Основы управления ИТ проектами.

Лекция 1.

Термин "ИТ-проект" обычно используется для обозначения деятельности, связанной с использованием или созданием некоторой информационной технологии. Это приводит к тому, что ИТ-проекты охватывают очень разнообразные сферы деятельности: разработку программных приложений, создание информационных систем, *развертывание* ИТ-инфраструктуры и пр. В этой книге мы будем часто говорить о проектах создания информационных систем, подразумевая реализацию каких-то информационных технологий.

С одной стороны, эти работы соответствуют классическому определению проекта : "Проект – это комплекс усилий, предпринимаемых с целью получения конкретных уникальных результатов в рамках отведенного времени и в пределах утвержденного бюджета, который выделяется на оплату ресурсов, используемых или потребляемых в ходе проекта". С другой стороны, они обладают известными отличительными особенностями:

- разделение на уровне идеологии заказчика и исполнителя: заказчиком, как правило, является бизнес, а исполнителем – ИТ-специалисты, и есть трудности в выявлении требований, ожиданий от проекта, в формировании технического задания. Существует также проблема эффективных коммуникаций;
- ответственность за результат проекта имеет "солидарный" характер. То есть здесь нельзя возложить ответственность за успех проекта только на исполнителя, точно так же, как нельзя говорить, что исключительно заказчик виновен в том, что проект не удался. В ИТ-проекте должны создаваться определенные условия для взаимодействия сторон, и стороны, участвующие в нем, несут равную ответственность за результаты проекта;
- зачастую реализация ИТ-проекта предусматривает изменение существующих организационных структур на предприятии;
- обычно в ИТ-проект вовлечено множество подразделений организации;
- существует высокая вероятность конфликтов между руководителем проекта, высшим руководством, руководителями подразделений и персоналом организации;
- многие ИТ-проекты имеют колоссальные бюджеты. В крупных компаниях масштабы проектной деятельности в области информационных технологий (ИТ) измеряются миллионами долларов, причем реализация новых проектов происходит постоянно. Если, например, промышленное предприятие достаточно один раз построить – и оно будет работать, не требуя регулярных инвестиций, то развитие ИТ-инфраструктуры в растущих компаниях требует больших и регулярных вложений. Большие бюджеты, в свою очередь, подразумевают больший уровень ответственности и, соответственно, больший уровень компетенции тех людей, которые этими проектами управляют.

Если говорить о реализации ИТ-проектов, следует обратить внимание на следующие особенности:

- зачастую в компании заказчика одновременно выполняются несколько ИТ-проектов;
- приоритеты выполнения проектов постоянно корректируются;
- по мере реализации проектов выполняется уточнение и корректировка требований и содержания проектов;
- велико влияние человеческого фактора: сроки и качество выполнения проекта в основном зависят от непосредственных исполнителей и коммуникации между ними;
- каждый исполнитель может принимать участие в нескольких проектах;
- налицо трудности планирования творческой деятельности, отсутствуют единые нормативы и стандарты;
- сохраняется повышенный уровень риска, вплоть до непредсказуемости результатов;
- происходит постоянное совершенствование технологии выполнения работ.

Анализ статистики показывает, что примерно 90 процентов ИТ-проектов аналогичны уже выполненным. У руководителя проекта имеется *опыт* реализации таких задач и понимание возможных проблем. В этих случаях иерархическая структура проекта и *работ*(ИСП/ИСР) формируется с применением подхода Top-down (сверху вниз), используется *типовая* структура проектной команды, планы проекта (план управления рисками, *план коммуникаций* и пр.) аналогичны планам предыдущих проектов. Однако 10 процентов проектов – инновационные, реализуемые "с нуля" и требующие творчества, нестандартных решений и управленческой смелости. Принятие решений в таких проектах характеризуется высокими рисками, что требует от руководителя глубоких знаний методики проектного управления и понимания особенностей её применения в сфере информационных технологий.

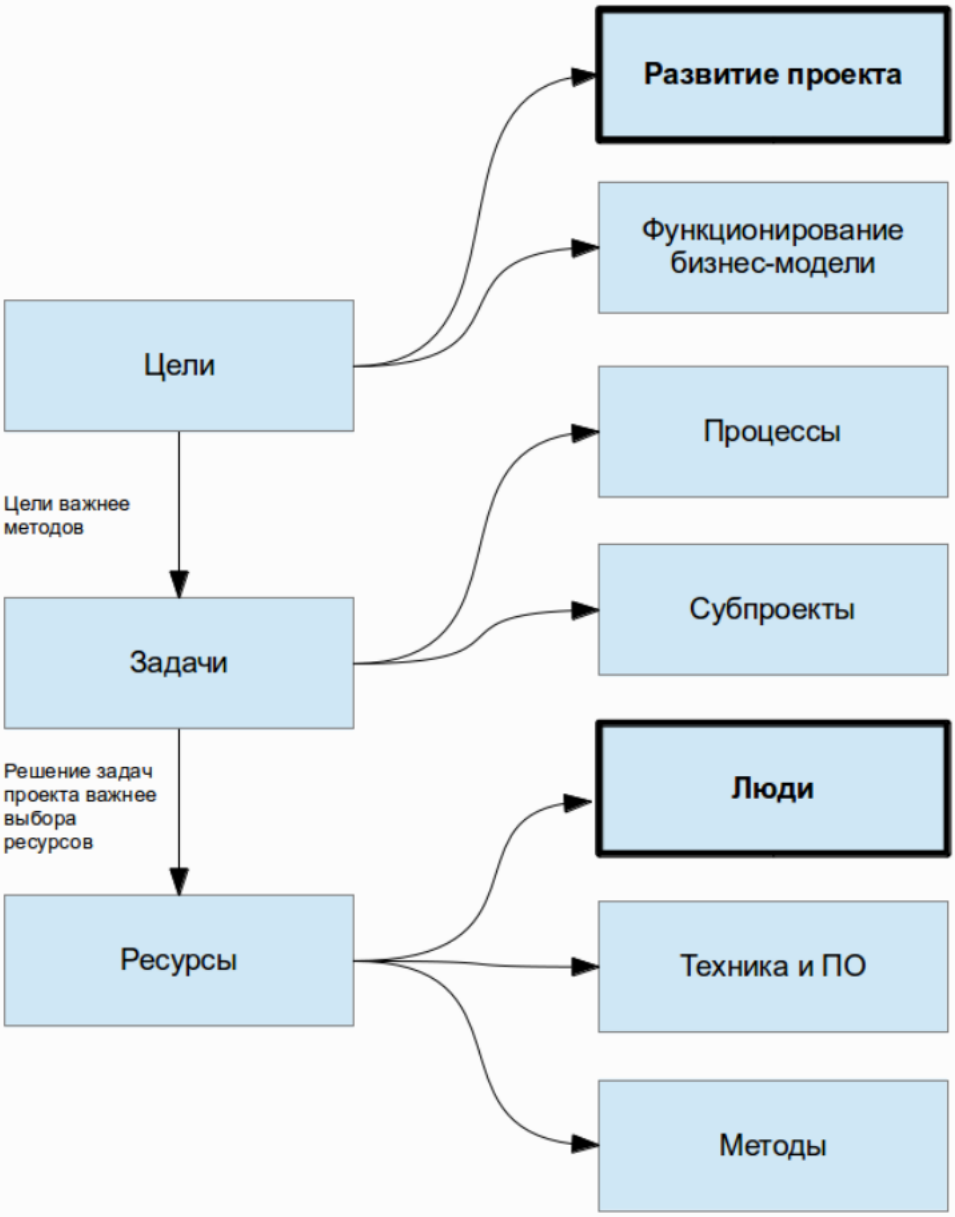
Применение методологии управления проектами позволяет зафиксировать цели и результаты проекта, дать им количественные характеристики, определить временные, стоимостные и качественные параметры проекта, создать реалистичный *план выполнения*проекта, выделить, оценить риски и предотвратить возможные негативные последствия во время реализации проекта.

