



Урок 6-6. Создание схемы функционирования проекта. Регулярное исполнение и Agile

2021

Организация спланированного проекта



Сводится к:

1. Формированию команды проекта
2. Максимально эффективным постановкам задач
3. Координации работающих внутри задач и над разными задачами

Отличается от процесса тем, что:

1. Команда собирается под проект, а не постоянно.
2. Задачи решаются не регулярные, а разные.



Постановка задач

Формулировка



Постановка включает **формулировку, донесение задачи до ответственного и исполнителей и принятие задачи оными.**

Формулировка должна содержать в себе ответы на вопросы:

1. Что нужно сделать? (предельно ТОЧНО и без воды какой нужно получить результат и с какими требованиями)
 - a. «Идите копайте» – плохо,
 - b. «Копать огород» – очень плохо
 - c. «Вскопать весь огород на полштыка и разрыхлить перекопанное» – хорошо
2. Кто выполняет задачу (исполнители).
3. Кто отвечает за результат (ответственный).
 - a. Если ответственного (старшего) среди исполнителей нет, а только исполнители, то ответственный – постановщик задачи.
4. Когда должна быть сделана задача
 - a. «Вскопать весь огород на полштыка и разрыхлить перекопанное сегодня к вечеру» – плохо
 - b. «Вскопать весь огород на полштыка и разрыхлить перекопанное сегодня к 19:00» – очень хорошо



Правильная постановка задачи – 50% ее успешного выполнения.



Постановка задач

Донесение

Донесение – передача формулировки исполнителям так, чтобы они ее увидели/услышали и поняли. Письменная постановка НАМНОГО лучше, т.к. исключает искажение смысла. В устной вы всегда получите «ну я вот так вот понял».

1. Голосом – будет работать иногда.
 - а. Всегда при устной постановке нужно ПЕРЕСПРОСИТЬ «Что и к какому сроку нужно сделать» и добиться того, чтобы исполнители точно формулировали результат.
2. Эл. почта – нормально, но не гарантирует, что задачу увидят
3. Месенджеры – хорошо
4. Специальный софт (таск-менеджеры) – отлично.



Постановка задач

Принятие



В случае, когда **руководитель проекта ставит задачу участнику проекта, обычно принятие происходит автоматически** В случае, когда руководитель проекта ставит задачу участнику проекта, обычно принятие происходит автоматически (в рамках проектного подчинения).

В случае, когда **задача в проект поступает извне и прямое подчинение отсутствует, нужно убедиться, что задача принята** (аналогично постановке задач вне SLA между подразделениями или вне административного подчинения).

В случае постановки в таск-менеджере принятие проконтролирует статусом сервис.

В случае **постановки устно или через обычные каналы коммуникации - нужно добиться того, чтобы принимающая сторона подтвердила принятие**. Без этого ожидать результат достаточно бессмысленно.





Постановка задач

Сроки

С целью минимизации срока выполнения задачи (проекта) сроки необходимо ставить по следующему правилу:
Реальный срок -10-15% времени. Исходим из того, что почти всегда срок пробьют на 10-15% 😊

Если выжимать сроки жестче – то исполнитель будет понимать их нереалистичность -> а значит будет заведомо не верить в то, что уложится в срок -> что приведет к тому, что ожидание (вероятность на результат) будет ОЧЕНЬ НИЗКИМИ, т.е. желания делать проект к сроку у исполнителя не будет. А значит велик риск вообще не получить результат.

Если ставить свободные сроки – срок будет пробит на 10-15% - т.е. результат получим с большим опозданием.

Исполнители понимают, что сроки бываютдвигаемые (устанавливает постановщик без привязки к внешнему событию) и недвигаемые (готовим проект к выставке, готовим софт к старту потока и т.д.). Вторые практически не пробиваются и исполнители отдадут все силы, чтобы уложится в них.

