



Начальная фаза проекта создания
(модернизации) ИС в RUP,
формирование и анализ требований

д.э.н., проф. Тельнов Ю.Ф.

Вопросы

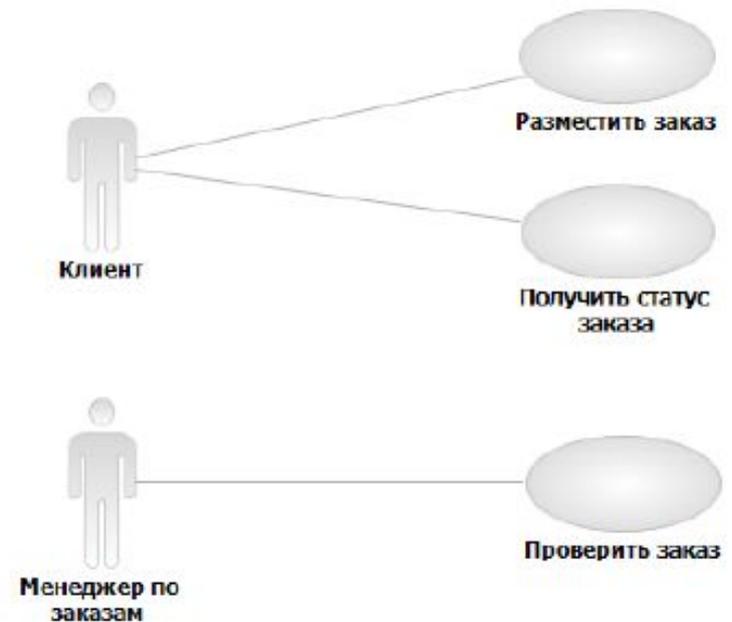
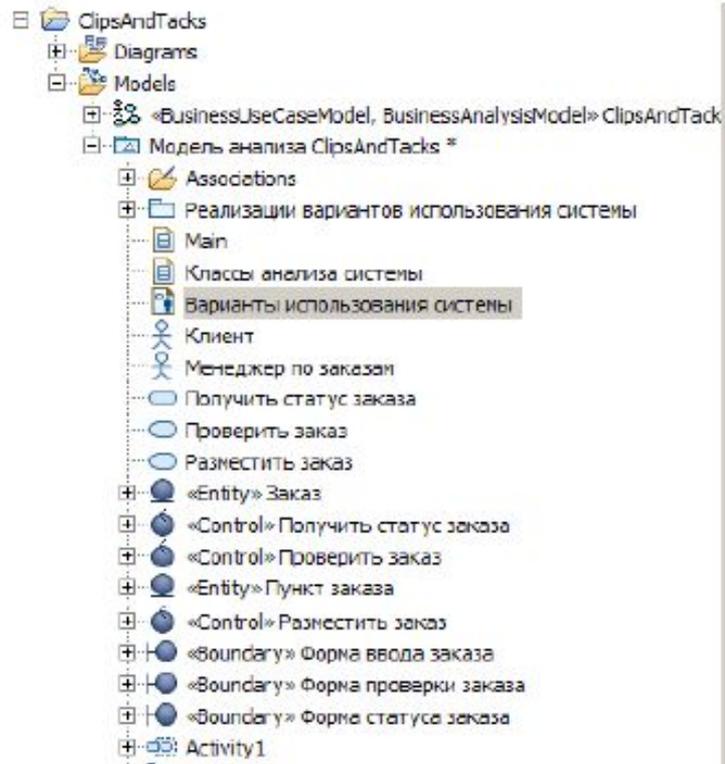
1. Цели (задачи) начальной фазы проекта
2. Роль аналитика на начальной фазе формирования требований
3. Моделирование вариантов (прецедентов) использования – функциональных требований в RSA

1. Цели (задачи) начальной фазы жизненного цикла (Inception)

1. Понять, что создавать - определить **общее описание** или **границы (рамки) проекта**, установить концепцию, цели и требования, назначение (для кого)
2. Определить **ключевые функции системы** – определить варианты (прецеденты) использования
3. Выявить хотя бы одно возможное **архитектурное решение** (одну возможную архитектуру), добиться соглашения между заинтересованными сторонами для дальнейшего продвижения
4. Оценить **стоимость, сроки и риски**, связанные с проектом - разработать экономическое обоснование, минимизировать риски
5. **Настройка процесса разработки** (настройка RUP)- решить какому процессу следовать и какие средства использовать

Веха целей жизненного цикла (LCO – Lifecycle Objective)

Диаграмма вариантов (прецедентов) использования



Приблизительный перечень артефактов начальной фазы (Крэг Ларман)

Артефакт	Пояснение
Видение и финансовые оценки проекта (vision)	Описываются общие задачи и ограничения, оценивается стоимость проекта и приводится заключение, в частности, купить или разработать
Модель прецедентов (Use-Case Model)	Описываются функциональные требования . Определяется перечень основных прецедентов использования, 10-20% детально описываются
Дополнительная спецификация (Supplementary Specification)	Описываются нефункциональные требования
Бизнес-правила (business rules)	Требования или политики, выходящие за пределы конкретного проекта (законодательство, регламенты, нормативы)
Словарь терминов (Glossary)	Ключевая терминология предметной области
Перечень рисков и план управления ими	Описываются экономические, технические и организационные риски, а также методы их преодоления
Прототипы и обоснование идеи	Разрабатываются для лучшего осмысления проекта и оценки технических идей (в основном прототипы интерфейсов)
План итерации	Действия по выполнению первой итерации фазы проектирования
План фазы проектирования	Описание использования ресурсов по этапам проектирования
Набор инструментов	Описание артефактов (разрабатываемых результатов) по этапам проектирования

I. Определить общее описание (границы) проекта

- Согласовать высокоуровневую концепцию:
 - Возможности и преимущества приложения (системы)
 - Решаемые проблемы
 - Конечные пользователи
 - Перечень функций высокого уровня (обзор ключевых прецедентов)
 - Наиболее существенные нефункциональные требования (ОС, СУБД, надежность, масштабируемость, качество, лицензирование)
- Сформировать широкое, но не глубокое описание системы
 - Определение и краткое описание акторов (пользователей и внешних систем и их информационных потребностей)
 - Определение и краткое описание вариантов использования (типичные случаи использования системы, совокупность типичных взаимодействий составляет прецедент использования – Use-Case)
 - Выделение наиболее критичных прецедентов использования (10-20%) – более детальное описание, остальные – один, два абзаца
- Подробное описание ключевых акторов и вариантов использования
 - Описание прецедентов использования (детальное описание основного потока и выделение альтернативных потоков событий) – 10-20 %
 - Прототипы пользовательского интерфейса