Для эффективного управления проект должен быть хорошо структурирован. Суть этого процесса сводится к выделению следующих основных элементов:

- фазы жизненного цикла проекта, этапов, работ и отдельных задач;
- организационная структура исполнителей проекта;
- структура распределения ответственности.

Жизненный цикл – это последовательность фаз проекта, через которые он должен пройти для гарантированного достижения целей проекта, в нашем случае – для реализации некоторой информационной технологии.

Организационная структура подразумевает выделение ролей исполнителей, которые необходимы для реализации проекта, *определение* взаимоотношений между ними и распределение ответственности за выполнение задач.
Менеджмент — это управление. В нашем случае, управление проектом.
Понятно, что управление проектом — это работа над его составляющими.



Важнейшие исключения из любых правил.

Основная ценность — это люди.
Основной приоритет — развитие проекта.

Менеджер работает с процессами. Процессы являются составной частью проектов.

Процесс может быть разовым или непрерывным, но он в любом случае итеративен. Это означает, что у каждого процесса есть циклические свойства — он легко может быть повторен, и даже для начала нового процесса возможно применять наработанный опыт — академические методики, личный опыт, опыт коллег и так далее.

До начала процесса необходимо формализовать исходные данные и выделить цели.

Этап анализа является опциональным. Он проводится, в зависимости от масштабов и цены процесса. Если процесс дорогой — все исходные данные подвергаются детализации, информация дополняется схемами и резюме.

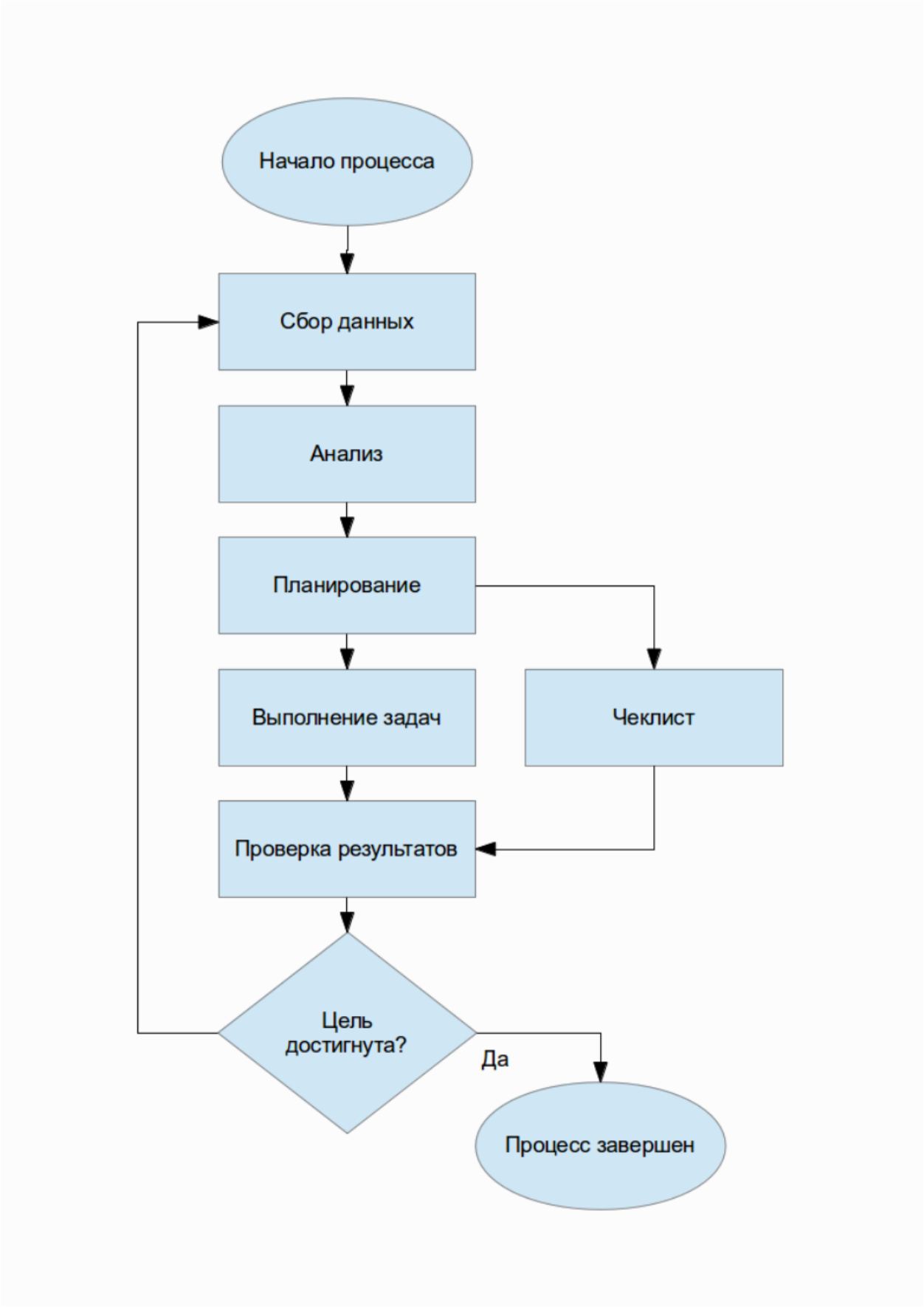
На этапе планирования выбираются методы решения задачи, определяется, как именно будет осуществляться процесс.

