

## Семинар 1. Управление разработкой ПО

### Задачи управления проектами

Это занятие посвящено управлению проектами по разработке, поддержке или модификации программного обеспечения. С достаточно общих позиций можно считать задачей управления проектами эффективное использование ресурсов для достижения нужных результатов. Всегда нужно получить как можно более хорошие результаты, используя при этом как можно меньше ресурсов. При этом в первую очередь возникают два вопроса: что именно считается «хорошим» результатом проекта и чем, какими ресурсами, можно пользоваться для его достижения.

Чтобы ответить на вопрос о том, какими критериями руководствуются при оценке проектов, и чего нужно добиваться, надо рассмотреть его с разных аспектов.

Одним из самых важных критериев является экономическая эффективность проекта, т.е. отношение суммы доходов разного рода, полученных в его результате, ко всем затраченным ресурсам. К сожалению, эти доходы чаще всего невозможно определить заранее. Поэтому при оценках проекта вместо дохода от его результатов рассматривают качество создаваемого ПО во всех его аспектах, т.е. набор имеющихся функций, надежность работы, производительность, удобство для всех категорий пользователей, а также удобство расширения и внесения изменений. Характеристики качества должно соотноситься с требованиями рынка или заказчика и с уже имеющимися продуктами.

С более общих позиций и экономические, и неэкономические показатели результативности проекта объединяют в понятие ценностей, создаваемых в его ходе. Эти ценности могут иметь различную природу в зависимости от проекта, от организации, в рамках которой он проводится, от национально-культурных особенностей рынка и персонала и пр. Кроме того, ценности выстраиваются в иерархию в зависимости от уровней рассмотрения проектов.

- В одном конкретном проекте основной ценностью может быть достижение запланированного качества результатов в указанный срок и в рамках определенного бюджета. В то же время, могут быть получены и другие ценности: достигнута высокая сплоченность команды; новые члены коллектива приобретут серьезные знания и полезные навыки; команда овладеет новыми технологиями; ее члены получают повышение и/или поощрения, которые повысят их лояльность компании и т.п.
- На уровне нескольких зависящих друг от друга проектов (такую группу проектов называют программой), в ходе которых создаются и дорабатываются несколько продуктов на единой платформе, а также могут оказываться различные услуги, связанные с этими продуктами, ценности связаны, прежде всего, с качеством общей архитектуры создаваемых продуктов. В одном проекте работа иногда ведется по принципу «сдали-и-забыли», т.е. основные усилия направлены на то, чтобы заказчик подписал акт приемки работ или его аналог, после чего поставщик перестает отвечать за результаты, поэтому часто такой аспект качества ПО, как удобство внесения изменений, игнорируется. Однако для бизнеса организации в целом проведение таких проектов небезопасно. Среди исследователей и экспертов-практиков

преобладает взгляд на любую программную систему как на систему развивающуюся, полезность которой значительно снижается, если нет возможности расширять ее, тем более — исправлять серьезные ошибки, которые всегда есть в сложных программах. Заказчик всегда сталкивается с проблемами поддержки ПО и рано или поздно столкнется и с необходимостью его развития. На уровне группы проектов игнорирование удобства модификации ПО, а также вопросов, связанных с организационными и экономическими последствиями изменений в общей архитектуре, просто губительно.

- На уровне организации в целом или подразделения, в рамках которого может одновременно проводиться много проектов, связанных по предметной области, используемым технологиям и просто по вовлеченным в них людям, возникают другие ценности. Это может быть отлаженность производственных процессов, высокая технологическая экспертиза и технологическое лидерство в своей области, низкая текучка кадров, повышение оборота, прибыли, капитализации, доли продаж в рамках отрасли, занимаемого среди поставщиков такой же продукции места по экономическим и технологическим показателям.

Поскольку каждый проект проводится в рамках какой-то организации, то принятая в ней система ценностей влияет и на оценку каждого конкретного проекта (см. далее). Основные виды ресурсов, используемых в любом проекте, следующие.

- **Время.**

Этот ресурс всегда жестко ограничен. Продолжительность проекта фиксирована, это одно из главных отличий проектов от обычной операционной деятельности, которой нужно заниматься неопределенной время. Чаще всего эти ограничения определяются интересами заказчика, выраженными в контракте, или решением руководства, основанном на анализе рынка и информации о действиях конкурентов.

Более того, даже при попытке создать «вялотекущий» проект без четкой установки его временных рамок (иногда при помощи такого приема руководство организации пытается выполнить нужные внутренние работы, не выделяя на них достаточных ресурсов), руководитель проекта должен настаивать на их определении. Иначе проекта как такового просто не получится — эффективное использование времени играет очень важную роль в успешности достижения необходимых результатов.

- **Бюджет.**

Бюджет проекта тоже всегда ограничен — деньги, как известно, лишними не бывают. Деньги часто рассматриваются как практически универсальный эквивалент других ресурсов: за счет вложения дополнительных денег пытаются выиграть во времени, привлечь дополнительный персонал и пр. Однако полностью в деньги можно перевести только используемое оборудование и материалы, да и то с некоторыми потерями во времени на их приобретение и подготовку к работе.

Вместе со временем бюджет задает основные ограничения на содержание и возможные результаты проекта.