2. Определить ключевые функции системы по критериям:

- Функциональность является ключевой для приложения или использует ключевые интерфейсы (анализ рисков, производительности, безопасности системы) – определяет архитектор приложения и менеджер
- Функциональность обязательно должна быть реализована, компонент, без которого не будет работать приложение – определяет архитектор приложения и менеджер
- Функциональность, некритичная с позиции менеджмента проектом – определяет менеджер проекта

Выбор прецедентов

А) Из 15 – 4 – для малой системы

В) Из 40 – 9 – для большой системы

С) При модернизации – 1 из имеющихся, 2 из 9

3. Архитектурное решение

Принятие решения о возможности реализации системы на основе обоснования проектных решений в рамках хотя бы одной архитектуры. Вопросы при выборе архитектурных решений:

- Наличие сходных систем. Какие технологии и архитектуры применяются в них?
- Анализ существующей архитектуры, обоснование необходимости ее развития
- Обоснование выбора новых технологий по затратам и рискам
- Обоснование выбора программных компонентов (СУБД, ПО промежуточного слоя,...) по затратам и рискам

А) 1 функциональный прецедент

В) 2 функциональный прецедента

С) 2 функциональных прецедента

Демонстрация макетов заказчикам

4. Оценка стоимости, сроков и рисков проекта

- Экономическое обоснование проекта – основание для адекватного финансирования проекта (совокупная стоимость владения, ROI, риски)

5. Настройка процесса разработки (настройка RUP) - решить какому процессу следовать и какие средства использовать

Рецензирование проекта. Веха целей жизненного цикла –

оценка:

- Согласие заинтересованных сторон по вопросу определения границ проекта и первоначальной его стоимости/сроков
- Согласие по набору требований и существует общая точка зрения на эти требования
- Согласие в оценке стоимости/сроков, приоритетах, рисках и процессе разработки и стратегии борьбы с рисками приемлемы

2. Роли участников проекта создания (модернизации ИС)

Требования	 Аналитик  Заинтересованные лица
Архитектура	 Архитектор
Разработка	 Разработчик
Тестирование	 Тестирующий
Управление проектом	 Руководитель проекта
Управление конфигурацией и изменениями	 Разработчик

Роль аналитика на начальной фазе формирования требований

- Понимание потребностей пользователей и других заинтересованных лиц
- Документирование и ранжирование требований
- Согласование требований со всеми заинтересованными лицами: пользователями, архитекторами, разработчиками, менеджерами проектов

Требования к системному аналитику

- Уметь взаимодействовать с заинтересованными лицами
- Хорошо понимать область проблемы и быть способным быстро приобретать знания
- Способность ясно и четко излагать свои мысли как письменно, так и устно
- Способность моделировать бизнес-процессы и формулировать требования
- Понимать место работы аналитика в жизненном цикле ИС

От требований к архитектуре



Список активностей (задач) системного аналитика

- Понимание бизнес-процессов – моделирование бизнес-процессов
- Определение потребностей заинтересованных лиц – работа в рабочих группах (очередность задачи, важнейшие нефункциональные требования)
- Создание концепции
 - Список заинтересованных лиц
 - Ограничения: бюджет, технологии, ОС-ы, совместимость с другими системами
 - Формулировка проблемы
 - Список свойств системы

Формулировка проблемы

- Описание проблемы
- Затрагивает <заинтересованные лица>, например, заказчики
- Проблема приводит к <недостатки>
- Успешное решение проблемы <выгоды>

Свойство, характеристика разрабатываемого продукта

- Описание
- Значимость
- Стоимость
- Приоритет разработки

Разработка модели прецедентов и глоссария – функциональных требований

1. Выявить акторов – проанализируйте свойства и запросы заинтересованных лиц. Описания – одна фраза на актор.
2. Выявить прецеденты (варианты) использования – взаимодействия акторов с системой и их группировка (активности/функции в бизнес-процессах)
3. Определение взаимодействия прецедента с другими пользователями или системами (акторами)
4. Описание акторов (по абзацу) и прецедентов (по нескольким абзацам). Может потребовать пересмотр состава прецедентов
5. Создание глоссария (объектная терминология)
6. Проверка глоссария – все ли объекты используются в прецедентах
7. Выявление критичных прецедентов использования (20% от общего числа)
8. Для наиболее критичных прецедентов подробное сценарное описание
9. **На стадии проектирования** – подробное описание всех остальных прецедентов использования и составление глоссария
10. Дополнительные связи наследования между акторами и включения, расширения и обобщения для прецедентов

Фиксация нефункциональных требований

- Атрибуты качества (удобство, надежность, производительность, поддержка и др.)
- Юридические и инструктивные требования и стандарты
- Прочие требования, требования к ОС, окружению, совместимости, протоколам и др.

Разработка прототипов пользовательского интерфейса

1. Концептуальный прототип – визуализация и демонстрация ключевых возможностей, не связанных с конкретными прецедентами использования
2. Прототипы прецедентов использования или раскадровки прецедентов использования (основные экраны, иллюстрирующие функциональность) - скриншоты

Обновление и уточнение требований

- Оценка рисков, сроков, стоимости
- Система управления изменениями совместно с менеджером проекта и группой управления изменениями

Обеспечить выпуск и тестирование требований – программы, документация, учебные материалы

3. Моделирование прецедентов (вариантов) использования – функциональных требований в RSA