Для обеспечения корректности приемки еще на этапе планирования составляется чеклист — список критериев, который однозначно дает понять, что проект завершен.

Естественно, исполнителю должна быть доступен максимальный объем информации, связанный с процессом, в котором он участвует — исходные данные, цели и требования в чеклисте.

Если процесс не является непрерывным — по достижению целей он может быть завершен.

При повторном выполнении процесса к исходным данным добавляются результаты предыдущей итерации.


Начало процесса
Сбор данных
Анализ
Планирование
Выполнение задач
Чеклист
Проверка результатов
Цель достигнута?
Да
Процесс завершен

Общепринятого подхода к определению жизненного цикла проекта, его фаз, стадий и этапов не существует и, вероятно, не может существовать. Так как все эти характеристики зависят от специфики конкретного проекта, условий его реализации и опыта участников. Тем не менее логика и содержание процессов развития проектов имеют много общего и наиболее полно и ясно представлены в схеме жизненного цикла проекта американского Института управления проектами (PMI) (рис. 4).

Рис. 4 Жизненный цикл проекта (PMI, США)

1.5.1. Основные фазы жизненного цикла проекта

Типичный жизненный цикл проекта, как видно из рис. 4, состоит из четырех фаз:

1. Начальная фаза (концепция).

2. Фаза разработки.

3. Фаза реализации.

4. Фаза завершения.

Какие работы входят в состав основных фаз проекта?

1.5.2. Начальная фаза

Посвящена разработке концепции проекта и включает в себя:

· Сбор исходных данных и анализ существующего состояния

(предварительное обследование).

· Выявление потребности в изменениях (в проекте).

· Определение проекта:

· цели, задачи, результаты;

· основные требования, ограничительные условия, критерии;

· уровень риска;

· окружение проекта, потенциальные участники;

· требуемое время, ресурсы, средства и др.

· Определение и сравнительная оценка альтернатив.

· Представление предложений, их апробация и экспертиза.

· Утверждение концепции и получение одобрения для следующей фазы.

1.5.3. Фаза разработки

Разрабатываются основные компоненты проекта и осуществляется подготовка к его реализации. Основные работы этой фазы:

· Назначение руководителя проекта и формирование команды проекта, в первую очередь ключевых членов команды.

· Установление деловых контактов и изучение целей, мотивации и требований заказчика и владельца проекта, других ключевых участников.

· Развитие концепции и разработка основное содержания проекта:

· конечный результат (ы) и продукт (ы),

· стандарты качества,

· структура проекта,

· основные работы,

· требуемые ресурсы.

· Структурное планирование, в т. ч.:

· декомпозиция проекта, в т. ч. WBS,

· календарные планы и укрупненные графики работ и обеспечения,

· смета и бюджет проекта,

· потребность в ресурсах,

· процедуры УП и техника контроля,

· определение и распределение рисков.

· Организация и проведение торгов, заключение субконтрактов с основными исполнителями.

· Организация выполнения базовых проектных и опытно-конструкторских работ по проекту.

· Представление проектной разработки.

· Получение одобрения на продолжение работ по проекту.

1.5.4. Фаза реализации

Выполняются основные работы, необходимые для достижения проекта. Данная фаза включает в себя:

· Организация и проведение торгов, заключение контрактов.

· Полный ввод в действие разработанной системы УП.

· Организация выполнения работ.

· Ввод в действие средств и способов коммуникации и связи участников проекта.

· Ввод в действие системы стимулирования (участников) проекта.

· Детальное проектирование и технические спецификации.

· Оперативное планирование работ.

· Установление системы информационного контроля за ходом работ.

· Организация и управление материально-техническим обеспечением работ, в т. ч. запасами, покупками, поставками.

· Выполнение работ, предусмотренных проектом (в т. ч. производство строительно-монтажных и пуско-наладочных работ).

· Руководство, координация работ, согласование темпов, мониторинг прогресса. прогноз состояния, оперативный контроль и регулирование основных показателей проекта:

· ход работ, их темпы,

· качество работ и проекта,

· продолжительность и сроки,

· стоимость и другие показатели.

· Решение возникающих проблем и задач.

1.5.5. Завершающая фаза или окончание проекта

Достигаются конечные цели проекта, подводятся итоги, разрешаются конфликты и проводится закрытие проекта. Основные работы этой фазы:

· Планирование процесса завершения.

· Эксплутационные испытания окончательного продукта (ов) проекта.

· Подготовка кадров для эксплуатации создаваемого объекта.

· Подготовка документации, сдача объекта заказчику и ввод в эксплуатацию.

· Оценка результатов проекта и подведение итогов.

· Подготовка итоговых документов.

· Закрытие работ и проекта.

· Разрешение конфликтных ситуаций.

· Реализация оставшихся ресурсов.

· Накопление фактических и опытных данных для последующих проектов.

· Расформирование команды проекта.

