

# Use Cases

# План встречи

1. Что такое Use Case'ы?
2. Структура UC
3. Методы выявления UC
4. Синтез карты UC
5. Инструменты, литература, практика

# 1

Что такое Use Case'ы?

# Что такое требование? (1)

**Требование (requirement)** – это зафиксированное желание пользователя, которое должна выполнять система

**Требование** – желательное свойство, характеристика, условие или ограничение, которым должна удовлетворять система в процессе своей эксплуатации

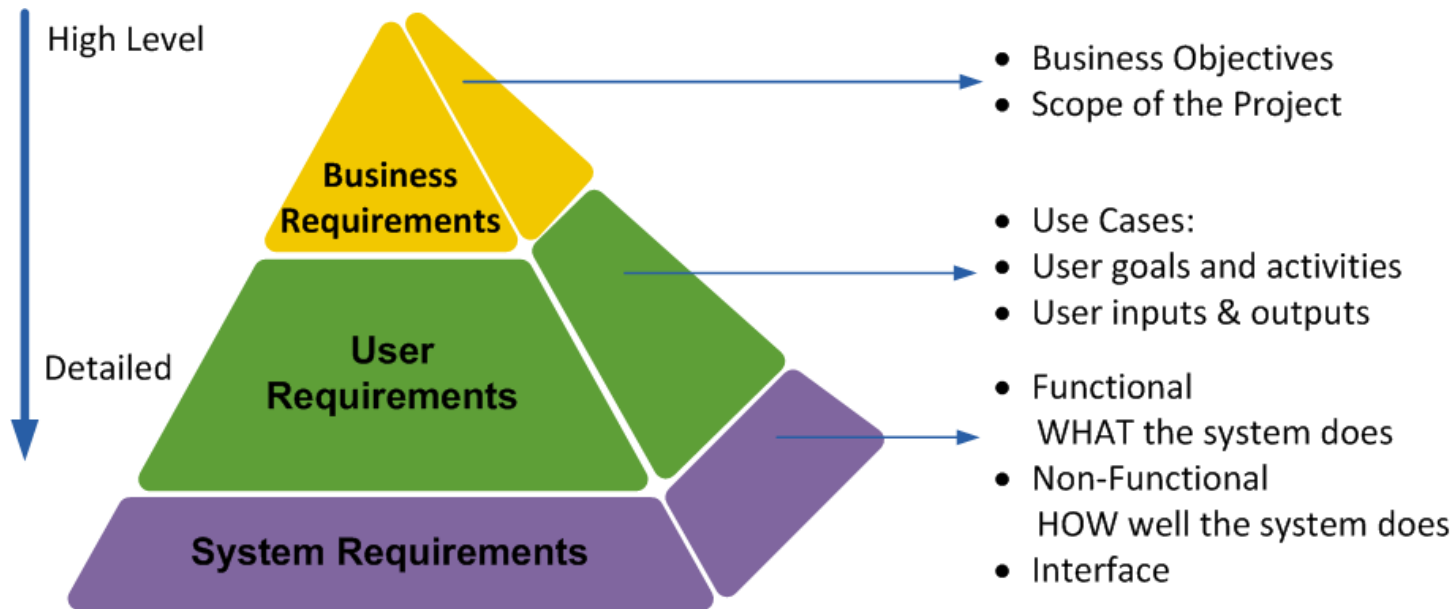
**Разработка требований (requirements engineering)** – процесс выявления, формулирования, анализа, документирования и проверки требований

# Что такое требование? (2)

Руководство к своду знаний по бизнес-анализу (BABOK) определяет требование следующим образом:

1. **Требование** — это задокументированное представление потребности.
2. Требования фокусируются на понимании, какой эффект будет достигнут, когда требование будет полностью удовлетворено.
3. Требования могут быть представлены в виде одного или нескольких документов, вид и содержание может широко варьироваться в зависимости от обстоятельств.

# Уровни требований



# Примеры требований

- Пользователь может вывести заявку на печать
- Система должна отобразить сообщение об ошибке, если пользователь ввел пароль менее 8 символов
- У команды поддержки должна быть возможность создавать заявки от пользователей, управлять ими и отображать статистику
- Система АТМ должна проверять достоверность PIN-кода в течение не более трех секунд
- Необходимо автоматизировать процесс подачи заявления на получение компенсации в условиях карантина/самоизоляции
- Время отклика для построения отчета должно занимать не более 5 сек для 90% операций

Напишите в чат, какое требование приведено.