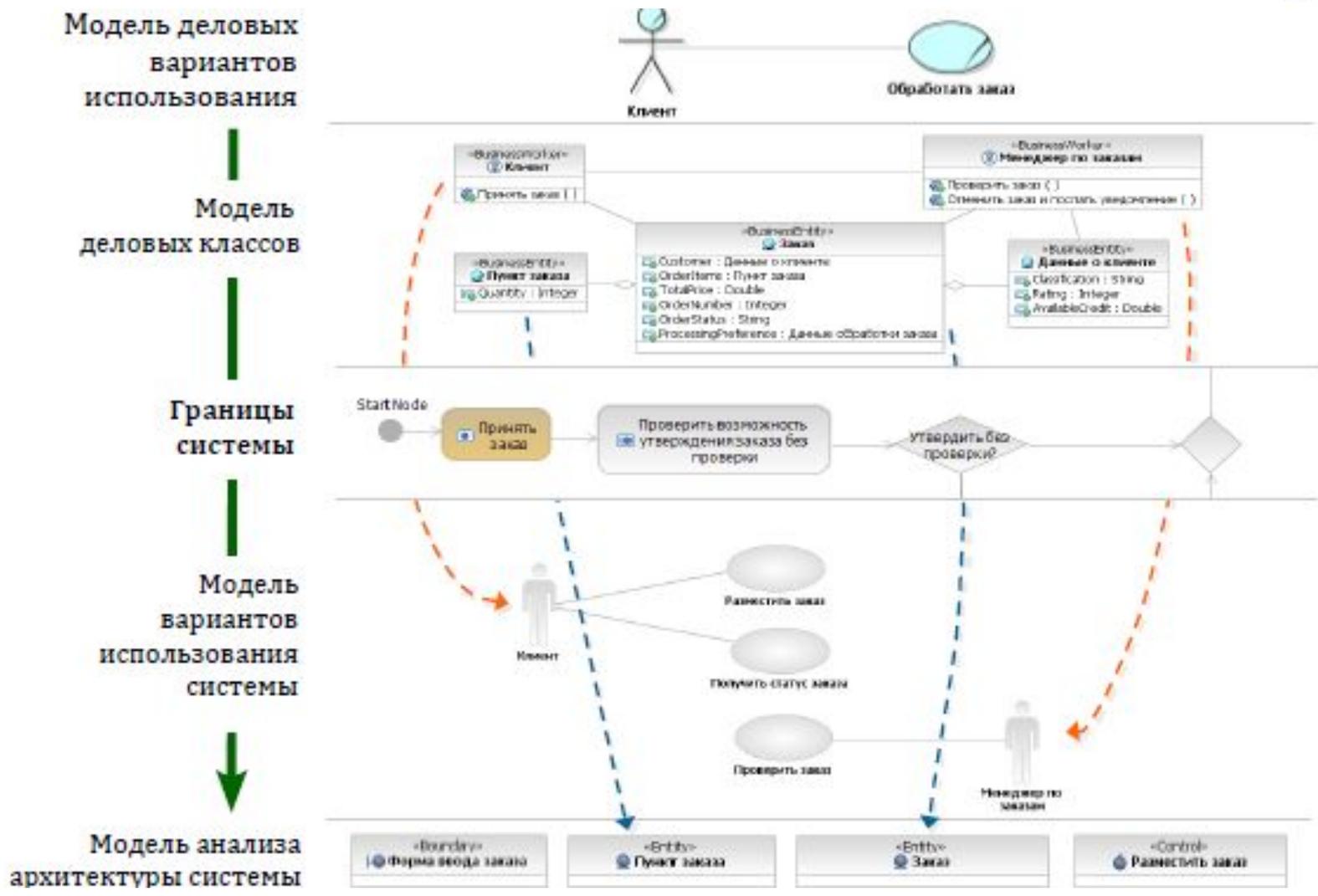


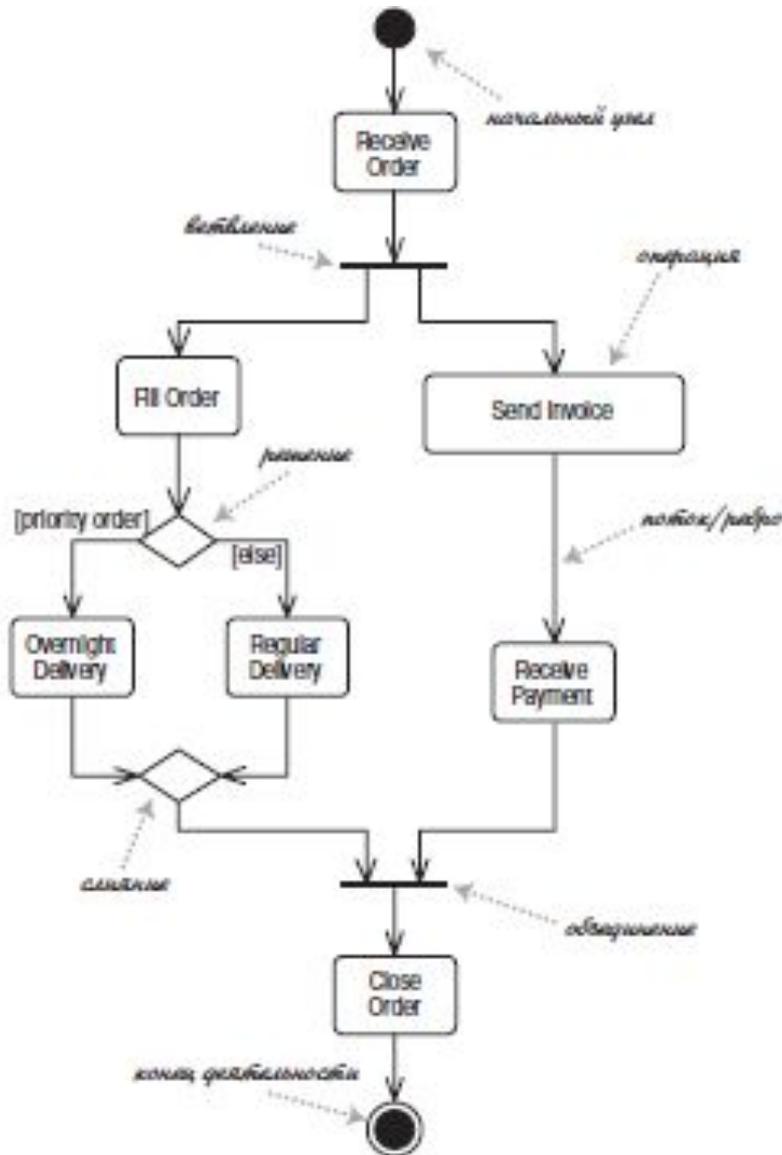
Диаграмма деятельности (activity – активностей)

- Диаграммы деятельности – это технология, позволяющая описывать логику процедур, бизнес-процессы и потоки работ.
- Во многих случаях DA напоминают блок-схемы, но принципиальная разница заключается в том, что они поддерживают параллельные процессы.