Обращаем ваше внимание на то, что работы последних трех фаз проекта могут выполнятся как последовательно, так и параллельно.

ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ

Требование – это условие, которому должно удовлетворять программное обеспечение, или свойство, которым оно должно обладать, чтобы:

- удовлетворить потребность пользователя в решении некоторой задачи;
- удовлетворить требования контракта, спецификации или стандарта.

Спецификация требований к ПО является основным документом, определяющим план разработки ПО. Все требования, указанные в спецификации, делятся на функциональные и нефункциональные. Функциональные требования определяют действия, которые должна выполнять система, без учета ограничений, связанных с ее реализацией. Тем самым функциональные требования определяют поведение системы в процессе обработки информации. Нефункциональные требования не определяют поведение системы, но описывают атрибуты системы или атрибуты системного окружения. Можно выделить следующие типы нефункциональных требований:

- требования к применению, которые определяют качество пользовательского интерфейса, документации и учебных курсов;
- требования к производительности, которые накладывают ограничения на функциональные требования, задавая необходимую эффективность использования ресурсов, пропускную способность и время реакции;
- требования к реализации, которые предписывают использовать определенные стандарты, языки программирования, операционную среду и др.;
- требования к надежности, которые определяют допустимую частоту и воздействие сбоев, а также возможности восстановления;
- требования к интерфейсу, которые определяют внешние сущности, с которыми может взаимодействовать система, и регламент этого взаимодействия.

Методы выявления требований:

- собеседование (интервьюирование);
- анкетирование;
- проведение совещаний;
- сессии по выявлению требований (мозговой штурм);
- раскадровка (storyboard);
- создание и демонстрация работающих прототипов;
- ролевые игры.

В ходе проекта работа с требованиями делится на следующие этапы:

- Определение типов требований и групп участников проекта, работающих с ними.
- Первичный сбор требований, их классификация и занесение в базу данных требований.
- Использование базы данных требований для управления проектом.

Любое требование в базе проекта имеет следующие атрибуты:

- Приоритет (высокий, средний, низкий).
- Статус (предложено, одобрено, реализовано, верифицировано).
- Стоимость реализации (высокая, средняя, низкая – или числовое значение).
- Сложность реализации (высокая, средняя, низкая).
- Стабильность (высокая, средняя, низкая).
- Ответственный исполнитель.

Хороший набор требований удовлетворяет следующим показателям качества (IEEE 830-1998 «Recommended Practice for Software Requirements Specifications»):

- Корректность или адекватность (соответствие реальным потребностям).
- Недвусмысленность (однозначность понимания).
- Полнота (отражение всех выделенных потребностей и всех возможных ситуаций, в

которых придется работать системе).

- Непротиворечивость (согласованность между различными элементами).
- Упорядоченность по приоритету и стабильности.
- Проверяемость (выполнение каждого требования нужно должно проверяться достаточно эффективным способом – непроверяемые требования должны быть удалены из рассмотрения или сведены к проверяемым вариантам).
- Модифицируемость (оформление в удобных для внесения изменений структуре и стилях).
- Прослеживаемость в ходе разработки (возможность увязать требование с подсистемами, модулями и операциями, ответственными за его выполнение, и с тестами, проверяющими его выполнение).

Выявленные требования к ПО оформляются в виде ряда документов и моделей. К основным документам, регламентируемым технологией Rational Unified Process, относятся:

- Концепция – определяет глобальные цели проекта и основные особенности разрабатываемой системы. Существенной частью концепции является постановка задачи разработки, определяющая требования к выполняемым системой функциям.
- Словарь предметной области (глоссарий) – определяет общую терминологию для всех моделей и описаний требований к системе. Глоссарий предназначен для описания терминологии предметной области и может быть использован как словарь данных системы.
- Дополнительные спецификации (технические требования) – содержит описание нефункциональных требований к системе, таких, как надежность, удобство использования, производительность, сопровождаемость и др.

Функциональные требования к системе моделируются и документируются с помощью вариантов использования (use case). *Вариант использования* (use case) – связный элемент функциональности, предоставляемый системой при взаимодействии с действующими лицами. *Действующее лицо* (actor) – роль, обобщение элементов внешнего окружения системы, ведущих себя по отношению к системе одинаковым образом.

Каждый вариант использования фиксирует соглашение между участниками проекта относительно поведения системы. Он описывает поведение системы при различных условиях, как реакцию системы запрос одного из участников, называемого основным действующим лицом. Основное действующее лицо инициирует взаимодействие с системой, чтобы добиться некоторой цели. Система отвечает, соблюдая интересы всех участников.

Варианты использования – это вид документации, применяемый, когда требуется сконцентрировать усилия на обсуждении принципиальных функциональных требований к разрабатываемой системе, а не на подробном их описании. Стиль их написания зависит от масштаба, количества участников и критичности проекта. Формат описания варианта использования (по Коберну):

- Имя – цель в виде краткой активной глагольной фразы
- Контекст использования – более длинное описание цели
- Область действия
- Основное действующее лицо
- Участники и интересы
- Предусловие (определяет, выполнение какого условия гарантирует система перед тем, как разрешить запуск варианта использования)
- Минимальные гарантии (наименьшие обещания системы участникам, в частности, когда цель основного действующего лица не может быть достигнута)
- Гарантии успеха (или постусловие – postcondition – устанавливает, что интересы участников удовлетворяются по успешном завершении варианта использования в

конце основного сценария)

- Триггер (событие, которое запускает вариант использования)
- Основной сценарий (простой для понимания типичный сценарий, в котором достигается цель основного действующего лица и удовлетворяются интересы всех участников)
- Расширения (запускаются при возникновении определенного условия, содержат последовательность шагов, описывающих, что происходит при этом условии, и заканчивается достижением цели или отказом от неё)
- Список изменений в технологии и данных
- Вспомогательная информация

Существуют четыре уровня точности описания вариантов использования, расположенные по степени повышения точности:

- Действующие лица и цели (перечисляются действующие лица и все их цели, которые будет обеспечивать система). На этом уровне определяется границы системы, контекст в котором она работает. Действующее лицо может: быть активным, посылая запросы в систему; быть пассивным, получая данные от системы. Его роль может исполнять человек – пользователь, устройство, внешняя программная система, время (в случае если функциональность запускается по расписанию), температура или другое свойство состояния окружающей среды, играющее роль триггера.
- Краткое изложение варианта использования (в один абзац) или основной поток событий (без анализа возможных ошибок).
- Условия отказа (анализ мест возникновения возможных ошибок в основном потоке событий).
- Обработка отказа (написание всех альтернативных потоков событий).

