Лекция 5 Организация проектной команды

**Организация проектной команды**

Каждый проект разработки ПО имеет свою организационную структуру, которая

определяет распределение ответственности и полномочий среди участников

проекта, а также обязанностей и отношений отчетности. Чем меньше проект,

тем больше ролей приходится совмещать одному исполнителю.

Роли и ответственности участников типового проекта разработки ПО можно

условно разделить на пять групп:

1. Анализ. Извлечение, документирование и сопровождение требований к

продукту.

2. Управление. Определение и управление производственными

процессами.

3. Производство. Проектирование и разработка ПО.

4. Тестирование. Тестирование ПО.

5. Обеспечение. Производство дополнительных продуктов и услуг.

Группа анализа включает в себя следующие роли:

 Бизнес-аналитик. Построение модели предметной области (онтологии).

 Бизнес-архитектор. Разрабатывает бизнес-концепцию системы.

Определяет общее видение продукта, его интерфейсы, поведение и

ограничения.

 Системный аналитик. Отвечает за перевод требований к продукту в

функциональные требования к ПО.

 Специалист по требованиям. Документирование и сопровождение

требований к продукту.

 Менеджер продукта (функциональный заказчик). Представляет в

проекте интересы пользователей продукта.

Группа управления состоит из следующих ролей:

 Руководитель проекта. Отвечает за достижение целей проекта при

заданных ограничениях (по срокам, бюджету и содержанию),

осуществляет операционное управление проектом и выделенными

ресурсами.

 Куратор проекта. Оценка планов и исполнения проекта. Выделение

ресурсов.

 Системный архитектор. Разработка технической концепции системы.

Принятие ключевых проектных решений относительно внутреннего

устройства программной системы и еѐ технических интерфейсов.

 Руководитель группы тестирования. Определение целей и стратегии

тестирования, управление тестированием.

 Ответственный за управление изменениями, конфигурациями, за сборку

и поставку программного продукта.

В производственную группу входят:

 Проектировщик. Проектирование компонентов и подсистем в

соответствие с общей архитектурой, разработка архитектурно значимых

модулей.

 Проектировщик базы данных.

 Проектировщик интерфейса пользователя.

 Разработчик. Проектирование, реализация и отладка отдельных

модулей системы.

В большом проекте может быть несколько производственных групп,

ответственных за отдельные подсистемы. Как правило, проектировщик

выполняет роль лидера группы и управляет своим подпроектом или пакетом

работ. Стоит не забывать, что руководитель проекта делегирует полномочия, но

не ответственность.

Группа тестирования в проекте состоит из следующих ролей:

 Проектировщик тестов. Разработка тестовых сценариев.

 Разработчик автоматизированных тестов.

 Тестировщик. Тестирование продукта. Анализ и документирование

результатов.

Участники группы обеспечения, как правило, не входят в команду проекта. Они

выполняют работы в рамках своей процессной деятельности. К группе

обеспечения можно отнести следующие проектные роли:

 Технический писатель.

 Переводчик.

 Дизайнер графического интерфейса.

 Разработчик учебных курсов, тренер.

 Участник рецензирования.

 Продажи и маркетинг.

 Системный администратор.

 Технолог.

 Специалист по инструментальным средствам.

 Другие.

В зависимости от масштаба проекта одну роль могут исполнять несколько

человек. Например, разработчики, тестировщики, технические писатели.

Некоторые роли всегда должен исполнять только один человек. Например,

Руководитель проекта, Системный архитектор. Один человек может исполнять

несколько ролей. Возможны следующие совмещения ролей:

 Руководитель проекта + системный аналитик (+ системный архитектор)

 Системный архитектор + разработчик

 Системный аналитик + проектировщик тестов (+ технический писатель)

 Системный аналитик + проектировщик интерфейса пользователя

 Ответственный за управление конфигурациями + ответственный за

сборку и поставку (+ разработчик)

Крайне нежелательно совмещать следующие роли:

 Разработчик + руководитель проекта

 Разработчик + системный аналитик.

 Разработчик + проектировщик интерфейсов пользователя.

 Разработчик + тестировщик

Не раз приходилось наблюдать, как в критические периоды проекта его

менеджер-разработчик с увлечением правит очередные баги, а проектная

команда в полном составе стоит у него за спиной и наблюдает за этим

процессом. Это плохой пример руководства проектом.