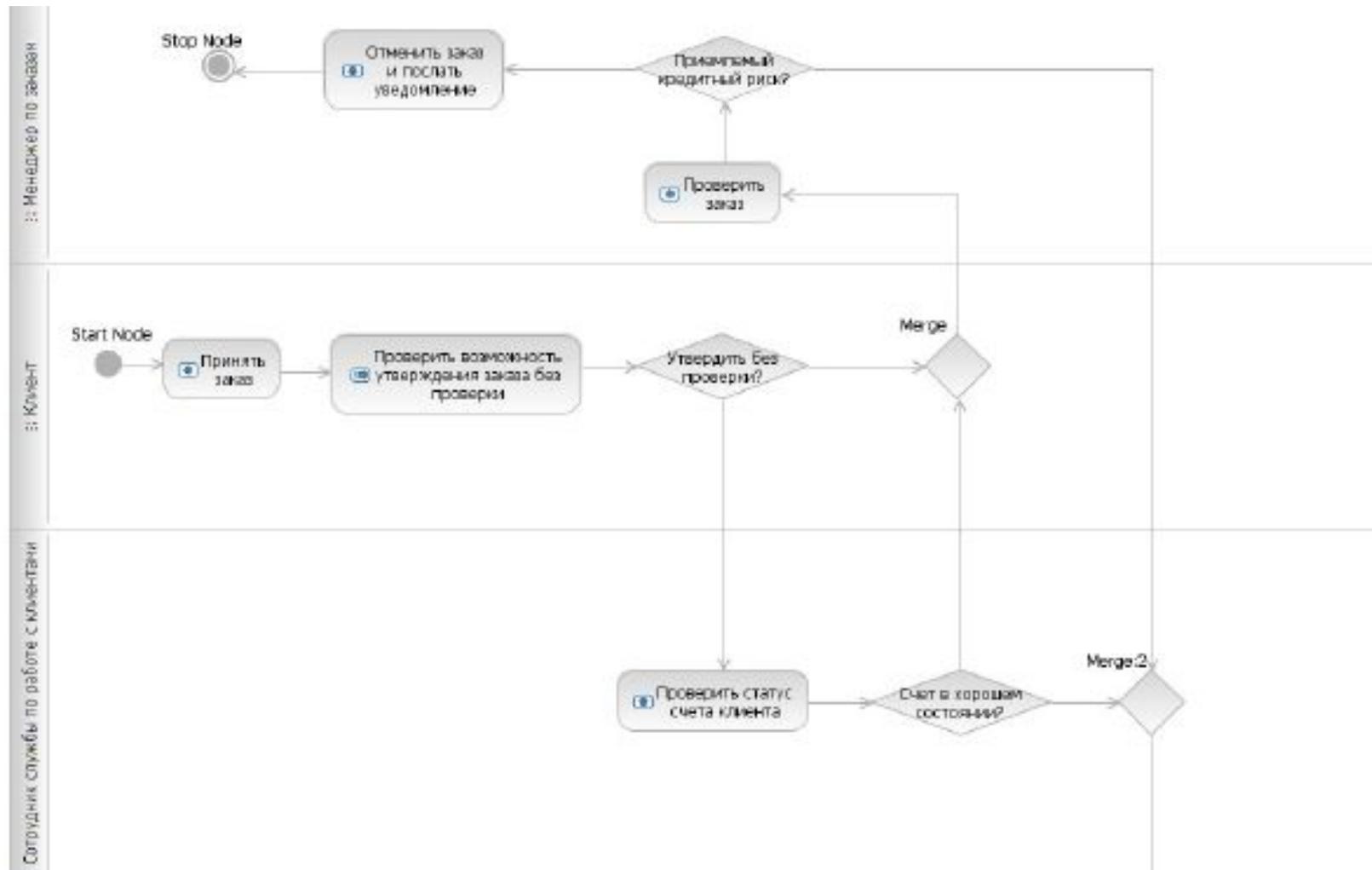
Типы элементов

- Действие – action
- Управляющий поток – control flow
- Принятие решений – Decision (Xor)
- Разделение потока - Fork (And)
- Объединение потоков – Join (And)
- Слияние потоков – Merge (Or)
- Начальная и конечная точка (Initial, End)
- Разбиение деятельности (Partition)