Варианты ответов:

1. Бизнес-требование
2. Пользовательское требование
3. Функциональное требование
4. Нефункциональное требование

# С какими артефактами связаны требования?

- ▶ Видение проекта (Vision)
- ▶ Словарь терминов (Glossary)
- ▶ Бизнес-правила (Business rules)
- ▶ Модель прецедентов (Use-Case Model)
- ▶ Дополнительная спецификация (Supplementary Specification)
- ▶ ...



# Что такое Use Case?

Синонимы:

- ▶ Use Case, UC
- ▶ Вариант использования
- ▶ Прецедент

Простое определение:

**Use Case** – сценарий использования системы

# Пример УС (Сжатый формат) (1)

## Игра в кости

Игрок отправляет запрос на выбрасывание костей. Система выполняет вычисление и отображает результаты: если сумма очков составляет 7, игрок считается победителем, в противном случае — проигравшим.

# Пример УС (Сжатый формат) (2)

## Обработка продажи

Покупатель подходит к кассе с выбранными товарами. Кассир с помощью POS-системы регистрирует каждый товар и вычисляет общую сумму. Покупатель оплачивает. Система печатает чек. Покупатель получает товарный чек и покидает магазин с покупками.

# Что такое Use Case? (1)

- ▶ Набор взаимосвязанных успешных и неудачных сценариев, описывающий использование системы действующим лицом для решения одной из задач
- ▶ Набор повествовательных историй об использовании системы
- ▶ Текст, описывающий сценарий (возможно, не один) взаимодействия (кого или чего?) с системой, приводящий (возможно, не приводящий) к значимому (для кого?) результату

# Что такое Use Case? (2)

## Use case:

- ▶ Выражает поведенческую часть соглашения между участниками системы.
- ▶ Описывает поведение и взаимодействие участников при различных условиях.
- ▶ Показывает, как достигается (или не достигается) цель основного действующего лица.
- ▶ Собирает сценарии, имеющие отношение к цели основного действующего лица.

# Что такое Use Case? (3)

1. **Соглашение о поведении** рассматриваемой системы. Описывает поведение системы в ответах на запрос одного из участников в различных условиях. (Алистер Коберн)
2. **Набор последовательностей действий системы**, в том числе вариантов таких последовательностей, в результате которых получается наблюдаемый **результат**, который обладает некой ценностью для кого-то из участников процесса (Глоссарий UML)

# Требованиям какого уровня соответствуют прецеденты?

Напишите в чат свой  
вариант ответа.

Варианты ответов:

1. Бизнес-требование
2. Пользовательское требование
3. Функциональное требование
4. Нефункциональное требование

# Итоги:

# 1

## Что такое Use Case'ы?

- ▶ Разработка ПО начинается с анализа требований
- ▶ Use-Case Model – один из артефактов анализа требований
- ▶ Use-Case Model – набор сценариев использования
- ▶ Use Case = UC = Вариант использования = Прецедент
- ▶ UC – описание сценариев взаимодействия участников для достижения результата

Вопросы?



# 2

## Структура УС

# Базовые термины УС

**Actor (участник, исполнитель, действующее лицо)** – сущность, обладающая поведением.

**Пример:** роль (например, оператор 2й линии поддержки биллинговой системы), система или ее компонент, организация.

**Scenario (сценарий)** – последовательность действий между исполнителем и системой. Это один конкретный случай использования системы, экземпляр прецедента.

**Пример:** сценарий подключения опции, сценарий биллинга, неуспешный сценарий совершения платежа из-за прерванной транзакции

# Пример сценария

1. Пользователь дает команду на регистрацию.
2. BIS отображает форму регистрации.
3. Пользователь заполняет поля формы и подтверждает регистрацию.
4. BIS проверяет и подтверждает правильность заполнения формы.
5. BIS отправляет на указанный e-mail письмо с запросом для подтверждения регистрации.
6. Пользователь подтверждает регистрацию.
7. BIS регистрирует пользователя.