Программисты любят и умеют программировать. Пусть они этим и занимаются.

Не стоит загружать программистов несвойственной для них работой. В каждом

проекте разработки программного продукта много других работ: бизнес-анализ,

проектирование эргономики, графический дизайн, разработка пользовательской

документации. Эти работы с программированием не имеют ничего общего. Для

них требуются совершенно другая квалификация и другой склад мышления.

При кустарном производстве программ эти задачи, как правило, поручаются

программистам, которые это делать не умеют и не любят. Получается обычно

плохо, да еще и дорого. В силу своей интроверсии, граничащей с аутизмом,

программист просто не в состоянии увидеть свою программу чужими глазами –

глазами пользователей. Никто уже не хочет работать с программами с

технологической парадигмой навороченного пользовательского интерфейса -

кустарным творением программистов - когда для того чтобы работать с

системой, надо обязательно знать, как она устроена. Это типичное творение

программиста, которому гораздо важнее видеть, как работает его программа,

чем разбираться в том, что она делает для пользователя. Поэтому, необходимо

привлекать в проектную команду бизнес-аналитиков, эргономистов, художников-

дизайнеров, документалистов. Разделение труда и специализация - залог

перехода от кустарного производства к более эффективному промышленному

производству.

Из профессиональных программистов получаются отличные тестировщики.

Лучшая команда тестирования, которую я встречал, была в Luxoft. Это были

маститые программисты из одного академического НИИ с опытом 20-30 лет.

Они не осваивали новые программистские технологии, но исключительно

эффективно ломали то, что было сделано на их основе. Однако, совмещать

одновременно роли программиста и тестировщика – плохая практика. Хороший

программист убежден, что он пишет программы правильно и ему

психологически тяжело допустить, что где-то в его коде может быть ошибка. А

ошибки есть всегда!

Организационная структура проекта обязательно должна включать в себя

эффективную систему отчетности, оценки хода выполнения проекта и систему

принятия решений. Можно рекомендовать еженедельные собрания по статусу

проекта, на которых анализируются риски, оцениваются результаты,

достигнутые на предыдущей неделе, и уточняются задачи на новый период.

В модели Scrum рекомендуются ежедневные совещания по состоянию работ –

«Stand Up Meeting», но мне кажется, что это применимо, скорее, для небольших

рабочих групп от 3 до 5 разработчиков. Хотя в критические периоды проекта,

приходилось проводить и ежедневные совещания.

Важно помнить, что организационная структура проекта – «живой» организм.

Она начинает складываться на стадии планирования и может меняться в ходе

проекта. Нестабильность организационной структуры (частые замены

исполнителей) – серьезная проблема в управлении сложными программными

проектами, поскольку существует время вхождения в контекст проекта, которое

может измеряться месяцами.

**Жизненный цикл проекта. Фазы и продукты**

Ранее уже отмечалось, что каждый программный продукт имеет свой

жизненный цикл, в который проект разработки очередного релиза входит как

одна из фаз. Аналогично, каждый проект разработки ПО имеет свой

собственный жизненный цикл, который состоит из четырех фаз (Рисунок 5.1).