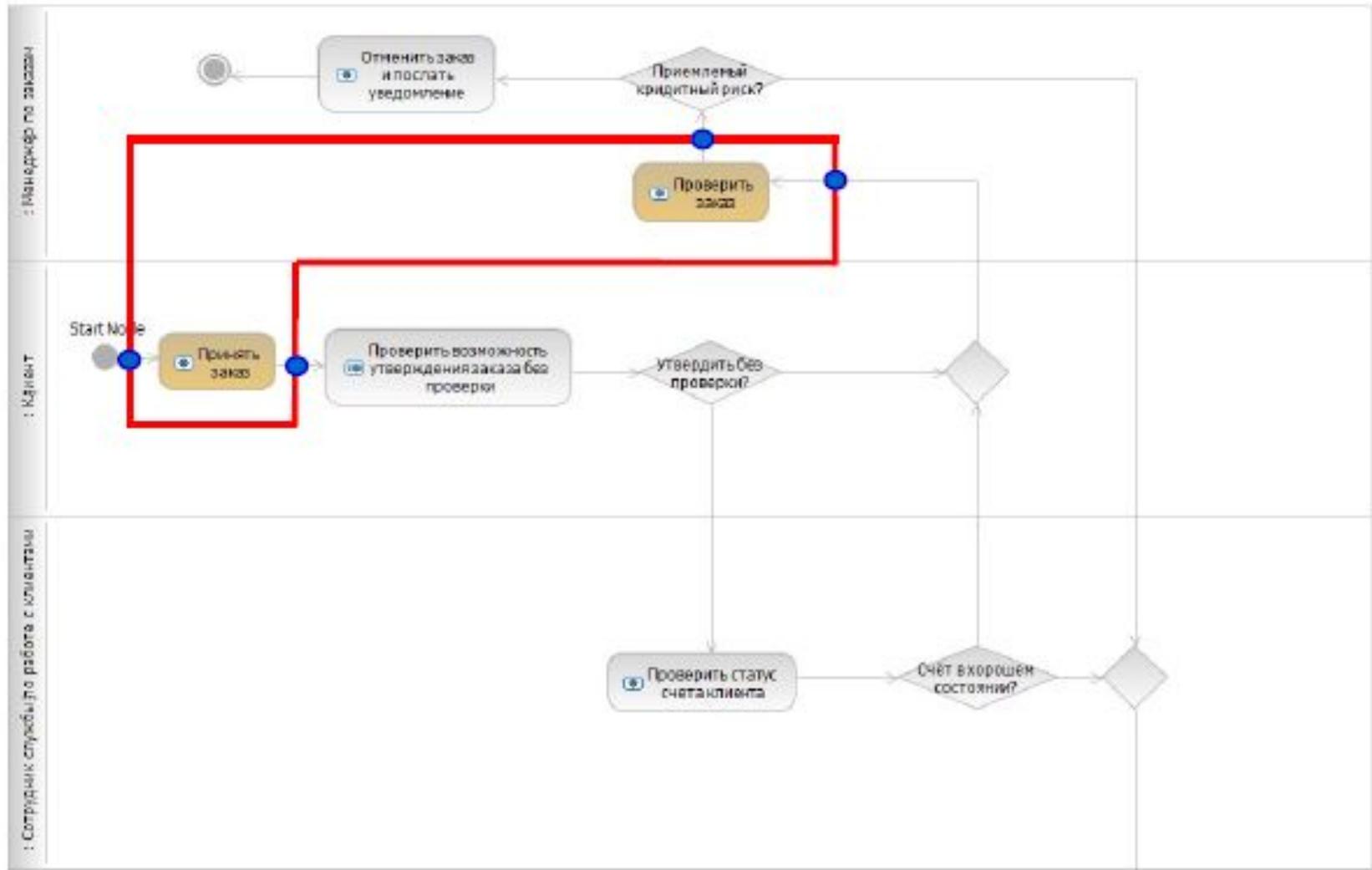
Пример диаграммы деятельности



Пример диаграммы активностей



Бизнес-процесс – границы системы



Модель прецедентов (вариантов) ИСПОЛЬЗОВАНИЯ

Определение

Модель вариантов использования — это абстрактное представление взаимодействия системы с её окружением: пользователями, внешними системами и устройствами.

■ Основные элементы модели:

- Акторы (от англ. *actor* — субъект, действующее лицо)
- Варианты использования
- Отношения между акторами и вариантами использования

■ Дополнительные элементы модели:

- Пакеты вариантов использования
- Зависимости (отношения типа `<<include>>` и `<<extend>>`)

**Управление
сложностью**

Роль модели прецедентов (вариантов) использования

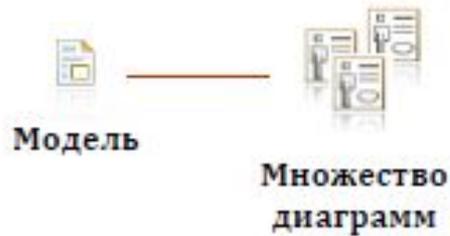
Требования – это весь набор прецедентов, т.е. модель функционирования системы и её окружения.

- Модель прецедентов выступает в качестве объединяющего начала всего проекта
 - Основная спецификация функциональных требований к системе
 - Основа для анализа и проектирования
 - Вносит вклад в итерационное планирование
 - Основа определения сценариев тестирования системы
 - Основа для пользовательской документации

Диаграмма вариантов использования

- Диаграмма вариантов использования применяется для визуализации подмножества модели вариантов использования с целью упрощения коммуникаций по ходу проекта

Один элемент модели может быть отображен на нескольких диаграммах



Графические элементы диаграммы



Характеристики акторов

- Актор (исполнитель роли) – это сущность, обладающая поведением (человек, подразделение, внешняя компьютерная система)
- Основной актор – его задачи связаны с использованием системы (прецедента), инициирует работу системы, задачи определяют названия прецедентов
- Вспомогательный актор – обслуживает систему, выполняет сервисные функции, воспринимает запросы от прецедента (варианта) использования, предоставляет информацию

Описание актора

- Каждый актор должен иметь свое краткое описание
 - Кого или что представляет актор ?
 - Какие свои цели актор достигает с помощью системы?
 - Основные характеристики актора.
- Примеры важных характеристик актора:
 - Уровень знаний актора о предметной области
 - Уровень навыков актора при работе с компьютером
 - Родной язык актора
 - Особые отметки об ограниченных возможностях при работе с компьютером

Поиск акторов

- Кандидатами на роль акторов являются:
 - Пользователи, которые будут работать с системой при выполнении своих задач
 - Внешнее оборудование, программное обеспечение и другие системы
- Присвойте каждому такому кандидату имя и дайте краткое описание
- Опишите область ответственности актора и цели, которые актор будет стремиться достичь при работе с системой



Исключите акторов, которые не имеют каких-либо целей

- ✓ Эти вопросы помогут выявить акторов:
 - Кто будет использовать систему?
 - Кто будет поставлять, использовать или удалять информацию из системы?
 - Кто заинтересован в определённых функциях или услугах системы?
 - Кто будет поддерживать и обслуживать систему?
 - Что будет внешними ресурсами системы?
 - Каким другим системам необходимо будет взаимодействовать с системой?

Вариант (прецедент) использования – USE-CASE

Понятие варианта использования

Определение

Вариант использования описывает взаимодействие между одним или несколькими акторами и системой, ориентированное на достижение видимого результата, имеющего ценность для актора, инициировавшего такое взаимодействие.

- Функциональность системы определяется различными вариантами её использования
- Каждый вариант использования ассоциирован с целью одного из акторов



Цель актора должна читаться в названии варианта использования

- Коллекция вариантов использования составляет возможные пути использования системы

Поиск вариантов использования

■ Для каждого актора спросите себя:

- Какие цели актер будет стремиться достичь с помощью системы?

