять на себя обязательства по оценке стоимости и графика на весь жизненный цикл разработки продукта прежде, чем у нее будет достаточно информации о продукте, это поставит всех под угрозу. Это связано с тем, что на ранней стадии проекта, когда еще информации чрезвычайно мало, команда с большой долей вероятности сделает плохую оценку. Поэтому есть риск, что расходы окажутся настолько большими, что компания просто обанкротится, а заказчик потеряет все свои деньги.

Таким образом, преимуществом поэтапного подхода оценивания является то, что он позволяет проекту быть управляемым множеством небольших обоснованных решений, а не одним большим, преждевременным решением.

Планирование денежных потоков

Одной из главных проблем любого бизнеса является правильное планирование денежных потоков. Даже рентабельные предприятия терпят банкротство из-за нехватки денежных поступлений. Нельзя только по уровню прибыли судить о мере финансовой устойчивости компании.

Управление денежными потоками -- это одна из наиболее важных задач финансового менеджмента. Для обеспечения платежеспособности компании и выполнения всех финансовых обязательств необходимо рациональное распределение и управление денежными потоками в организации. Необходимо спланировать синхронность поступления и расходования денежных средств и таким образом поддержать текущую платежеспособность предприятия.

Resource spreadsheet																									
ID	Resource Name	Week of Jun 15			Week of Jun 22			Week of Jun 29			Week of Jul 6			Week of Jul 13			Week of Jul 20			Week of Jul 27					
		S	M	T	W	T	F	S	S	M	T	W	F	S	S	M	T	W	F	S	S	M	T	W	F
1	Homeowner	4	4	4	4	4		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
2	Teens	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24	24
3	Contractor	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
4	Youth group	40	40	40	40	40		8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
5	Rototiller							8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8

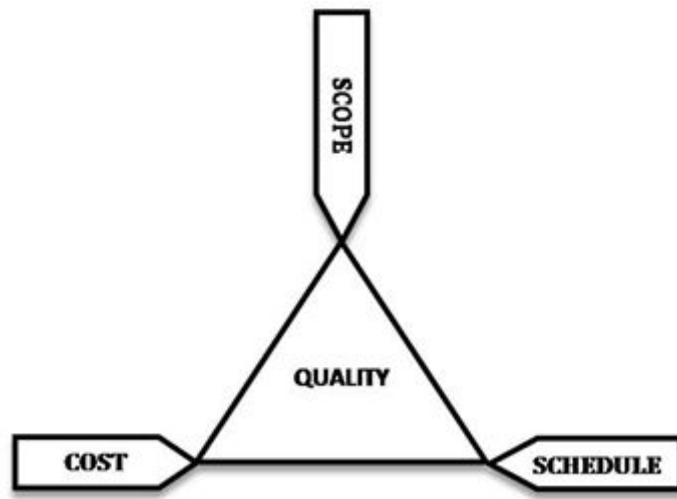
Cash flow schedule						
ID	Resource Name	Week 1	Week 2	Week 3	Week 4	Week 5
1	Labor	\$3,400	\$1,360	\$1,440	\$1,040	\$1,040
2	Contractor	\$0	\$0	\$0	\$2,200	\$0
3	Equipment	\$0	\$120	\$40	\$0	\$0
4	Materials	\$0	\$1,000	\$2,000	\$0	\$0
5	Weekly total	\$3,400	\$2,480	\$3,480	\$3,240	\$1,040
Hours per day						

При планировании денежных потоков вы определяете, когда и сколько денег поступит или будет уплачено по счетам, чтобы обеспечить нормальную деятельность предприятия. При этом необходимо учесть возможный временной сдвиг между реальным заключением договора и фактическим получением денег. В начале проекта вы скорее всего получите лишь небольшой аванс, а остальные деньги будете получать частями по достижении заранее оговоренных ключевых точек, либо, в худшем случае, в самом конце, когда проект будет завершён. Ну и надо понимать, что банковский перевод от заказчика к вам и от вас к сотрудникам тоже занимает время, а люди привыкли получать зарплату примерно в одно и то же время, и могут расстроиться, если будут осуществляться задержки.

Треугольник равновесия

В каждом проекте есть три основных момента:

- стоимость
- ресурсы (люди, компьютеры, время...)
- функциональность



Если вы измените одну из этих переменных, то это неизбежно отразится и на оставшихся переменных тоже. Действительно, если вы увеличите функциональность вашего проекта (добавите новые фичи), то вы не сможете это сделать с той же стоимостью за то же самое время. Это очевидно, поскольку у вас работы больше стало. Если у вас становится меньше денег или людей, то это, разумеется, скажется на качестве продукта, вы не реализуете то же самое количество функциональности за то же самое время. Точно так же, чтобы обеспечить такое же качество продукта за более короткий период времени, придется увеличить стоимость. Задача менеджера проекта, состоит в том, чтобы сбалансировать эти переменные, чтобы создать оптимальное равновесие между ценой и качеством.

Есть еще одна интересная картинка:

Памятка заказчику



Она показывает то же самое: не бывает быстро, качественно и дешево. Если вы делаете быстро и дешево -- получается криво; дешево и качественно -- долго; быстро и качественно -- дорого и т.д.