- **Персонал.**

Персонал иногда рассматривается как возобновляемый ресурс, имеющий денежный эквивалент («найдем проектировщика за 1500 у.е. в месяц»). Однако чаще всего люди ведут себя не совсем так, как оборудование или мебель, — они не позволяют себя «передвигать», «убирать» и «добавлять» с такой же легкостью. Имея определенный персонал, нельзя получить нужный результат с помощью заранее известной последовательности действий. Даже для получения одних и тех же результатов от одного и того же человека в разных обстоятельствах требуется применять различные подходы. Обычно лишь работников среднего уровня можно найти в нужном количестве за среднюю зарплату, а чтобы нанять высокопрофессионального ответственного члена проектной команды (руководителя, бизнес-аналитика, архитектора, специалиста по проектированию интерфейсов и пр.), часто нужно пойти на достаточно высокие затраты. Или же придется идти на аренду такого специалиста, что стоит еще дороже. С другой стороны, человека, который со временем вырастет до такого уровня, можно нанять практически случайно. Большое значение для успеха проекта также имеет построение из отдельных людей настоящей команды, которая дала бы больший результат, чем сумма результатов отдельной работы ее членов. Зачастую действительно сложное ПО без такой команды в рамках выделенного бюджета создать невозможно.

- **Используемое оборудование, инструменты, материалы, и т.п.**

Это ресурсы классического типа, которые могут использоваться однократно или многократно, могут изнашиваться или не изнашиваться в результате использования (а также просто со временем) и достаточно адекватно могут быть обменены на деньги.

Дальнейшее содержание занятия посвящено общим вопросам управления проектами и отдельным деятельности, которым менеджер проекта должен уделять внимание в его ходе. Рассматриваемые подходы в основном применимы к проектам в любой области, а при управлении разработкой ПО нужно учитывать следующие ее особенности.

- Создаваемые программы нематериальны. Это порождает проблемы двух видов.
  - Программы обладают потрясающей гибкостью, они не оказывают сопротивления воздействиям, как физические материалы. Все знают, что построить дом можно из твердого и плотного материала, а чтобы сделать батут нужно использовать материал мягкий, гибкий и прочный. В мире же программ можно построить все, что угодно, из одних и тех же базовых конструкций.
  - Поэтому иногда кажется, что раз суть требуемых изменений в программе понятна, на их воплощение нужно немного усилий. Это не так. Работа с элементами программ в этом аспекте не слишком отличается от работы с кирпичами и строительными блоками. А если эти блоки еще и стоят кое-как, то при попытке передвинуть их программиста вообще может «завалить» — отладка полученной программы потребует колоссальных усилий.
  - Движение к нужному результату при разработке ПО очень тяжело проконтролировать. При возведении здания или постройке корабля можно непосредственно наблюдать за тем, как продвигается работа. При создании сложной программной системы силами многих разработчиков нужно аккуратно подбирать индикаторы того, как идут дела, иначе легко впасть в заблуждение относительно истинного положения вещей.

- Программные системы практически всегда уникальны.

Каждая из них обладает своим набором характеристик (включая все реализуемые функции, производительность при их выполнении, все элементы пользовательского интерфейса и т.п.), так или иначе отличающихся от характеристик других программ, даже делающих «то же самое». Если обладающая нужными свойствами (в том числе и подходящей ценой) программа уже имеется, незачем создавать ее заново — достаточно приобрести ее или взять ее код и скомпилировать. Поэтому практически каждая *разрабатываемая* программа уникальна — она должна иметь такие характеристики, которыми не обладает ни одна уже созданная.

- Тем самым, почти каждый проект по разработке ПО включает элементы творчества, создания того, чего еще никто не делал. Крупные же проекты требуют решения сразу нескольких ранее не решенных задач. Управление проектами с элементами творческой деятельности очень сильно отличается от управления проектами, в которых заранее ясно, что именно надо делать и как.
- Другое следствие этой уникальности ПО — отсутствие стандартных процессов разработки. Нет целостных подходов к созданию ПО, которые годились бы для всех случаев, а не только для определенного вида проектов. Кроме того, для хорошо определенных процессов, таких, как RUP, XP, Microsoft Solution Framework или DSDM (Dynamic Systems Development Method, Метод разработки динамичных систем), недостаточно четко очерчены области их применимости. Каждый раз менеджеру проекта приходится только на основании своего опыта и советов экспертов принимать решение о том, какой процесс разработки использовать и как его модифицировать для достижения большей эффективности в конкретном проекте.
- Есть много аргументов в пользу того, что программный код является проектом, а не конечным продуктом.

При разработке ПО переход от проекта к продукту почти полностью автоматизирован — требуется лишь скомпилировать код и развернуть систему в том окружении, где она будет работать. А само программирование гораздо больше напоминает разработку проекта здания, чем его строительство по уже готовому проекту. То же, что в разработке ПО обычно называется проектом или дизайном, представляет собой лишь набросок окончательного проекта, определяющий основные его черты и требующий дальнейшей детализации. Таким образом, разработка программ отличается от других инженерных видов деятельности тем, что в основном состоит из проектирования, а не изготовления продукта.

Это еще одна причина того, что программирование всегда включает элемент творчества. Кроме того, проблемы, с которыми сталкивается руководитель проекта разработки ПО, гораздо более похожи на проблемы периода проектирования здания, самолета или корабля, чем на проблемы периода их постройки.

### **Окружение проекта**

Среди аспектов окружения проекта, оказывающих на его ход существенное влияние, отметим структуру проводящей проект организации, организационную культуру различных

вовлеченных в проект организаций, которую руководителю проекта надо учитывать при выработке стратегии поведения, а также заинтересованных в проекте лиц.

### **Структура организации-исполнителя проекта**

Полномочия руководителя и ход проекта в значительной мере зависят от структуры организации, в рамках которой проводится проект, т.е. от тех правил, согласно которым в этой организации группируются ресурсы и происходит выделение ресурсов под проекты. Различают следующие структуры организаций.