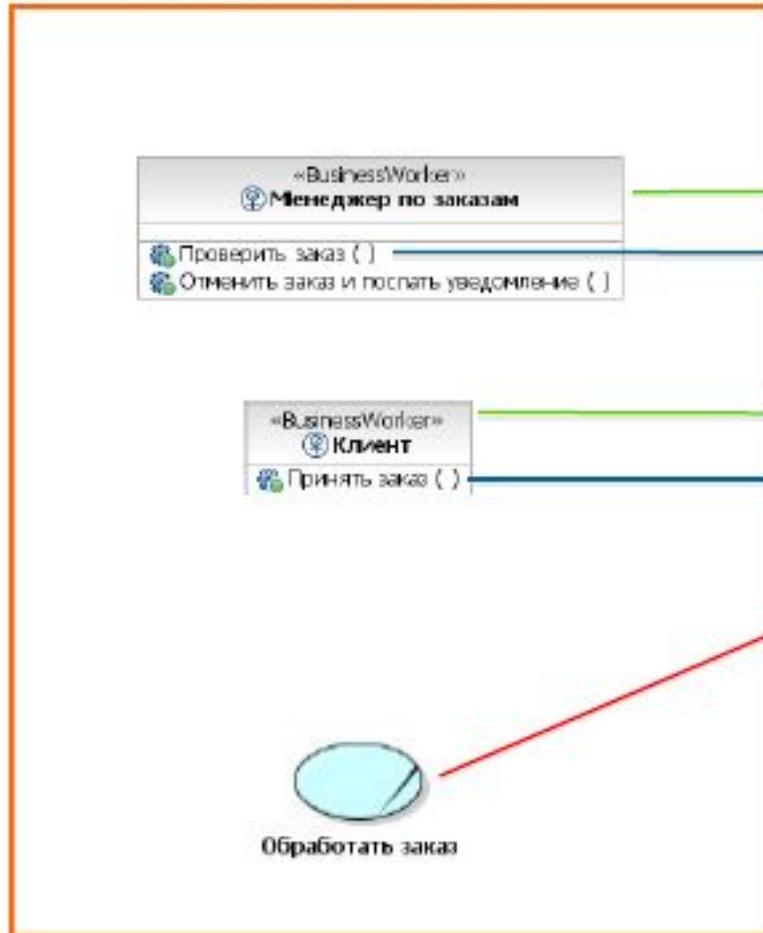


Если цели нет, то либо её нужно указать, либо исключить актора из модели

- Какие свои основные задачи актер хочет решить с помощью системы?
- Будет ли актер создавать, хранить, менять, удалять или читать данные в системе?
- Нужно ли будет актору информировать систему о каких-либо событиях, случившихся вне системы?
- Нужно ли будет системе информировать актора об определённых событиях, произошедших внутри системы? Например, о потере соединения.
- Будет ли актер осуществлять запуск или остановку системы?

Отображение бизнес-модели в модель ПО

Элементы бизнес-модели



Элементы модели ПО

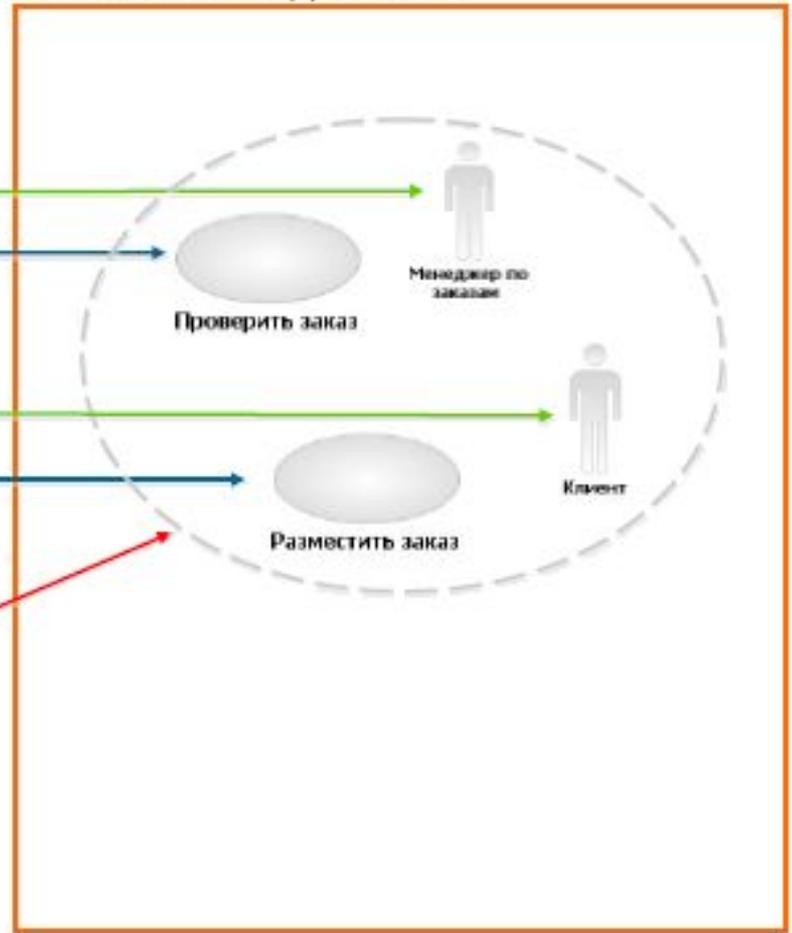
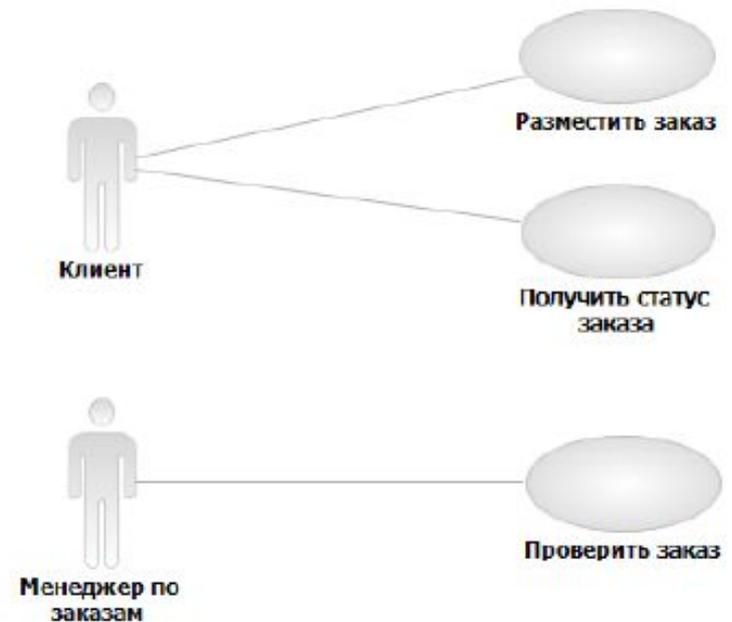
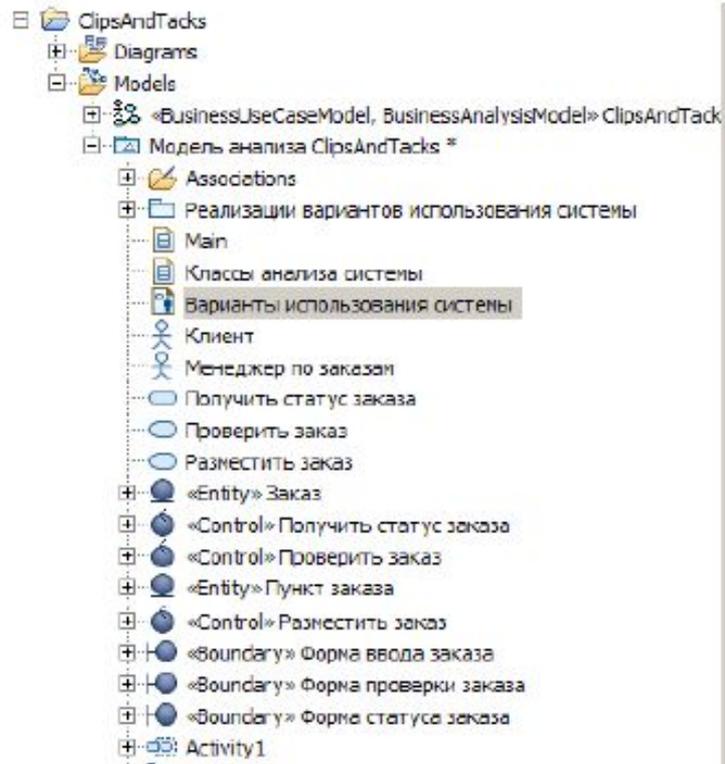