# Как правильно описывать шаги сценария

- ▶ Каждый шаг должен начинаться с подлежащего — кто или что выполняет действие.
  - ▶ Затем сказуемое — какое действие.
  - ▶ Сказуемое должно быть в настоящем времени и в действительном залоге
  - ▶ Дальше всё остальное
- 
- ✓ *«Пользователь выбирает населенный пункт»*
  - ✓ *«Система отображает список товаров»*
  - ✓ *«Клиент вводит карточку и PIN-код»*
  - ✗ *Сказуемое без подлежащего. «Данные сохраняются в БД» «Вводится значение атрибута».*
  - ✗ *Описание элементов интерфейса « «Пользователь нажимает на кнопку в левом верхнем углу»*

## Есть сценарий регистрации пользователя:

1. Пользователь дает команду на регистрацию.
2. BIS отображает форму регистрации.
3. Пользователь заполняет поля формы и подтверждает регистрацию.
4. BIS подтверждает правильность заполнения формы.
5. BIS отправляет на указанный e-mail письмо с запросом для подтверждения регистрации.
6. Пользователь подтверждает регистрацию.
7. BIS регистрирует пользователя.

## Чего не хватает в этом сценарии?



# Пример сценария

Альтернативные потоки выносятся из основного сценария в расширения

1. Пользователь дает команду на регистрацию.
2. BIS отображает форму регистрации.
3. Пользователь заполняет поля формы и подтверждает регистрацию.
4. BIS проверяет и подтверждает правильность заполнения формы.
5. BIS отправляет на указанный e-mail письмо с запросом для подтверждения регистрации.
6. Пользователь подтверждает регистрацию.
7. BIS регистрирует пользователя.

Расширения:

## 2а (Форма регистрации недоступна):

1. BIS отображает сообщение с призывом повторить действие или обновить страницу.  
Сценарий возвращается на шаг1.

## 3а (Пользователь закрывает форму):

1. Система отображает предупреждающее сообщение, что данные будут потеряны.  
Пользователь подтверждает. Завершение сценария

# И еще одно определение UC

**Use case** – набор сценариев использования, в котором каждый экземпляр сценария представляет собой последовательность действий, выполняемых системой для достижения ощутимого для конкретного исполнителя результата (Rational Unified Process)

«Экземпляр сценария» – это отдельная ветка сценария

# На чем фокусируется УС?

Чаще всего прецедент (УС) ориентирован на цели и задачи **пользователя**.

В процессе написания задаются вопросы:

- ▶ Кто является пользователем системы?
- ▶ Каковы типичные сценарии использования системы?
- ▶ Каковы цели и задачи пользователей?



# Форматы УС

**Сжатый – аннотация в одном абзаце** – обычно описывает только главный успешный сценарий.

**Пример:** *сценарий оформления продажи*

**Свободный – неформальный стиль описания** – описание занимает несколько абзацев и охватывает различные сценарии.

**Пример:** *сценарий регистрации в BIS*

**Расширенный (fully dressed) формат** – наиболее подробный стиль описания УС.

# Структура fully-dressed UC

- ▶ Название прецедента
- ▶ Рамки (границы)
- ▶ Уровень
- ▶ Основной исполнитель
- ▶ Заинтересованные лица и их требования
- ▶ Предусловия
- ▶ Триггер
- ▶ Результат
- ▶ Основной успешный сценарий
- ▶ Расширения
- ▶ Специальные требования
- ▶ Частота использования
- ▶ Открытые вопросы

# Расширенный формат УС (1)

**Название.** Оформление продажи

**Рамки.** Приложение автоматизации торговли NextGen.

**Уровень.** Задача, определенная пользователем.

**Основной исполнитель.** Кассир.

**Заинтересованные лица и их требования.**

- Кассир. Хочет точно и быстро ввести данные, не допуская ошибок в платеже, поскольку недостача вычитается из его зарплаты.
- Продавец. Хочет получить свои комиссионные от продажи.
- Покупатель. Хочет купить товары и быстро оформить покупку. Хочет получить подтверждение факта покупки для возможного возврата товара.
- Компания. Хочет аккуратно записать транзакцию и удовлетворить интересы покупателя. Хочет удостовериться, что служба авторизации платежей зафиксировала все данные о платеже. Заинтересована в обеспечении устойчивости к сбоям; хочет продолжать регистрировать продажи, даже если серверные компоненты (например, служба удаленной проверки кредитоспособности) недоступны. Хочет автоматически обновлять бухгалтерскую документацию и вести складской учет.
- Менеджер. Хочет быстро выполнять необходимые операции и легко решать задачи кассира.
- Государственные налоговые службы. Хотят получать налог от каждой продажи. Таких служб может быть несколько, в том числе национальная и местная.
- Служба авторизации платежей. Хочет получать цифровые запросы на авторизацию в корректном формате и отвечать на эти запросы.