Введение перечисленных уровней преследует своей целью грамотное планирование и экономию времени разработки. В итерационном цикле создания системы не следует пытаться за один прием подробно описать все требования, их нужно постепенно уточнять, повышая уровень точности. Выбор первоочередных вариантов использования для уточнения определяется их приоритетами. Факторами ранжирования вариантов использования (и вообще всех требований) по приоритетам являются:

• существенное влияние на архитектуру системы;
• рискованные, сложные для реализации или срочные функции;
• применение новой, неапробированной технологии;
• значимость в экономических процессах.

Правила написания сценариев:

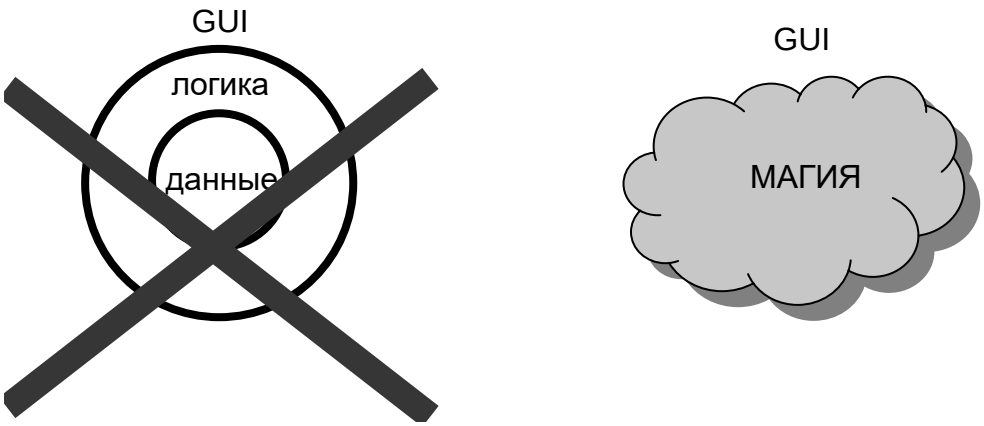
1) Используйте простые предложения: Подлежащее...сказуемое...прямое дополнение... предложный оборот (Система...удерживает...сумму...из остатка на счёте).
2) Ясно укажите, кто «владеет мячом». На каждом шаге одно из действующих лиц "владеет мячом" – сообщением и данными, которые одно действующее лицо передаёт другому.
3) Пишите, глядя на вариант использования с точки зрения пользователя, а не системы.
4) Не показывайте слишком незначительные, мелкие действия.

Виды альтернативных сценариев:

- Некорректное действие действующего лица (ввод неверного пароля).
- Бездействие основного действующего лица (истечение времени ожидания пароля).
- Предложение "система подтверждает" связано с обработкой неподтверждения (неверный учётный номер).
- Несоответствующая реакция второстепенного действующего лица или её отсутствие (истечение времени ожидания ответа).
- Внутренняя ошибка в разрабатываемой системе, которая должна быть обнаружена и обработана в обычном порядке (заблокирован автомат для выдачи наличных).

- Неожиданная и необычная ошибка, которую необходимо обработать (обнаружено повреждение журнала транзакций).
- Критически важные недостатки в производительности системы (время реакции не укладывается в 5 секунд).

Писать варианты использования следует с позиции пользователя, т. е.:



Типичные ошибки в сценариях:

- Отсутствует система.
- Отсутствует основное действующее лицо.
- Слишком много деталей пользовательского интерфейса.
- Слишком низкий (подробный) уровень описания.

Связи между вариантами использования и действующими лицами отображаются на диаграмме вариантов использования:

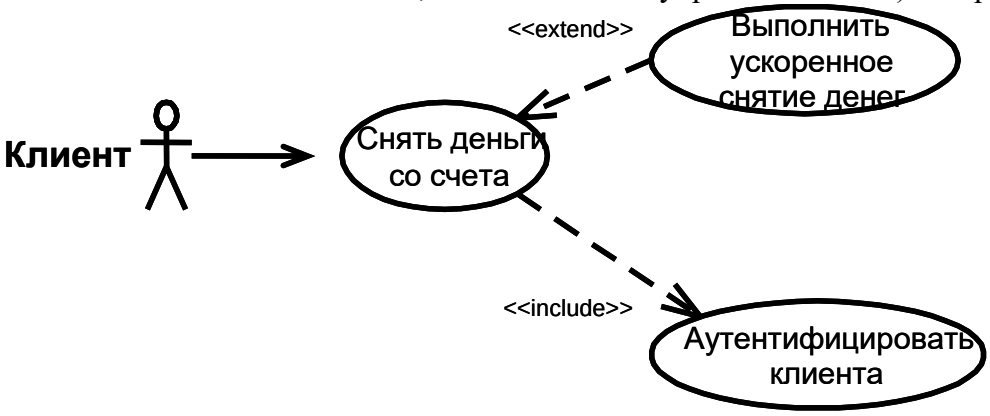
Правила составления этих диаграмм:

1) Каждый вариант использования должен быть инициирован действующим лицом.
2) Не моделируйте связи коммуникации между действующими лицами.
3) Не соединяйте стрелкой (связью коммуникации) два варианта использования непосредственно. Диаграммы данного типа описывают только, какие варианты использования доступны системе, а не порядок их выполнения. На диаграммах варианты использования могут быть связаны либо зависимостями (связями включения или расширения) или обобщением (наследованием). Для отображения порядка