Диаграмма вариантов (прецедентов) использования



Дополнительные элементы модели вариантов использования

Пакет вариантов использования

- Используется для структурирования модели вариантов использования с целью упрощения анализа, коммуникации, навигации и планирования
- Примеры пакетов:

Делим задачу на подзадачи



Варианты использования для задачи обработки заказов

Упрощаем взаимодействие с заинтересованными лицами

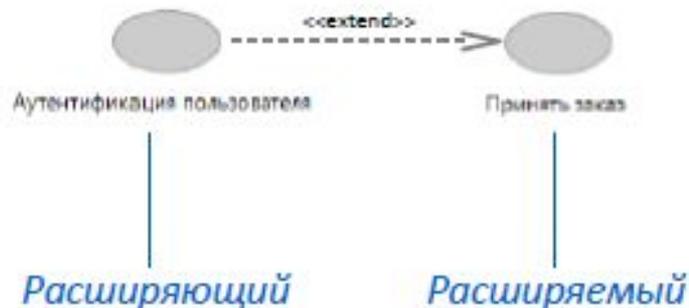


Дополнительные элементы модели вариантов использования

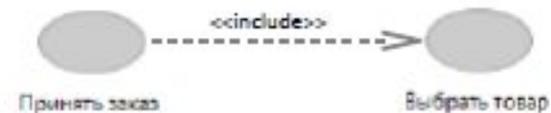
Зависимости

- Поддерживают повторное использование общего поведения
 - Тип зависимости `<<extend>>` используется, чтобы включить дополнительное поведение от расширяющего варианта использования к расширяемому
 - Тип зависимости `<<include>>` используется, чтобы включить в вариант использования общие для нескольких вариантов использования поведение

Пример `<<extend>>`

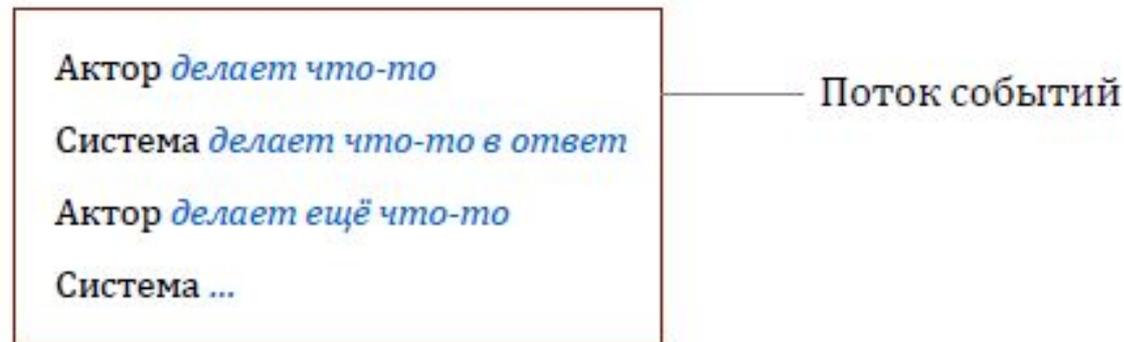


Пример `<<include>>`



Структура диалога актора и системы

- Вариант использования описывает взаимодействие между акторами и системой в форме диалога, имеющего следующую структуру:



Пример потока событий *

Основной поток событий:

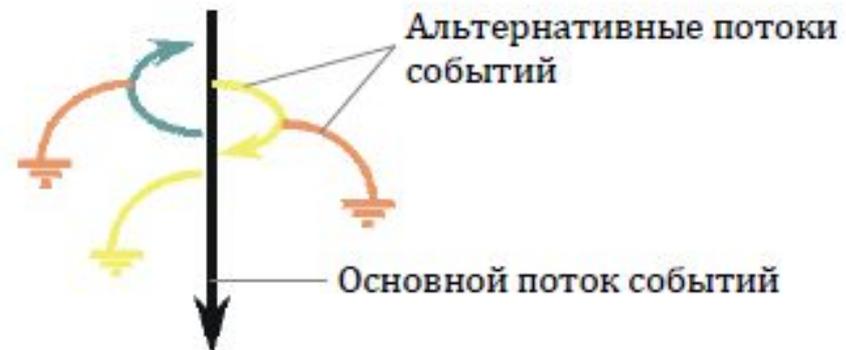
1. Клиент *открывает каталог товаров*
2. Система *предлагает выбрать товар*
3. Клиент *указывает товар из предложенного списка*
4. Система *принимает выбор и предлагает указать количество*
5. Клиент *указывает количество товара*
6. Система *принимает количество товара и выводит сумму*
7. Клиент *подтверждает заказ*
8. Система *сохраняет заказ клиента в базе заказов*

Понятие потока событий

Сценарий или поток событий - последовательность действий или взаимодействий между актерами и системой