**Предусловия.** Кассир идентифицирован и аутентифицирован.

**Результаты (Постусловия).** Данные о продаже сохранены. Налоги корректно вычислены. Бухгалтерские и складские данные обновлены. Комиссионные начислены. Чек сгенерирован. Авторизация платежа выполнена.

# Расширенный формат УС (2)

## Основной успешный сценарий (или основной процесс)

1. Покупатель подходит к кассовому аппарату POS-системы с выбранными товарами.
2. Кассир открывает новую продажу.
3. Кассир вводит идентификатор товара.
4. Система записывает наименование товара и выдает его описание, цену и общую стоимость.  
Цена вычисляется на основе набора правил.  
*Кассир повторяет действия, описанные в пп. 3-4, для каждого наименования товара.*
5. Система вычисляет общую стоимость покупки с налогом.
6. Кассир сообщает покупателю общую стоимость и предлагает оплатить покупку.
7. Покупатель оплачивает покупку, система обрабатывает платеж.
8. Система регистрирует продажу и отправляет информацию о ней внешней бухгалтерской системе (для обновления бухгалтерских документов и начисления комиссионных) и системе складского учета (для обновления данных).
9. Система выдает товарный чек.
10. Покупатель покидает магазин с чеком и товарами (если он что-то купил).

# Расширенный формат УС (3)

## Расширения (или альтернативные потоки)

\*а. Каждый раз при попытке менеджера отменить операцию выполняются следующие действия.

1. Система переходит в режим обслуживания менеджера.
2. Менеджер или кассир выполняют одну операцию в режиме обслуживания менеджера, т.е. изменяют баланс, завершают продажу и т.п.
3. Система возвращается в режим обслуживания кассира.

\*б. При каждом выходе системы из строя.

Для ввода системы в строй и корректной обработки платежа нужно обеспечить восстановление всех транзакций и событий с любого шага сценария.

1. Кассир перезапускает систему, регистрируется и предлагает восстановить предыдущее состояние.
2. Система восстанавливает предыдущее состояние.

2а. Система определяет аномалию, повлекшую сбой.

1. Система уведомляет об ошибке кассира, регистрирует ошибку и переходит в начальное состояние.

2. Кассир начинает новую продажу.

1а. Покупатель или менеджер хотят досрочно прервать продажу.

1. Кассир завершает операцию.

# Расширенный формат УС (4)

- 2. Система отображает состояние завершенной операции с общей стоимостью покупки.
  - 2а. Информация о продаже не найдена.
    - 1. Система сигнализирует об ошибке кассира.
    - 2. Кассир начинает новую операцию продажи и вводит все необходимые элементы.
  - 3. Кассир продолжает обслуживание клиента (вводит новые товары и обрабатывает платеж).
- 2а. Покупатель сообщает кассиру о льготах.
  - 1. Кассир проверяет наличие льгот и вводит шифр льготного покупателя.
  - 2. Система записывает статус покупателя, который будет использоваться при начислении налоговых платежей.
- 3а. Неправильный идентификатор (идентификатор не найден в системе).
  - 1. Система уведомляет об ошибке и отменяет ввод данного наименования товара.
  - 2. Кассир реагирует на ошибку.
    - 2а. Существует числовой идентификатор товара.
      - 1. Кассир вручную вводит идентификатор товара.
      - 2. Система отображает описание товара и стоимость.
    - 2а. Неправильный идентификатор. Система сигнализирует об ошибке. Кассир делает вторую попытку.
  - 2б. Не существует числового идентификатора, но на упаковке указана цена товара.
    - 1. Кассир просит менеджера ввести информацию о покупке данного товара.
    - 2. Менеджер выполняет операцию.
    - 3. Кассир вводит стоимость товара.
  - 2в. Кассир обращается за справкой об идентификаторе товара и его стоимости к справочной системе.
  - 2г. Кассир запрашивает информацию об идентификаторе и цене товара по телефону и вручную вводит эти данные.