- **Функциональная.**

В такой организации подразделения выделяются по их области деятельности или этапам производственных процессов — в ней есть финансовый, плановый, маркетинговый, опытно-конструкторский и производственный отделы. Проекты ведутся сотрудниками нескольких разных подразделений, а руководство проектом осуществляется за счет координации их деятельности, через руководителей соответствующих отделов. Руководитель проекта практически всегда член дирекции. Выделение ресурсов, необходимых проекту, должно осуществляться на уровне дирекции, которая дает поручения руководству отделов выделить соответствующую часть ресурсов. Такая схема позволяет собрать вместе сотрудников, обладающих знаниями и умениями в одной области, и развивать их экспертизу. Она помогает выполнять очень крупные проекты. С другой стороны, она не слишком гибка и предполагает высокую косвенность управления проектом и ограниченность общего количества проектов, проводимых организацией.

- **Проектная.**

В организации такого типа подразделения выделяются для проведения конкретных проектов. Руководитель такого временного подразделения является руководителем соответствующего проекта и полностью распоряжается выделенными для него ресурсами.

Эта схема обладает высокой гибкостью и приспособляемостью под нужды проекта, но может требовать дополнительных усилий для составления проектной команды, поскольку слабо мотивирует развитие персонала.

- **Продуктовая.**

Подразделения такой организации отвечают за разработку, развитие и поддержку определенных продуктов или семейств близких продуктов. В каждом таком подразделении может одновременно выполняться несколько проектов, связанных с данным продуктом. Руководителями проектов обычно являются сотрудники этого отдела, которые вполне распоряжаются выделенными для проекта ресурсами.

Продуктовая схема позволяет дополнить гибкость и простоту управления проектами в проектной схеме легкостью подбора подходящего персонала. Недостатком ее может являться выработка слишком узкой специализации у сотрудников и трудности расформирования большого подразделения при отказе от продолжения работ над некоторым продуктом.

- **Ориентированная на клиента.**

Подразделения таких организаций формируются для удовлетворения нужд крупных клиентов или групп клиентов. Проекты для такого клиента ведутся внутри соответствующего

подразделения.

Эта схема позволяет уменьшить усилия, необходимые для понимания нужд клиентов. В целом она похожа на продуктовую, но при возникновении нужды в новом продукте может осложнить подбор персонала в соответствующую группу.

- **Территориальная.**

Подразделения формируются согласно географическому положению. Проекты бывают локальными, целиком проводящимися в рамках одного подразделения, или распределенными — включающими ресурсы нескольких подразделений.

В ее рамках удобнее проводить локальные проекты, а распределенные всегда требуют дополнительных усилий по координации работ.

- **Матричная.**

Это гибрид нескольких схем, обычно проектной или продуктовой и функциональной. В такой организации есть и функциональные подразделения, в которых группируются ресурсы, и проектные группы, формируемые под конкретный проект из служащих функциональных подразделений. Ресурсы проекта передаются в соответствующую группу, и ими распоряжается руководитель проекта. Руководители функциональных подразделений, тем не менее, могут даже во время проекта иметь определенную власть над своими подчиненными.

Эта схема может сочетать достоинства функциональной и проектной, но может и порождать проблемы, связанные с двойной подчиненностью участников проектных групп и разницей между возложенной на них ответственностью и предоставленными полномочиями.

### **Организационная культура**

При выборе той или иной стратегии действий менеджер проекта должен учитывать и организационную культуру организации-исполнителя и других связанных с проектом организаций. Организационной или корпоративной культурой называют совокупность общих убеждений, норм, ценностей и принятых стилей поведения служащих данной организации. Выделяют следующие виды организационной культуры.

- **Иерархическая (закрытая).**

Работа такой организации основана на формальных правилах, четко определенных полномочиях ее служащих и отношениях между ними. Развитие в ней представляется как стабильный и плавный, без изменений сложившийся структуры, рост показателей эффективности и рентабельности основных операций. Решения принимаются только формально уполномоченными на это лицами, многие решения требуют прямого вмешательства высшего руководства.

Работа в такой организации может продвигаться, если она никак не нарушает сложившихся отношений или же патронируется руководством, имеющим полномочия менять существующий порядок в необходимых пределах. Любой новый подход закрепляется формулировкой новых правил и стандартов и повышением эффективности работы организации и ее подразделений при сохранении стабильной структуры бизнес-процессов.

- **Рыночная (открытая).**

Деятельность этой организации ориентирована на завоевание и удержание ее места на рынке. Основные приоритеты — конкурентоспособность, хорошая репутация на рынке и совместная работа всех служащих над их достижением и поддержкой. Решения принимаются руководителями отдельных групп на основании экспертных оценок, исходящих из указанных целей, и обеспечения измеримых результатов на пути к ним. Работа в такой организации продвигается, если она поддерживает ее конкурентоспособность, позволяет оценить вклад отдельных лиц, стимулирует сотрудничество между работниками. Новый подход должен преподноситься в виде гибких, адаптирующихся под нужды рыночной целесообразности норм и получить одобрение экспертов.

- **Инновационная (*произвольная*).**

Работа организации этого типа ориентирована на инновации, создание передовых, уникальных продуктов и услуг. Ценностями в ней служат творческая атмосфера, технологическое лидерство, внедрение новаторских подходов. Работа в ней продвигается, если поддерживает творческие инициативы вовлеченных в нее служащих, использует передовые достижения в некоторой области, способствует созданию уникальных технологий, продуктов и услуг. Нужно убедить каждого участника или «законодателей мод» в полезности ведущейся деятельности, подтвердить использование новаторских подходов и предоставить свободу творческих изменений.