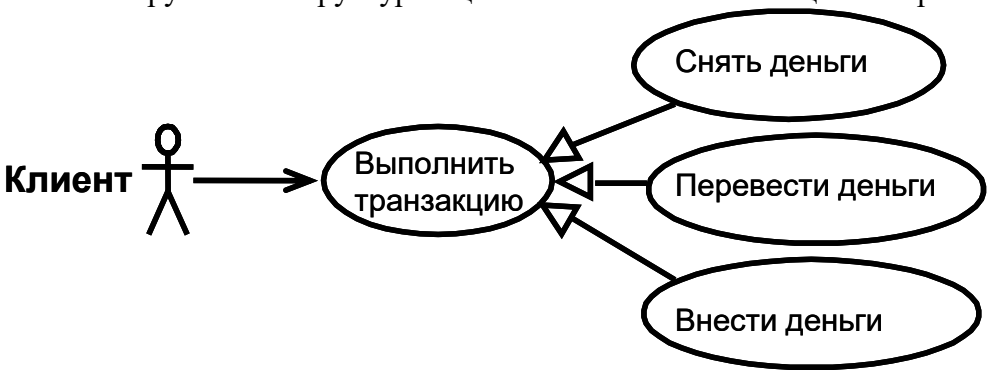
выполнения вариантов использования применяют диаграммы деятельности.

4) Избегайте многочисленных и запутанных связей между действующими лицами и вариантами использования.

Для снижения сложности начальная модель вариантов использования может быть подвергнута структуризации, в ходе которой выделяются вспомогательные варианты использования (включаемые или расширяющие). Сложность снижается за счет вынесения части описаний из основного варианта использования во включаемый (который может быть включен в несколько ВИ, что еще больше упрощает модель) или расширяющий.



Инструментом структуризации также является обобщение вариантов использования.



При обобщении в базовый ВИ выносится описание общее для всех наследников, дочерние ВИ специализируют описание базового, они участвуют во всех связях расширения и/или включения базового. Обобщение может быть использовано и для действующих лиц. В таком случае дочерние действующие лица наследуют связи коммуникации родительского действующего лица.

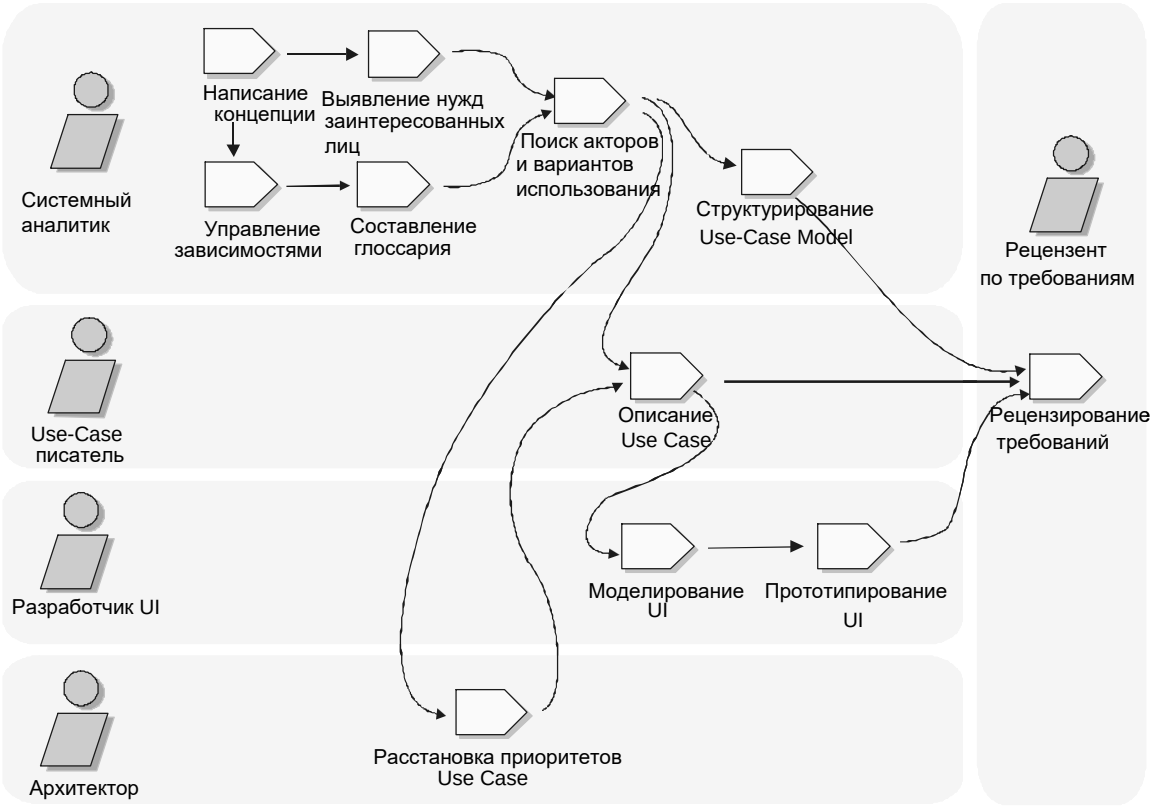
Методика моделирования вариантов использования в технологии Rational Unified Process предусматривает специальное соглашение, связанное с группировкой структурных элементов и диаграмм модели. Это соглашение включает следующие правила:

• Все действующие лица, варианты использования и диаграммы вариантов использования помещаются в пакет с именем Use Case Model.