# Расширенный формат УС (5)

- 3б. В рамках одной категории существует несколько различных наименований товара и идентифицировать конкретное наименование не нужно (например, 5 пакетов леденцов).
  - 1. Kassир может ввести идентификатор категории товара и количество единиц.
- 3в. Требуется вручную ввести категорию и цену товара (например, при покупке цветов).
  - 1. Kassир вручную вводит категорию товара и цену.
- 3-6а. Покупатель просит кассира отменить покупку одного из товаров.
  - 1. Kassир вводит идентификатор товара для удаления из продажи.
  - 2. Система отображает обновленную промежуточную стоимость.
- 3-6б. Покупатель просит кассира отменить продажу.
  - 1. Kassир отменяет продажу.
- 3-6в. Kassир приостанавливает продажу.
  - 1. Система записывает сведения о продаже таким образом, чтобы они были доступны с любого терминала POS-системы.
  - 2. Система печатает чек о приостановленной продаже.
- 4а. Сгенерированная системой цена товара не устраивает покупателя (например, у него есть дисконтная карта и он рассчитывает на более низкую цену товара).
  - 1. Kassир обращается к менеджеру.

# Расширенный формат УС (6)

- 2. Менеджер прерывает операцию.
- 3. Кассир вводит команду об изменении цены.
- 4. Система вычисляет новую цену.
- 5а. Система выявляет сбой при взаимодействии с внешней службой вычисления налога.
  - 1. Система перезапускает службу с данного узла POS-системы и продолжает работу.
    - 1а. Система определяет, что служба не перезапускается.
      - 1. Система сигнализирует об ошибке.
      - 2. Кассир может вручную вычислить и ввести сумму налога либо отменить продажу.
- 5б. Покупатель сообщает о положенной ему скидке (например, для сотрудников данного предприятия или постоянных покупателей).
  - 1. Кассир отправляет запрос на скидку.
  - 2. Кассир вводит идентификационные данные покупателя.
  - 3. Система представляет сумму скидки, вычисленную на основе дисконтных правил.
- 5в. Покупатель сообщает об открытом в данном магазине кредите на фиксированную сумму и просит оформить продажу с его помощью.
  - 1. Кассир отправляет запрос на оформление платежа с использованием открытого кредита.
  - 2. Кассир вводит идентификационную информацию о покупателе.
  - 3. Система снижает стоимость покупки (вплоть до 0) и уменьшает оставшуюся сумму кредита.
- 6а. Покупатель сообщает, что хочет оплатить покупку наличными, но у него недостаточно денег.
  - 1а. Покупатель использует альтернативный способ платежа.
  - 1б. Покупатель просит кассира отменить продажу. Кассир отменяет продажу в системе.



# Расширенный формат УС (7)

## 7а. Оплата наличными.

1. Кассир вводит предложенную покупателем сумму.
2. Система вычисляет положенную сдачу и открывает кассу с наличностью.
3. Кассир складывает поученные деньги и выдает сдачу покупателю.
4. Система регистрирует платеж наличными.

## 7б. Оплата по кредитной карточке.

1. Покупатель вводит информацию о своей кредитной карточке.
2. Система отображает необходимые данные.
3. Кассир вводит подтверждение.

### 3а. Кассир отменяет платеж.

1. Система возвращается в режим ввода товара.
4. Система отправляет запрос на авторизацию платежа внешней системе службы авторизации платежей и запрашивает подтверждение платежа.

### 4а. Система определяет сбой при взаимодействии с внешней системой.

1. Система сигнализирует об ошибке кассиру.
2. Кассир просит покупателя изменить тип платежа.
5. Система получает информацию о подтверждении платежа и сообщает об этом кассиру.

### 5а. Система получает информацию об отказе в выполнении платежа.

1. Система сообщает об отказе кассиру.
2. Кассир просит покупателя изменить тип платежа.

# Расширенный формат УС (8)

5б. Превышен интервал ожидания.

1. Система сообщает об этом кассиру.

2. Кассир выполняет вторую попытку или просит покупателя изменить тип платежа.

6. Система регистрирует платеж по кредитной карточке, включая информацию о подтверждении платежа.

7. Система предоставляет механизм ввода подписи для платежа по кредитной карточке.

8. Кассир просит покупателя подписать чек на оплату по кредитной карточке. Покупатель подписывает чек.

9. Кассир помещает подписанный чек в кассу.

7в. Оплата чеком.

7г. Оплата по дебитной карточке.

7д. Кассир отменяет платеж.

1. Система возвращается в режим ввода информации о товарах.

7е. Покупатель предоставляет купоны.

1. До обработки платежа кассир вводит информацию о каждом купоне, в соответствии с которой система снижает стоимость покупки. Система регистрирует предъявленный купон для нужд бухгалтерии.

1а. Купон действует не для всех выбранных товаров.

1. Система сигнализирует об ошибке кассиру.

# Расширенный формат УС (9)

9а. Генерация чека.

1. Система выводит формы и чеки для каждого товара.