- **Семейная (*синхронная*).**

Такая организация обладает хорошей внутренней атмосферой, активно использует программы вовлечения сотрудников в бизнес и корпоративные обязательства перед ними. Ее ценностями являются гармоничное развитие ее служащих, их профессиональный рост и повышение навыков командной работы, традиции организации, сплоченность коллектива и моральный климат.

Работа в такой организации продвигается, если она обеспечивает профессиональный рост и развитие персонала, повышение удовольствия от работы, обеспечивает движение организации к некоторой цели, достижение которой необходимо обосновать как дополнительную ценность (например, представив как наиболее вероятный сценарий развития данной отрасли).

### **Заинтересованные в проекте лица**

Ход проекта также существенно зависит от намерений, действий, информирования и поддержки нужд его *участников* или *заинтересованных лиц (stakeholders)* — всех лиц и организаций, имеющие связанные с проектом интересы, или тех, на ком результаты проекта как-то отразятся. Заинтересованные в проекте лица могут играть в нем следующие роли.

- **Спонсор проекта.**

Это лицо или группа лиц, предоставляющая ресурсы для проекта или формирующая инвестиционно привлекательную ситуацию вокруг него. Очень важно еще до начала проекта определить его спонсора, поскольку именно с ним нужно решать все вопросы, касающиеся обеспечения проекта ресурсами.

- **Менеджер проекта.**

Это лицо, ответственное за управление проектом, т.е. обеспечение наилучших его результатов в рамках заданных ограничений.

Иногда эту роль называют руководителем проекта, но в последнее время одной из функций руководителя принято считать лидерство: наличие авторитета, умение вовлекать людей в проект, создавать команду, вести людей за собой. Менеджер в классическом понимании не обязан быть лидером, он выполняет только административные обязанности.

- **Лидер проекта.**

Это наиболее авторитетный человек в команде проекта, к чьему мнению прислушиваются, кто принимает большинство технических решений по ходу проекта. Часто лидер и менеджер проекта — одно лицо.

- **Заказчик.**

Это лицо или организация, которые получают результаты проекта в собственность того или иного вида.

- **Пользователи.**

Эта лица и организации, непосредственно использующие результаты проекта в своей деятельности.

- **Организация-исполнитель.**

Это организация, в которой проводится проект и которая несет ответственность перед заказчиком за его выполнение. Такая организация может быть создана для проведения одного конкретного проекта и состоять только из его команды.

- **Команда проекта.**

Это служащие организации-исполнителя, выполняющие работы по проекту. Менеджер проекта и лидер проекта входят в команду.

- **Команда управления проектом.**

Это часть команды проекта, непосредственно участвующая в деятельности по управлению проектом. Сюда входят менеджер проекта и его лидер, может входить секретарь менеджера, эксперты, помогающие в принятии решений и т.д.

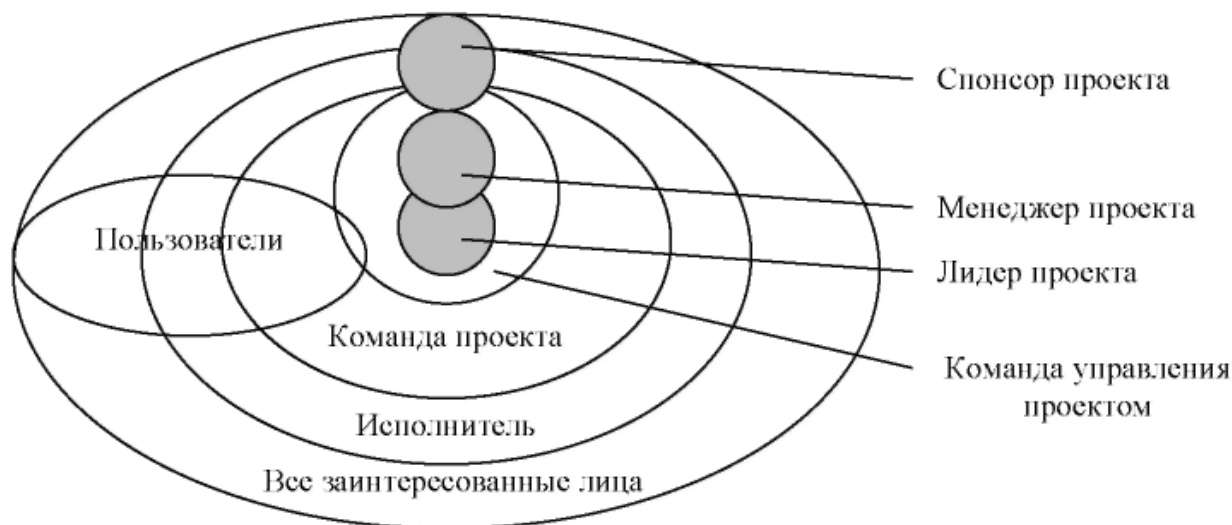


Рисунок 82. Взаимоотношения между заинтересованными лицами проекта.



- ***Источники влияния.***

Это лица, группы лиц или организации, не связанные прямо с проведением проекта и использованием его результатов, но имеющие свои интересы в том или ином развитии событий в проекте и способные повлиять на него. Это могут быть государственные и общественные организации, компании-конкуренты, средства массовой информации, отдельные лица, чью жизнь проект как-то затрагивает, служащие связанных с проектом организаций, способные повлиять на принимаемые решения, или те, чей карьерный рост может зависеть от проекта и пр.

При управлении проектом большое внимание должно уделяться потребностям и ожиданиям всех заинтересованных лиц. Обязанностью руководителя является формирование содержания проекта в таком виде, чтобы их потребности были как-то учтены.

