Раздел 2. Компании и команды, реализующие программные проекты

2.1 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.

2.2 Группы ролей

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.

2.3 Группы анализа

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.

2.4 Группы управления

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 её технических интерфейсов.
- Руководитель группы тестирования. Определение целей и стратегии тестирования, управление тестированием.
- Ответственный за управление изменениями, конфигурациями, за сборку и поставку программного продукта.

2.5 Роль менеджера проекта

Менеджер проекта – это лицо, назначаемое исполняющей организацией ответственным за достижение целей проекта. Роль менеджера проекта отличается от роли функционального менеджера или операционного менеджера. Как правило, функциональный менеджер сосредоточен на обеспечении надзора за некоей зоной управления, а операционные менеджеры несут ответственность за определенное направление основной деятельности компании.

В зависимости от структуры организации менеджер проекта может подчиняться функциональному менеджеру. В других случаях менеджер проекта может быть одним из нескольких менеджеров проектов, подотчетных менеджеру портфеля или программы, который несет ответственность за проекты в масштабах предприятия. В структуре такого типа менеджер проекта тесно сотрудничает с менеджером портфеля или программы для достижения целей проекта и обеспечения соответствия плана проекта комплексному плану программы.

Многие инструменты и методы управления проектами специфичны для управления проектами. Тем не менее, понимание и применение знаний, инструментов и методов, признанных в качестве хорошей практики, недостаточно для эффективного управления проектами. В дополнение к специальным навыкам и знанию общего менеджмента, необходимым для проекта, эффективное управление проектами требует наличия у менеджера проекта следующих характеристик:

1. **Знания**. Это относится к тому, что менеджер знает об управлении проектами.

2. **Результативность**. Это относится к тому, что менеджер способен сделать или достичь, применяя свои знания об управлении проектами.

3. **Личные качества**. Это относится к тому, как менеджер проекта ведет себя во время выполнения проекта или связанной с ним деятельности. Личная эффективность охватывает установки, основные личностные

характеристики и лидерские качества – способность управлять командой проекта при достижении целей и уравновешивании ограничений проекта.

2.6 Производственная группа

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 под проектом или пакетом работ. Стоит не забывать, что руководитель проекта делегирует полномочия, но не ответственность.

2.7 Группа тестирования

Группа тестирования в проекте состоит из следующих ролей:

- Проектировщик тестов. Разработка тестовых сценариев.
- Разработчик автоматизированных тестов.
- Тестировщик. Тестирование продукта. Анализ и документирование рузультатов.

2.8 Группа обеспечения

В зависимости от масштаба проекта одну роль могут исполнять несколько человек. 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.

2.9 Совмещение ролей

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)

2.10 Нежелательное совмещение ролей

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.

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.

2.11 Что надо делать для успеха программного проекта

1. Ставим цели

1.1. Концепция определяет ясные недвусмысленные цели.

1.2. Все члены команды считают концепцию реалистичной.

1.3. У проекта имеется обоснование экономической эффективности.

1.4. Разработан прототип пользовательского интерфейса.

1.5. Разработана спецификация целевых функций программного продукта.

1.6. С конечными пользователями продукта налажена двухсторонняя связь

Определяем способ достижения целей

2.1. Имеется детальный письменный план разработки продукта.

2.2. В список задач проекта включены «второстепенные» задачи (управление конфигурациями, конвертация данных, интеграция с другими системами).

2.3. После каждой фазы проекта обновляется расписание и бюджет.

2.4. Архитектура и проектные решения документированы.

2.5. Имеется план обеспечения качества, определяющий тестирование и рецензирование.

2.6. Определен план многоэтапной поставки продукта.

2.7. В плане учтены обучение, выходные, отпуска, больничные.

2.8. План проекта и расписание одобрен всеми участниками команды.

Контролируем и управляем реализацией

3.1. У проекта есть куратор. Это такой топ-менеджер исполняющей компании, который лично заинтересован в успехе данного проекта.

3.2. У проекта есть менеджер, причем только один!

3.3. В плане проекта определены «бинарные» контрольные точки.

3.4. Все заинтересованные стороны могут получить необходимую информацию о ходе проекта.

3.5. Между руководством и разработчиками установлены доверительные отношения.

3.6. Установлена процедура управления изменениями в проекте.

3.7. Определены лица, ответственные за решение о принятии изменений в проекте.

3.8. План, расписание и статусная информация по проекту доступна каждому участнику.

3.9. Код системы проходит автоматическое рецензирование.

3.10. Применяется система управления дефектами.

2.12 Анализируем угрозы

4.1. Имеется список рисков проекта. Осуществляется его регулярный анализ и обновление.

4.2. Руководитель проекта отслеживает возникновение новых рисков.

4.3. Для каждого подрядчика определено лицо, ответственное за работу с ним.

Работаем над созданием команды

5.1. Опыт команды достаточен для выполнения проекта.

5.2. У команды достаточная компетенция в прикладной области.

5.3. В проекте имеется технический лидер.

5.4. Численность персонала достаточна.

5.5. У команды имеется достаточная сплоченность.

5.6. Все участники привержены проекту.

Оценка и интерпретация теста

Оценка: сумма баллов, каждый пункт оценивается от 0 до 3:

- 0 — даже не слышали об этом;
- 1 — слышали, но пока не применяем;
- 2 — применяется частично;
- 3 — применяется в полной мере.

Поправочные коэффициенты:

- для малых проектов (до 5 человек) — 1.5;
- для средних (от 5 до 20 человек) — 1.25.

Результат:

- <40 — завершение проекта сомнительно.
- 40-59 — средний результат. В ходе проекта следует ожидать серьезные проблемы.
- 60-79 — хороший результат. Проект, скорее всего, будет успешным.
- 80-89 — отличный результат. Вероятность успеха высока.
- >90 — великолепный результат. 100% шансов на успех.

Этот чек-лист перечисляет, *что* надо делать для успеха программного проекта, но не дает ответ на вопрос *как* это следует делать. Именно об этом пойдет речь в остальных лекциях.