9б. Покупатель просит выдать ему подарочный чек (без указания цены).

1. Кассир вводит запрос на подарочный чек, и система выдает его.

9в. В принтере закончилась бумага.

1. Система сообщает о проблеме.
2. Кассир добавляет бумагу.
3. Кассир запрашивает новый чек.

## Специальные требования

- Сенсорный экран с интерфейсом пользователя для большого плоского монитора. Текст должен быть виден с расстояния один метр.
- Отклик службы авторизации в 90% случаев приходит в течение 30 секунд.
- Каким-то образом нужно обеспечить робастное восстановление информации в случае сбоя при доступе к удаленным службам, таким как система складского учета.
- Возможность локализации (представления на различных языках) отображаемого текста.
- Возможность добавления новых бизнес-правил на шагах 3 и 7 в процессе функционирования системы.
- ...

# Расширенный формат UC (10)

## Список технологий и типов данных

- \*а. Менеджер прерывает продажу путем ввода специальной карточки или указания кода авторизации с клавиатуры.
- 3а. Идентификатор товара считывается со штрихкода (при наличии последнего) лазерным сканером или вводится с клавиатуры.
- 3б. Идентификатор товара может определяться по схемам кодирования UPC, EAN, JAN или SKU.
- 7а. Информация об открытом кредите вводится с помощью считывающего устройства или с клавиатуры.
- 7б. Подпись при оплате чеком ставится на бумажном документе. Однако ожидается, что в течение двух лет большинство покупателей будут требовать цифровые устройства считывания подписи.

**Частота использования:** почти постоянно.

## Открытые вопросы

- Изучить законодательство по налогообложению.
- Исследовать вопрос восстановления удаленных служб.
- Какая настройка потребуется для различных типов магазинов.
- Должен ли кассир снимать кассу при выходе из системы.
- Может ли пользователь сам использовать устройство считывания данных с карточки или это должен делать кассир.

# Термины УС(1)

## Уровни УС:

- ▶ **Обобщенный** (business-goal level use case) описывает систему на уровне бизнес-процессов.
- ▶ **Пользовательский** (user-goal level use case) описывает сценарии достижения цели главного исполнителя.
- ▶ **Вспомогательный** (subfunction-level use case) описывает вспомогательные действия, необходимые для достижения цели исполнителя.

**Вопрос:** Является ли УС «Аутентификация в системе» вариантом использования?

# Термины УС(2)

**Заинтересованные лица и их потребности** – Описывает все, что служит удовлетворению запросов заинтересованных лиц. Кроме того, начиная описание прецедента с перечня заинтересованных лиц и их интересов, можно более точно определить функции системы

Например:

- *Кассир. Хочет точно и быстро ввести данные, не допуская ошибок в платеже, поскольку недостача вычитается из его зарплаты.*
- *Продавец. Хочет получить свои комиссионные от продажи.*

# Термины УС(3)

**Предусловия (preconditions)** — это перечень предпосылок, которые всегда должны выполняться до начала сценария прецедента.

**Триггер** — определяет событие, которое запускает вариант использования. Иногда триггер предшествует первому шагу варианта использования, иногда сам является первым шагом.

**Пример:** *система банкомата начинает работать, только когда пользователь вставляет в банкомат свою карточку.*  
*Нет смысла говорить «Пользователь решил использовать банкомат».*  
*Триггер «Клиент вставляет карточку» является также первым шагом варианта использования.*

# Термины УС(4)

**Результаты или постусловия (postconditions)** – Описывают, какие условия обязательно должны выполняться в случае успешного завершения сценария. Эти результаты должны удовлетворять интересам заинтересованных лиц.

**Пример:** *данные о продаже сохранены. Чек сгенерирован.*

**Минимальные гарантии** — наименьшее обещание системы участникам, если цель основного действующего лица не может быть достигнута.

**Пример:** *запись в лог файл кода ошибки, отображение сообщения об ошибке.*



# Основной успешный сценарий

В разделе основного сценария описываются три вида действий:

1. Взаимодействие между исполнителями
2. Верификация (обычно со стороны системы)
3. Изменение состояния системы

Первый шаг не всегда попадает под эту классификацию и часто является *триггером*

*Совет:* Выносите все условия или возможные варианты развития событий в раздел расширений.

# Расширения сценария (1)

**Расширение сценария** - это фрагмент сценария, который запускается при возникновении конкретного условия в другом сценарии.