Для этого сначала надо выявить самих заинтересованных лиц и установить их потребности и ожидания. Нужно выработать компромиссное решение (причем часто не одно, а несколько альтернативных), удовлетворяющее противоречивые интересы разных групп. Стоит уделить время тому, чтобы выяснить юридические обязательства проекта перед каждым из них, набор благоприятных возможностей или проблем, которые они могут создать, определить их возможные стратегии и стратегию поведения по отношению к каждому заинтересованному лицу и форму взаимодействия с ним, в частности, его информирования. При этом следует учитывать степень влияния каждого из заинтересованных лиц, силу и длительность воздействия, которое они могут оказать на проект, действуя ему на пользу или во вред. Обычной ошибкой является пренебрежение теми заинтересованными лицами, которые не связаны с проектом юридически, но имеют в нем свои интересы и могут серьезно влиять на его ход.

Нужно проводить переговоры, обеспечивать информационное освещение проекта, поддерживать контакты так, чтобы показывать разным лицам, как их нужды будут учтены, убеждать их действовать на благо проекта (или не действовать против него). По крайней мере, следует вовремя узнавать об их возможных шагах и планировать ответные или поддерживающие действия.

### **Виды деятельности, входящие в управление проектом**

Виды деятельности, которыми приходится заниматься руководителю проекта или группе управления проектом для обеспечения его успешного выполнения, можно разделить на следующие области.

- ***Управление содержанием проекта и качеством.***

В эту область входят четкое определение целей проекта, его точного содержания (project score — что именно должно быть сделано в его рамках, какие результаты должны быть получены, включая все промежуточные, и какие работы должны быть проведены для этого), определение критериев качества результатов, процедур его обеспечения и контроля, выполнение этих процедур, а также критерии завершения проекта и действия по его завершению.

- ***Управление ресурсами проекта.***

Эта область включает выполнение оценок ресурсоемкости работ, включая их стоимость и продолжительность, составление графиков выполнения работ в проекте. Кроме того, практически отдельной областью является *управление персоналом проекта*, которое включает планирование использования персонала в проекте, обучение персонала и набор команды проекта, организацию командной работы и мотивацию, делегирование полномочий и управление конфликтами.

- *Управление рисками.*

Управление проектными рисками связано с преодолением или использованием на благо проекта последствий неопределенности, всегда окружающей его, и ошибок, которые люди допускают в любой сложной деятельности. Управление рисками включает их выявление, анализ и оценку, выработку стратегии действий в отношении рисков, их отслеживание в ходе проекта и реализацию действий по их уменьшению, преодолению или использованию.

- *Управление коммуникациями и информационное обеспечение проекта.*

Включает составление предложений по проекту, выбор продавцов, поставщиков и субподрядчиков, ведение переговоров, определение информационной политики для внешних заинтересованных лиц и внутри проектной команды, выработку способов информирования участников проекта и согласования решений, определение процедур подготовки и содержания отчетов, передачу результатов в использование.

- *Управление конфигурациями и изменениями.*

Эта область связана с установлением процедур обработки изменений и запросов на их выполнение, включения их в содержание проекта (или удаления за его рамки). В рамках этой же деятельности должны определяться конфигурации материалов проекта — согласованные наборы определенных версий всех продуктов и документов, вырабатываемых в ходе проекта, правила выделения таких конфигураций и перехода от одной конфигурации к другой.

- *Управление проектной средой и технологиями.*

В рамках этой деятельности определяются методы, техники, инструменты и технологии, используемые в проекте, подбираются необходимые инструменты, материалы и помещения, они подготавливаются к использованию и настраиваются под нужды проекта.

- *Контроль и мониторинг состояния проекта.*

Эта область включает все действия, связанные с получением достоверной информации о текущем состоянии проекта, отслеживание выполнения всех планов и процедур. Полученные данные анализируются и могут приводить к выполняемым по ходу проекта действиям из других областей, направленным на предотвращение возникновения проблем и разрешение уже появившихся.

Определение любой процедуры, которое вырабатывается в ходе одной из деятельности в рамках управления проектом, должно включать определение условий ее запуска, входных документов и материалов, действий, которые необходимо выполнить, указание возможных способов их выполнения, условий окончания и выходных документов. В ходе проекта все установленные процедуры должны выполняться при достижении условий их запуска.

Далее подробнее будут рассмотрены перечисленные области деятельности, кроме последних трех. Некоторые вопросы, относящиеся к последним трем, более техническим областям, будут освещены в рамках других областей.

### **Управление содержанием проекта и качеством**

Управление содержанием проекта (project scope) является одним из критически важных для его успеха видов деятельности. Проект с нечетко определенным содержанием обречен на неудачу. Ясное же его определение — как постановка правильного вопроса — дает половину успешного ответа.

Большинство действий, связанных с определением содержания проекта, должно быть проведено уже на этапе подготовки к нему, еще до официального начала, — иначе не будет достаточных данных для принятия решения о запуске проекта.

Начинать определение содержания проекта надо с выяснения его целей. Основная цель должна описываться одной фразой и делать понятным предназначение проекта, очерчивать реальные потребности некоторых лиц и организаций, для удовлетворения которых предназначены его результаты.

После определения целей можно уточнять общие требования к продуктам или услугам, которые должны быть получены в результате проекта, основные ограничения на возможные решения, способы получения решений, включающие несколько альтернатив.

Основные элементы содержания проекта следующие.

- ***Целевые критерии проекта.***

Эти критерии определяют ряд показателей и их значения, означающие успех проекта, а также другие значения, говорящие о необходимости его закрытия. Цели проекта обязательно должны иметь составляющие, ориентированные на клиентов и пользователей его результатов, иначе даже самый успешный внутри организации проект может привести к созданию продуктов, которые не будут использоваться.