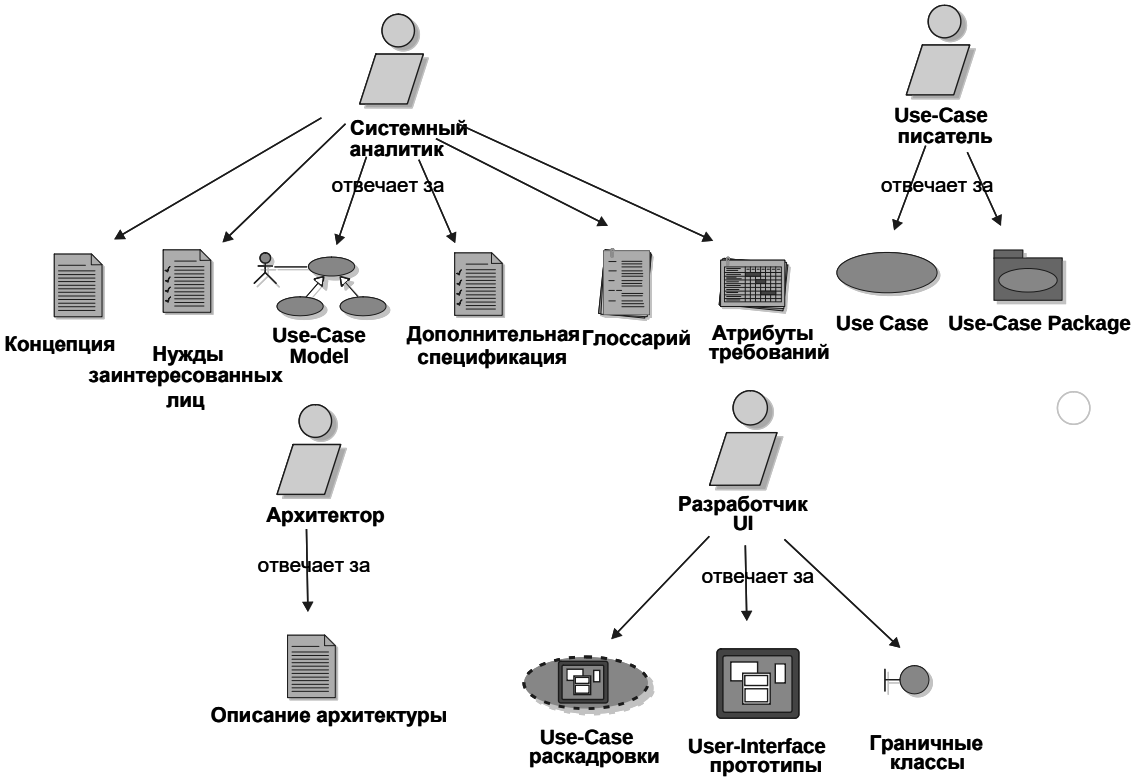
• Если моделируется сложная многофункциональная система, то совокупность всех действующих лиц и вариантов использования может разделяться на пакеты. В качестве принципов разделения могут использоваться:

- структуризации модели в соответствии с типами пользователей (действующих лиц);
- функциональная декомпозиция;
- разделение модели на пакеты между группами разработчиков (в качестве объектов управления конфигурацией).

Дисциплина определения требований в рамках RUP описывается следующим набором ролей и деятельностей:



Наборы рабочих продуктов определения требований:

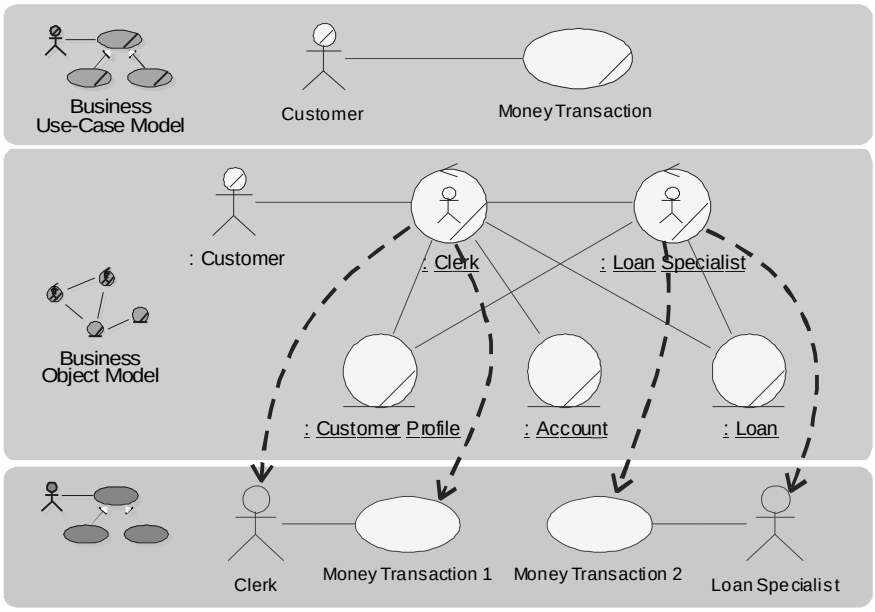


Спецификация требований в технологии Rational Unified Process не требует обязательного моделирования бизнес-процессов организации, для которых создается ПО, однако, наличие бизнес-моделей существенно упрощает построение системной модели вариантов использования. При переходе от бизнес-модели к начальной версии модели вариантов использования применяются следующие правила:

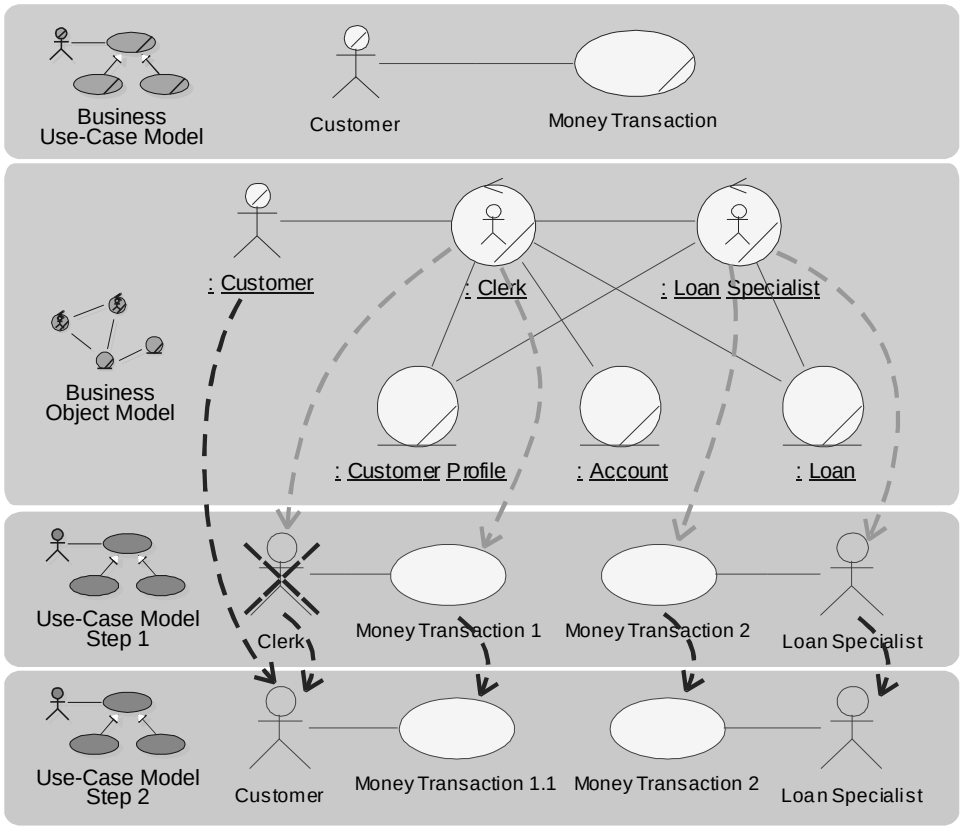
• Для каждого исполнителя в модели бизнес-анализа, который в перспективе станет пользователем новой системы, в модели вариантов использования создается действующее