Всегда старайтесь привязать срок к какому-то внешнему недвигаемому событию – это снимает с вас негатив по ожидаемым переработкам:

- Приедет самый главный акционер проверять из Лондона.
- Заказчику нужно к концерту 25 мая.
- Нужно сдать отчет в налоговую до 30 апреля.



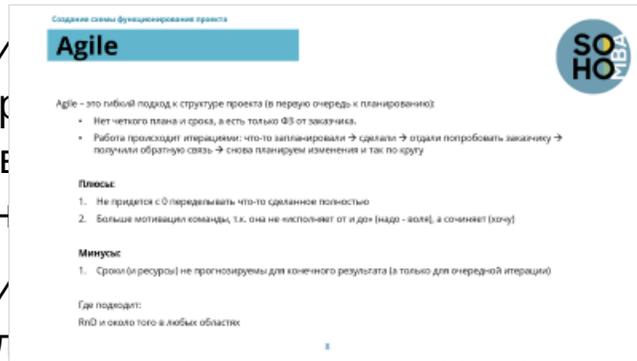
Распределение ролей и подчинение

В проекте есть **формальные** роли:

1. Руководитель проекта – назначается владельцем проекта
2. Команда проект (исполнители) – люди, которые участвуют в проекте и подчиняются в его рамках руководителю
3. Аудиторы проекта – наблюдатели за проектом

и **функциональные** роли (каменщик, столяр, программист).

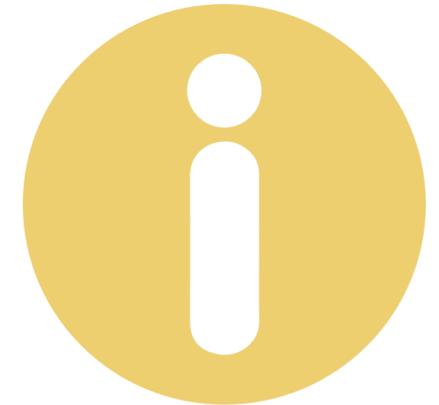
Подчи
(подпр
постав
админ
Напри
монол



команды строится руководителем в рамках задач
ответственный -> исполнители. В рамках
ставит подзадачи исполнителям

дали команду каменщиков и команду
«застелить фундамент к пятнице» на обе команды и

назначая одного из монолитчиков – главным – он выдает административное
руководство этому сотруднику над всеми остальными членами обеих команд.





Координация в рамках проекта или задачи

Координация – это обеспечение информационной прозрачности и функционального взаимодействия исполнителей:

- В рамках 1 задачи
- По связанным задачам

В процессах достигается регулярными SLA, в проектах – встречами:

1. Во время постановки задачи
2. Синхронизирующими

Таковые встречи по мере необходимости организует руководитель проекта (ответственный по задаче).

Повестка такой встречи обычно включает:

1. Доклад о состоянии задач к текущему моменту (если эта информация не прозрачна для всех) и проблемах в реализации.
2. Обозначение краткосрочных планов.
3. Моментальная выработка схем взаимодействия и обозначение индивидуальных обсуждений для решений конкретных вопросов.

Суммарно такие координационные совещания не должны занимать более 2 часов в неделю, дальше все не будут воспринимать информацию. Лучше 1 час в неделю.

Agile



Agile – это гибкий подход к структуре проекта (в первую очередь к планированию):

- Нет четкого плана и срока, а есть только ФЗ от заказчика.
- Работа происходит итерациями: что-то запланировали → сделали → отдали попробовать заказчику → получили обратную связь → снова планируем изменения и так по кругу

Плюсы:

1. Не придется с 0 переделывать что-то сделанное полностью
2. Больше мотивации команды, т.к. она не «исполняет от и до» (надо - воля), а сочиняет (хочу)

Минусы:

1. Сроки (и ресурсы) не прогнозируемы для конечного результата (а только для очередной итерации)

Где подходит:

RnD и около того в любых областях



Спасибо за внимание!