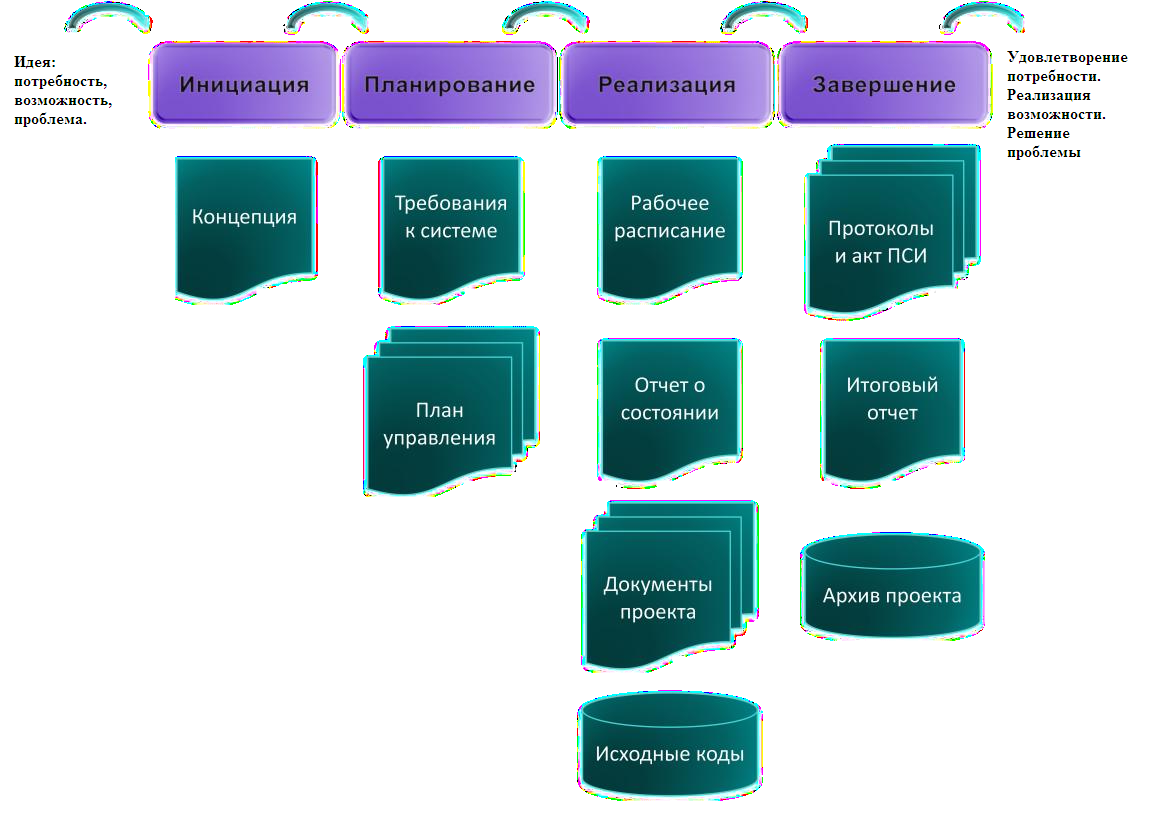


Рисунок 5.1. Жизненный цикл и основные продукты программного проекта

На фазе инициации проекта необходимо понять, что и зачем мы будем делать

– разработать концепцию проекта. Фаза планирования определяет, как мы

будем это делать. На фазе реализации происходит материализация наших

идей в виде документированного и протестированного программного продукта.

И, наконец, на фазе завершения мы должны подтвердить, что мы разработали

именно тот продукт, который задумали в концепции проекта, а также провести

приемо-сдаточные испытания (ПСИ) продукта на предмет соответствия его

свойств, определенным ранее требованиям.

Как правило, редкий проект выполняется в соответствие с первоначальными

планами, поэтому важным элементом фазы завершения является «обратная

39

связь»: анализ причин расхождения и усвоение уроков на будущее. Помним, что

управляющая система без обратной связи не может быть устойчивой.

Более подробно о каждой фазе проекта и их продуктах будет рассказано в

последующих лекциях.

Завершая обзор управления проектами «с высоты птичьего полета»,

необходимо упомянуть еще об одной особенности проекта по сравнению с

операционной деятельностью. Если в операционной деятельности ресурсы

расходуются более-менее равномерно по времени, то в проектном управлении

расходование ресурсов в единицу времени имеет явно выраженное

колоколообразное распределение (Рисунок 5.2)