лицо с таким же наименованием. В состав действующих лиц включаются также внешние системы, играющие в бизнес-процессах пассивную роль источников информации.

• Варианты использования для данного действующего лица создаются на основе анализа обязанностей соответствующего исполнителя (в простейшем случае для каждой операции исполнителя создается вариант использования, реализующий данную операцию в системе).



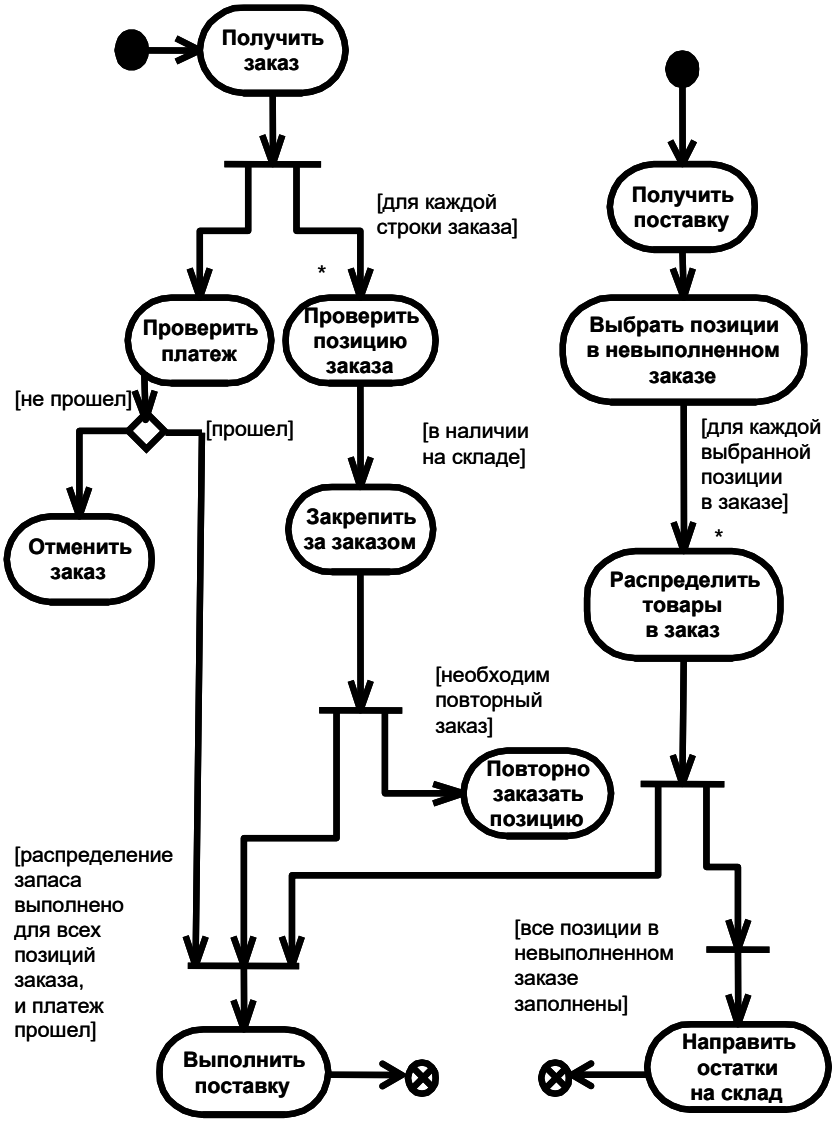
Такая начальная версия модели описывает минимальный вариант системы, пользователями которой являются только исполнители бизнес-процессов. Если в дальнейшем, в процессе развития системы, ее непосредственными пользователями будут становиться действующие лица бизнес-процессов, то модель вариантов использования будет соответствующим образом модифицироваться.



Для описания функциональных требований используются диаграммы деятельности:

- для описания поведения, включающего большое количество параллельных процессов
- для анализа варианта использования (описывают последовательность действий и их взаимосвязь)
- для анализа потоков работ (workflow) в различных вариантах использования. Когда варианты использования взаимодействуют друг с другом, диаграммы деятельности являются средством представления и анализа их поведения.

Пример диаграммы деятельности для двух синхронных потоков:



Модель вариантов использования можно считать завершенной, если есть утвердительный ответ на следующие вопросы:

- Можно ли на основании модели сформировать четкое представление о функциях системы и их взаимосвязях?
- Присутствует ли каждое функциональное требование хотя бы в одном варианте использования? Если требование не нашло отражение в варианте использования, оно не будет реализовано.
- Учли ли вы, как с системой будет работать каждое заинтересованное лицо?
- Какую информацию каждое заинтересованное лицо будет передавать системе?
- Учли ли вы все внешние системы, с которыми будет взаимодействовать данная?