**Точка расширения** — это метка или краткое имя точки линии поведения в основном сценарии, в которой вариант использования расширения может его прервать.

**Условие расширения** — это обстоятельства, при которых меняется линия поведения.

*Совет.* Описывайте условия как факты, выявляемые системой или исполнителем. Сравните следующие два пункта.

A. 5a. Система выявляет сбой при коммуникации с внешней службой вычисления налога.

B. 5a. Внешняя система вычисления суммы налога не работает.

# Расширения сценария (2)

**Расширения (альтернативные потоки)** – ответвления от основного сценария. Описание расширения состоит из условия и способа обработки.

## Основной сценарий:

...

3. Кассир сканирует идентификатор товара
4. Система отображает описание товара

## Расширения:

3а. Идентификатор не распознан:

1. Система уведомляет об ошибке и отменяет ввод товара

3б. Ручной ввод товара:

1. Кассир вручную вводит товар

# Расширения сценария (3)

## **Основной сценарий (sunny day)**

Основной способ достижения цели основного действующего лица

## **Альтернативный сценарий**

Другой способ достижения цели основного действующего лица

## **Исключительный сценарий (raining day)**

Неуспешный сценарий. Поток, который не приводит к достижению цели

# Примеры условий неудачи

## Примеры исключительных условий для ВИ «Снятие денег через банкомат»

1. Карточка повреждена
2. Карточка «чужого» банка
3. Неверный PIN
4. Клиент вовремя не ввел PIN
5. Банкомат не работает
6. Сеть не работает
7. Клиент вовремя не ввел сумму
8. Запрошенная сумма слишком велика
9. Купюры замялись во время выдачи
10. Клиент вовремя не забрал деньги из раздаточного устройства

# Типовые ошибки

1. Отсутствует система
2. Отсутствует основное действующее лицо
3. Детализация пользовательского интерфейса
4. Низкие уровни цели
5. Несоответствие цели и содержания, смешение нескольких сценариев, изменение действующего лица

# Типовые ошибки (1)

## Вариант использования:

### Снять наличные со счета

**Область действия:** банкомат

**Уровень:** цель пользователя

**Основное действующее лицо:** клиент

1. Клиент вводит карточку и PIN.
2. Клиент вводит "снять деньги" и количество.
3. Клиент забирает наличные, карточку и квитанцию
4. Клиент уходит.

## Вариант использования:

### Снять наличные со счета

**Область действия:** банкомат

**Уровень:** цель пользователя

**Основное действующее лицо:** клиент

1. Получает карточку для банкомата, PIN.
2. Получает в качестве типа транзакции "снять деньги".
3. Получает требуемую сумму.
4. Удостоверяется, что на счете достаточно денег.
5. Выдает деньги, квитанцию, карточку.
6. Возвращается в исходное состояние.

# Типовые ошибки (2)

**Вариант использования:**

---

## **Совершить покупку**

**Область действия:** приложение для электронной торговли

**Уровень:** цель пользователя

**Основное действующее лицо:** клиент

1. Система устанавливает экран для ввода ID и пароля.
2. Клиент вводит ID и пароль, щелкает по кнопке "ОК".
3. Система подтверждает ID и пароль, отображает персональные данные.
4. Клиент вводит имя и фамилию, номер дома, название улицы, города, штата, почтовый индекс, номер телефона и щелкает по кнопке "ОК".
5. Система подтверждает, что пользователь ей известен.
6. Система выводит список товаров, имеющихся в наличии.
7. Клиент щелкает по изображению товара, который хочет купить, вводит рядом с каждой картинкой количество, по окончании щелкает по кнопке "Готово".
8. Система удостоверяется с помощью системы управления складами, что на складе есть достаточное количество требуемых товаров. ... и т.д.



# Типовые ошибки (3)

**Вариант использования:**

---

**Совершить покупку**

**Область действия:** приложение для электронной торговли

**Уровень:** цель пользователя

**Основное действующее лицо:** клиент (пользователь)

1. Пользователь регистрируется в системе с помощью ID и пароля.
2. Система подтверждает ID и пароль.
3. Пользователь вводит имя.
4. Пользователь вводит адрес
5. Пользователь вводит номер телефона.
6. Пользователь указывает товар.
7. Пользователь указывает количество.
8. Система подтверждает, что пользователь ей известен.
9. Система подключается к системе управления складами.
10. Система запрашивает у складской системы текущие уровни запасов.