- Каждый вариант использования включает в себя **один** *основной поток событий* и **множество** *альтернативных потоков событий*
- Основной поток событий:
 - Описывает взаимодействие между актерами и системой с положительным исходом, когда актер достигает своей цели
 - Представляет собой базовое поведение системы для данного варианта использования
- Альтернативные потоки событий:
 - Описывают дополнительные способы взаимодействия актора с системой, для которых возможен отрицательный исход

Может быть упрощенный или усложненный поток событий получения положительного результата



Пример потоков событий

Основной поток событий:

1. Клиент открывает каталог товаров
2. Система предлагает выбрать товар
3. Клиент указывает товар из предложенного списка
4. Система принимает выбор и предлагает указать количество
5. Клиент указывает количество товара
6. Система принимает количество товара и выводит сумму
7. Клиент подтверждает заказ
8. Система сохраняет заказ клиента в базе заказов

Альтернативный поток событий:

- 7а. Клиент не подтверждает заказ
- 7а.1. Клиент отклоняет заказанный товар
 - 7а.2. Система предлагает выбрать другой товар

Проект интерфейса пользователя

- Варианты использования помогают проектировать интерфейс пользователя
 - Эскизы интерфейса пользователя
 - Раскадровки (последовательности эскизов)
- Поток событий между актором и системой могут быть визуализированы с использованием диаграмм последовательности UML

Пример потока событий

Основной поток событий:

1. Клиент открывает каталог товаров
2. Система предлагает выбрать товар
3. Клиент указывает товар из предложенного списка
4. Система принимает выбор и предлагает указать количество
5. Клиент указывает количество товара
6. Система принимает количество товара и выводит сумму
7. Клиент подтверждает заказ
8. Система сохраняет заказ клиента в базе заказов

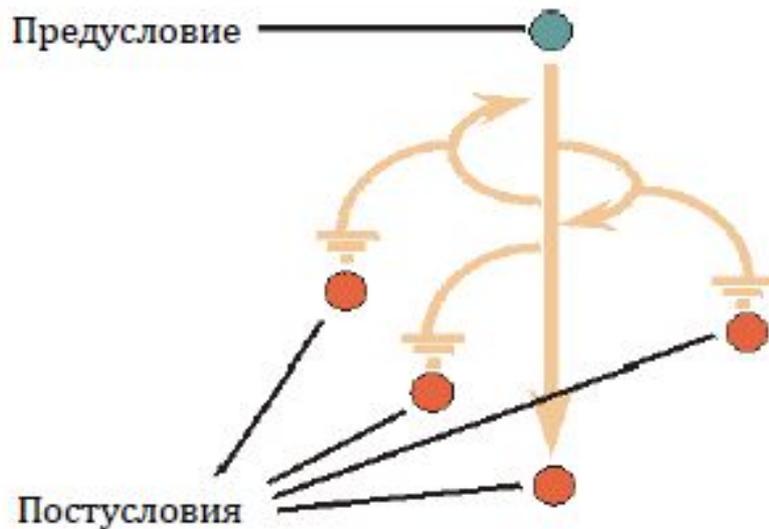
Визуализация потока событий



Предусловия и постусловия

Определение

Предусловие — это состояние системы и её окружения, требуемое для того, чтобы вариант использования стартовал. **Постусловия** — это состояния системы, которые могут быть по завершении варианта использования.



Пример предусловия:

Клиент должен быть зарегистрированным в системе.

Пример постусловия:

По завершении варианта использования, журнал транзакций сбалансирован со счетами клиента.

Шаблон полного описания прецедента

Раздел описания прецедента	Комментарий
Название прецедента	Осмысленное название основной функции основного актора, например, оформление заказа клиента
Рамки	Название системы (подсистемы), например, п.с. «Выполнение заказов клиентов»
Уровень	« <u>Задача, определенная пользователем</u> » или вспомогательная функция»,
Основной актер	Лицо, инициирующее и реализующее работу сценария, например, Менеджер клиентов
Заинтересованные лица	Кто заинтересован в реализации этого сценария и чего они хотят, например, Клиенты и их цели
Предусловие	Условия успешной реализации сценария, например, наличие товара на складе, кредитоспособность клиента
Основной успешный поток событий (сценарий)	Типичный сценарий, например, обслуживание кредитоспособного клиента при наличии товара на складе - пошаговое описание действий акторов и системы
Альтернативные потоки событий (расширения сценария)	Альтернативные успешные или неуспешные сценарии, например, отказ от оформления заказа, дополнительная закупка товара, оформления кредита клиенту пошаговое описание действий акторов и системы при невыполнении условия основного сценария
Постусловия (результаты)	Задается для каждого основного и альтернативного сценариев. Задается конечное состояние и условие перехода на следующее действие. Например, заказ оформлен , переход к прецеденту «Отгрузка товара» или заказ не оформлен , выход из системы, или дополнительная закупка завершена , переход на пункт основного сценария.

Шаблон полного описания прецедента

Раздел описания прецедента	Комментарий
Специальные требования	Нефункциональные требования, связанные с прецедентом. Например, отклик системы на авторизацию в 90 % случаев осуществляется в течение 30 секунд
Список технологий и типов данных	Методы ввода-вывода и форматы данных, например, ввод с использованием клавиатуры или сенсорного экрана
Частота использования	Определяет периодичность использования, например, ежедневно с 9-00 до 18-00
Открытые вопросы	Вопросы, не решаемые реализацией данного прецедента. Например, предоставление кредита клиенту осуществляется другим прецедентом использования

Декомпозиция функциональных требований (бизнес-функций) на функции компьютерной обработки данных в сценарии прецедента

Функциональное требование (бизнес-функция)	Функция компьютерной обработки данных
Прием документа (заявки, требования,)	Заполнение экранной формы (ввод данных)
	Контроль (Проверка) данных
	Запись в базу данных
	Вывод сообщений об ошибках
Формирование документа (договора, заказа, платежного документа, счета)	Выбор первичного документа (например, заявки)
	Отбор нормативно-справочной информации (например, ценник)
	Расчет показателей (стоимость=количество*цена)
	Запись документа в БД (распечатка, вывод на экран)

Декомпозиция функциональных требований (бизнес-функций) на функции компьютерной обработки данных

Функциональное требование (бизнес-функция)	Функция компьютерной обработки данных
Формирование отчета	Выбор ключевых признаков из справочников
	Группировка (сортировка) по ключевым признакам
	Подсчет итогов
	Формирование итогового отчета
	Запись в базу данных*
	Печать (вывод на экран)