- ***Иерархическая структура работ (work breakdown structure).***

Это иерархическое разбиение всей работы, которую необходимо выполнить для достижения целей проекта, на более мелкие операции и действия, до такого уровня, на котором способы выполнения этих действий вполне ясны и соответствующие работы могут быть оценены и спланированы. Она включает также определение промежуточных результатов всех составляющих эту структуру работ.

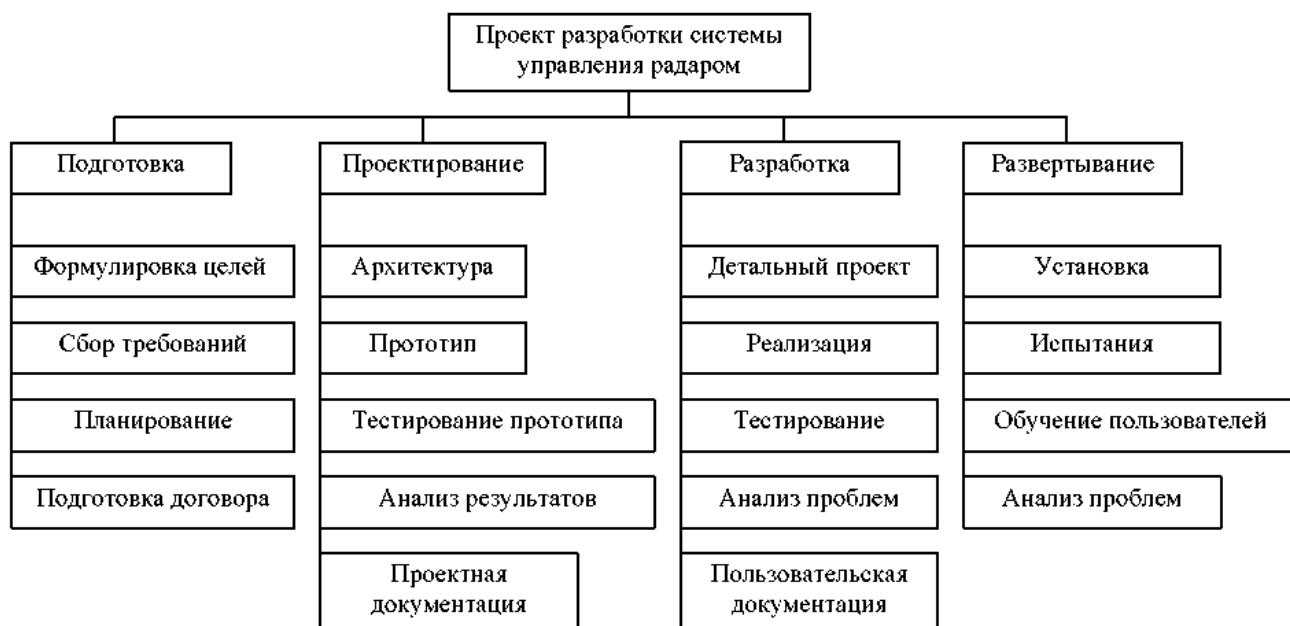


Рисунок 83. Пример структуры работ проекта, построенной на основе декомпозиции задач.

Обе составляющие содержания проекта, конечно же, требуют подтверждения у заинтересованных лиц. Только после определения этих двух элементов можно проводить обоснованное планирование работ и ресурсов для их выполнения, определение конфигураций проекта, необходимых для его выполнения инструментов и технологий и возможных рисков. Пропуски в структуре работ выливаются в неожиданные работы и возрастание затрат как времени, так и денег на проект в целом.

При разработке структуры работ можно взять за основу набор задач, полученных декомпозицией целей и задач проекта в целом, или набор промежуточных результатов (deliverables), передаваемых заказчику, который позволит постепенно получить необходимые итоговые результаты. Пример структуры задач для гипотетического проекта разработки ПО управления радаром аэропорта, построенный первым способом, показан на Рис. 83. Аналогичный пример, построенный вторым способом, показан на Рис. 84.

Управление качеством тоже становится возможным, когда выявляются все результаты, которые нужно получить в ходе проекта, и все работы, которые необходимо выполнить. Оно включает выделение ключевых показателей качества для всех результатов, окончательных и промежуточных, а также выполняемых работ. Должны быть определены используемые метрики качества, целевые значения, которые должны быть достигнуты по этим метрикам в ходе проекта, техники, которые позволят обеспечить достижение этих показателей, способы проверки того, что они достигнуты, а также процедуры устранения обнаруженных дефектов.



Рисунок 84. Пример структуры работ проекта, построенной на основе поставляемых результатов.

### Метрики ПО

Одна из базовых задач управления ресурсами — адекватная оценка затрат ресурсов на отдельные выполняемые в проекте работы. Такая оценка дает возможность планирования работ проекта и затрат на их проведение. При оценке ресурсоемкости работ основную роль играют выбранные *метрики*, с помощью которых измеряется сложность или трудоемкость работ, а также требующееся на их выполнение время.

При выборе метрик нужно четко определить, что именно мы хотим измерить, и постараться найти измеримый показатель, как можно более близкий к этому. Типичные ошибки состоят как в том, что ничего не измеряется, так и в том, что измеряются вещи, слабо связанные с тем, что хотелось бы оценивать, лишь бы что-нибудь измерять и заносить в отчеты стройные ряды чисел.