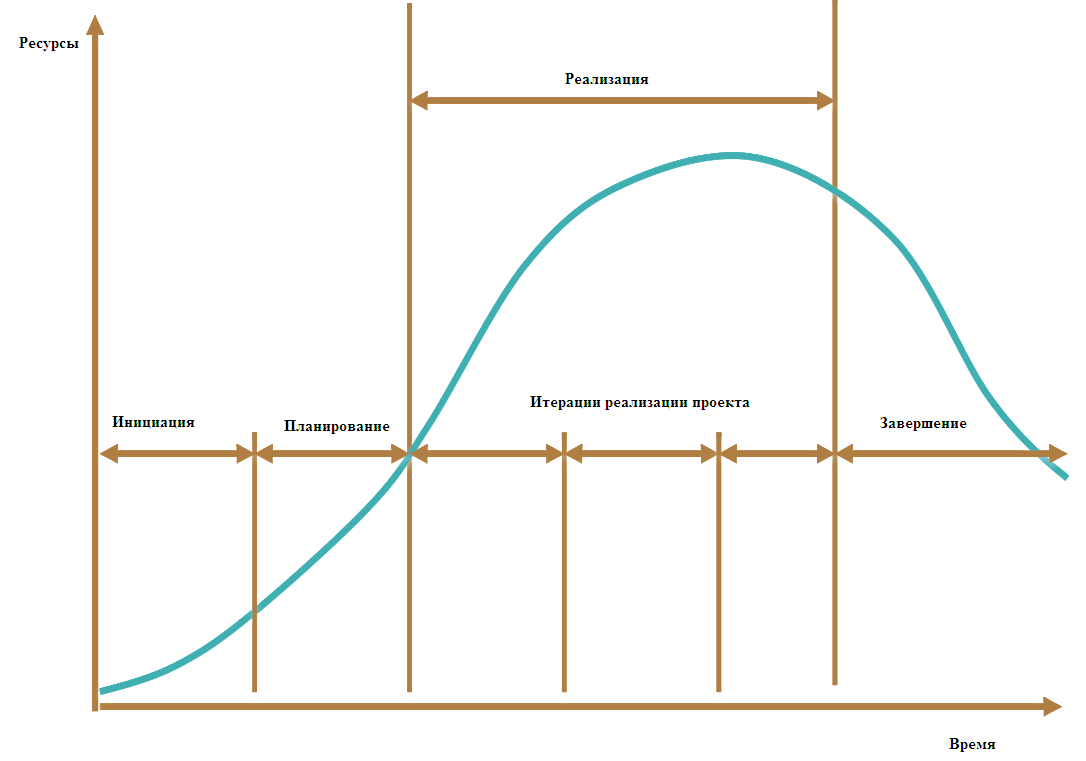


Рисунок 5.2. Распределение ресурсов по фазам проекта

Проект часто начинается с идеи, которая появляется у одного человека.

Постепенно, по мере формулирования, анализа и оценки этой идеи,

привлекаются дополнительные специалисты. Еще больше участников

требуется на фазе планирования проекта. Пик потребления ресурсов

приходится на фазу реализации.

В современных моделях разработки ПО реализация осуществляется на основе

сочетания итеративного и инкрементального подходов.

Итеративность предполагает, что требования к системе и ее архитектура

прорабатываются не один раз, а постепенно уточняются от итерации к

итерации. Это означает, что на каждой итерации происходит полный цикл

процессов разработки: уточнение требований, проектирование, кодирование,

тестирование и документирование.

Инкрементальность состоит в том, что результатом каждой итерации является

версия ПО, которая реализует часть функциональности будущего программного

продукта и может быть введена в тестовую или опытную эксплуатацию, а также

оценена заказчиком и будущими пользователями. Это означает, что после

каждой итерации происходит прирост требуемого функционала, а

нереализованных функций будущего продукта остается все меньше.

Сочетание итеративности и инкрементальности обеспечивает эффективность

разработки и существенное снижение рисков по ходу проекта. Об этом мы еще

будем говорить.

На последней фазе происходит постепенное высвобождение участников

проектной команды. Следует помнить, что проект должен иметь четкое

окончание во времени, после которого все работы по проекту закрываются, и на

проект перестают тратиться ресурсы. Не должно оставаться «зависших» работ.

**Выводы**

Проект - этот средство стратегического развития. Цель – описание того, что мы

хотим достичь. Стратегия – констатация того, каким образом мы собираемся эти

цели достигать. Проекты преобразуют стратегии в действия, а цели в

реальность.

Участников типового проекта разработки ПО можно условно разделить на пять

групп ролей:

1. Анализ. Извлечение, документирование и сопровождение требований к

продукту.

2. Управление. Определение и управление производственными

процессами.

3. Производство. Проектирование и разработка ПО.

4. Тестирование. Тестирование ПО.

5. Обеспечение. Производство дополнительных продуктов и услуг.

У программного проекта имеется четыре фактора, которые определяют его

успешность:

1. Выполнен в соответствие со спецификациями.

2. Выполнен в срок.

3. Выполнен в пределах бюджета.

4. Каждый участник команды уходил с работы в 18:00 с чувством успеха.

Задание:

1. Произвести анализ мирового рынка по разработке программного обеспечения для выработки собственного мнения.

2. Продолжить анализ лекции 2 [1] по рассмотренному материалу и подготовится к обсуждению.

Литература:

* 1. С.Архипенков. Лекции по управлению программными проектами. Москва, 2009, 127с.