Например, измерение времени, проводимого служащим на рабочем месте в качестве меры интенсивности его труда и вклада в процветание организации, не вполне адекватно: человек может занять себя на работе и другими вещами. Соответственно, введение административных мер по поощрению «много работающих» и наказанию «мало работающих» только на основании таких измерений может иметь «неожиданные» последствия — показатели «хорошей работы» будут расти, а реальная работа не станет выполняться быстрее или лучше.

На настоящий момент не существует достаточно адекватных и одновременно простых в использовании метрик трудоемкости разработки ПО, которые можно было бы оценивать до создания этого ПО и применять для планирования его разработки. Скорее всего, они и не будут найдены. В этой ситуации используются разнообразные эмпирические подходы, комбинирующие простые в использовании метрики со сложными, но более адекватными.

Одна из первых идей, которая приходит в голову при оценке трудоемкости и времени разработки ПО, — оценить сначала сложность или размер программы, а потом умножить полученную оценку на производительность исполнителей. Похоже, однако, что природа

разработки программ такова, что ее трудоемкость слабо связана с размерами результата (например, приводимая ниже модель COCOMO, описанная ниже, выражает эту связь в достаточно сложном виде). Часто оказывается, что оценить сразу трудоемкость по аналогии с имеющимися примерами можно точнее, чем оценив сначала размер. Тем не менее, метрики размера и сложности ПО часто используются для оценки трудоемкости разработки.

Самой простой и наиболее широко используемой метрикой является размер программы в строках ее кода (lines of code, LOC). Ее основное достоинство — понятность и простота вычисления. Ее недостатки — не очень хорошая адекватность в качестве метрики трудоемкости разработки программы, зависимость от используемых языков и технологий и трудность предварительной оценки размера ПО. Практика показывает, что качественная программа часто имеет несколько меньший объем, чем программа с теми же функциями, но менее удобная для сопровождения или совершающая больше ошибок. В то же время на разработку первой программы может уйти в два-три раза больше усилий.

С другой стороны, производительность разных разработчиков очень сильно отличается, но обычно руководители групп и организаций примерно представляют себе среднее значение по группе или организации. В терминах строк кода она обычно лежит в пределах от 5000 до 50000 строк хорошо отлаженного кода без учета комментариев за 1 человеко-год.

Более хитрой метрикой сложности программы являются функциональные точки (functional points, FP). Количество функциональных точек в программной системе вычисляется примерно следующим образом.

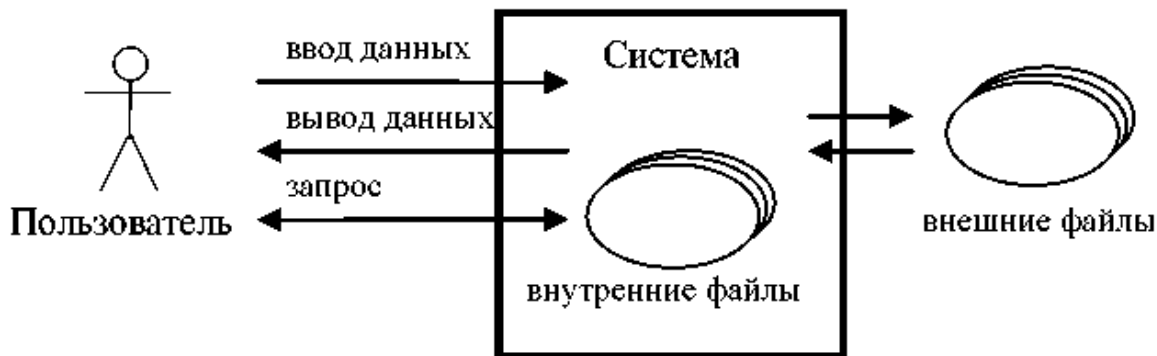


Рисунок 85. Схема рассмотрения системы при оценке ее сложности в функциональных точках.

- Выделяются обращения к системе с целью ввода данных, с целью получения каких-то уже имеющихся в ней данных (отчеты), и с запросами, в ходе которых данные вводятся в систему, перерабатываются и выдаются какие-то результаты. Дополнительно определяются группы взаимосвязанных данных (называемые файлами) внутри системы и аналогичные группы, лежащие вне ее, но используемые в ее работе.
- Для всех данных из перечисленных пяти категорий оценивается их сложность (по шкале «низкая»-«средняя»-«высокая»).
- Итоговая сложность программной системы вычисляется как сумма сложностей выявленных отдельных представителей этих пяти категорий. Сложность ввода, вывода,

запроса или группы данных вычисляется умножением оценки сложности составляющих данных на весовой коэффициент, который можно найти в стандартах или определить на основе собственного опыта. Обычно весовые коэффициенты групп данных больше, чем коэффициенты для вводов, выводов или запросов.

Количество строк кода, приходящихся на одну функциональную точку, зависит от используемых технологий и языка программирования и меняется от 300 для программирования на ассемблере до 5-10 для компонентных технологий на базе языков высокого уровня.

Другие исходные элементы используются при подсчете так называемых объектных точек. В этом случае рассматриваются экраны, формы и отчеты, присутствующие в системе, классы, а также модули, написанные на необъектных языках. Сложность каждого из таких элементов оценивается отдельно, после чего их сложности складываются, тоже с разными весовыми коэффициентами для разных категорий элементов.

Обе эти метрики хорошо применимы к так называемым информационным системам, т.е. системам, основные функции которых связаны с накоплением и хранением больших объемов данных, а также с предоставлением доступа и интерактивной обработкой запросов к ней. Оценка в таких терминах компилятора, системы обмена сообщениями или автоматизированной системы навигации корабля будет менее адекватной.

Наиболее известным методом оценки трудоемкости и времени проекта, основанным на большом количестве данных из проведенных ранее проектов, является *конструктивная модель стоимости* версии II (*Constructive Cost Model II, COCOMO II*).

В рамках этой модели оценки трудоемкости проекта и времени, требующегося на его выполнение, определяются тремя разными способами на разных этапах проекта.

- На самых ранних этапах, когда примерно известны только общие требования, а проектирование еще не начиналось, используется *модель состава приложения* (*Application Composition Model*). В ее рамках трудоемкость проекта оценивается в человеко-месяцах по формуле  $Effort = A * Size$ .
  - Size представляет собой оценку размера в терминах экранов, форм, отчетов, компонентов и модулей будущей системы. Каждый такой элемент оценивается с коэффициентом от 1 до 10 в зависимости от своей сложности.
  - Коэффициент  $A$  учитывает возможное переиспользование части компонентов и производительность разработки, зависящую от опытности команды и используемых инструментов и оцениваемую числом от 4 до 50.

$$A = (100 - (\text{процент переиспользования})) / \text{производительность}.$$

- На следующих этапах, когда требования уже в основном известны и начинается разработка архитектуры ПО, используется *модель этапа предварительного проектирования* (*Early Design Model*) и следующие формулы.

Для трудоемкости (в человеко-месяцах):

$$Effort = A * (Size)^B * M_E + (\text{трудоzатраты на автоматически генерируемый код})$$

Для времени (в месяцах):

$$Time = T * Efforts^{(0,28+0,2*(B-1,01))} * Sced$$

- Коэффициент  $A$  считается равным 2,45, а  $T$  считается равным 3,67.
- $Size$  — оценка размера ПО в тысячах строк кода.
- $B$  — фактор процесса разработки, который вычисляется по формуле:

$$B = 0,91 + 0,01 * \sum_{i=1...5} W_i,$$

где факторы  $W_i$  принимают значения от 0 до 5:

- $W_1$  — предсказуемость проекта для данной организации, от полностью знакомого (0) до совсем непредсказуемого (5);
- $W_2$  — гибкость процесса разработки, от полностью определяемого командой при выполнении общих целей проекта (0) до полностью фиксированного и строгого (5);
- $W_3$  — степень удаления рисков, от полной (0) до небольшой (5), оставляющей около 80% рисков;
- $W_4$  — сплоченность команды проекта, от безукоризненного взаимодействия (0) до больших трудностей при взаимодействии (5);
- $W_5$  — зрелость процессов в организации, от 0 до 5 в виде взвешенного количества положительных ответов на вопросы о поддержке ключевых областей процесса в модели CMM.
- $M_E$  — произведение семи коэффициентов затрат, каждый из которых лежит в интервале от 1 до 6:
  - возможности персонала;
  - надежность и сложность продукта;
  - требуемый уровень повторного использования;
  - сложность платформы;
  - опытность персонала;
  - использование инструментов;
  - плотность графика проекта.
- $Efforts$  обозначает оценку трудоемкости без учета плотности графика, а  $Sced$  — требуемое сжатие времени выполнения проекта.
- После того, как разработана архитектура ПО, оценки должны выполняться с использованием *постархитектурной модели (Post-Architecture Model)*. Формула для трудоемкости (в человеко-месяцах):
 
$$Efforts = A * (K_{req} * Size)^B * M_p + (\text{трудоzатраты на автоматически генерируемый код}).$$
 Для времени — та же формула, что и в предыдущей модели (в месяцах):
 
$$Time = T * Efforts^{(0,28+0,2*(B-1,01))} * Sced.$$
  - $Size = (\text{размер нового кода в тыс. строк}) + RSize$ , где
 
$$Rsize = (\text{размер переиспользуемого кода в тыс. строк}) * (100 - AT) / 100 * (AA + 0,4 * DM + 0,3 * CM + 0,3 * IM + SU) / 100$$
    - $AT$  — процент автоматически генерируемого кода;



- $AA$  — фактор трудоемкости перевода компонентов в повторно используемые, от 0 до 8;
  - $DM$  — процент модифицируемых для переиспользования проектных моделей;
  - $CM$  — процент модифицируемого для переиспользования кода;
  - $IM$  — процент затрат на интеграцию и тестирование повторно используемых компонентов;
  - $SU$  — фактор понятности переиспользуемого кода, от 10 для простого, хорошо структурированного, до 50 для сложного и непонятного; равен 0, если  $CM=DM=0$ .
- Все коэффициенты, кроме  $K_{req}$  и  $M_p$ , имеют те же значения, что и в предыдущей модели.
  - Коэффициент  $K_{req}$  вычисляется как  $(1 + (\text{процент кода, выброшенного из-за изменений в требованиях})/100)$ .
  - Коэффициент  $M_p$  является произведением 17-ти коэффициентов затрат, имеющих значения от 1 до 6:
    - надежность продукта;
    - сложность продукта;
    - размер базы данных разрабатываемого приложения;
    - требуемый уровень повторного использования;
    - требуемый уровень документированности;
    - уровень производительности по времени;
    - уровень требований к занимаемой оперативной памяти;
    - изменчивость платформы;
    - возможности аналитика проекта;
    - возможности программистов;
    - опыт работы команды в данной предметной области;
    - опыт работы команды с используемыми платформами;
    - опыт работы команды с используемыми языками и инструментами;
    - уровень текучести персонала в команде;
    - возможности используемых инструментов;
    - возможности общения между членами команды;
    - фактор сжатия графика проекта.

Для тех, кто уже отчаялся понять все хитросплетения этой модели, скажем, что имеются программные инструменты, автоматизирующие расчеты по ее формулам.