# Итоги:

# 2

## Структура УС

- ▶ УС ориентирован на цели и задачи участника
- ▶ Основные понятия УС: Actor и Scenario
- ▶ Форматы УС: Сжатый, Свободный, Разширенный
- ▶ Основной сценарий (sunny day). Основной способ достижения цели
- ▶ Альтернативный сценарий – другой способ достижения цели
- ▶ Исключительный сценарий (raining day) - неуспешный сценарий. Этот поток не приводит к достижению цели

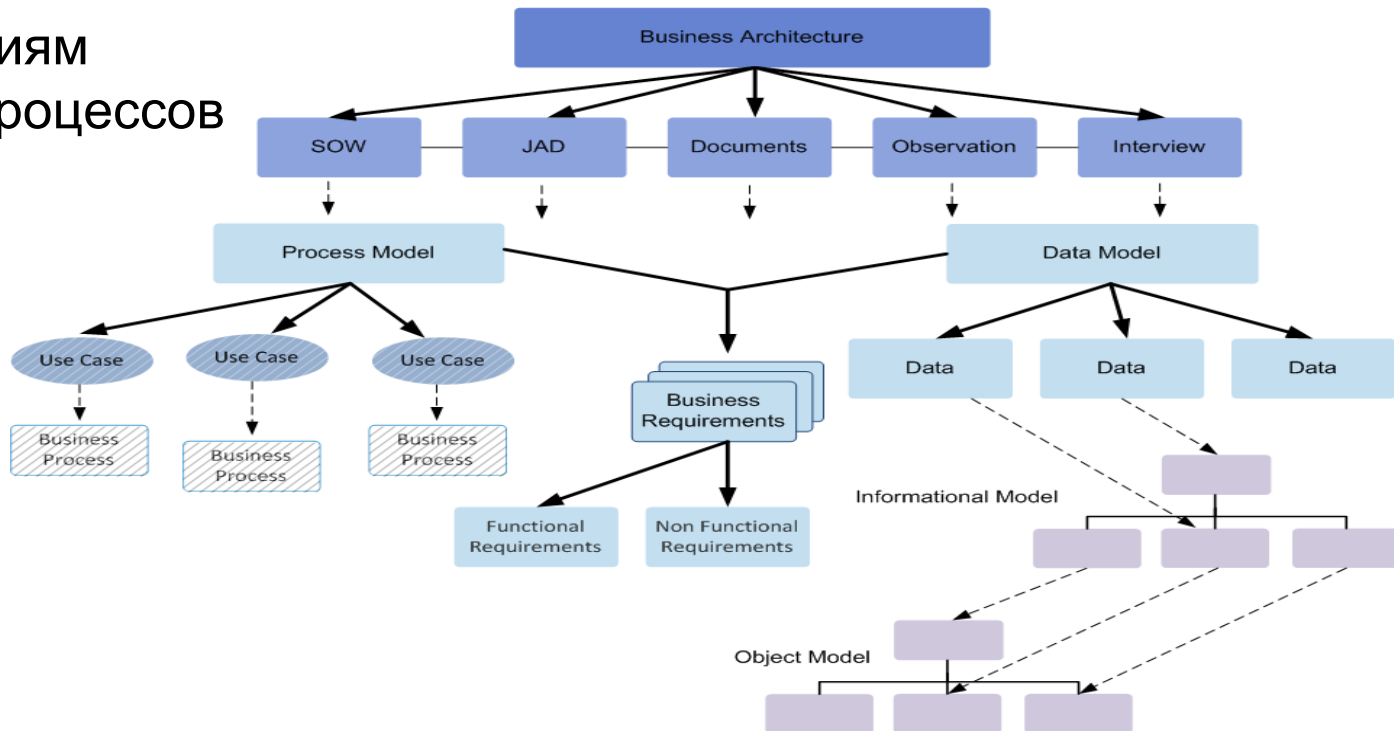
Вопросы?

# 3

## Методы выявления УС

# Как выделить УС?

1. По требованиям
2. По модели процессов



# Классический подход выявления УС

1. Определите границы системы
2. Идентифицируйте основных исполнителей
3. Для каждого исполнителя определите его задачи
4. Определите прецеденты, удовлетворяющие потребности каждого исполнителя
5. Grooming и детализация

# Определение основных исполнителей и задач

1. Кто запускает и выключает систему?
2. Кто является системным администратором?
3. Кто осуществляет управление пользователями и безопасностью?
4. Относится ли время к числу исполнителей, другими словами, должна ли система выполнять какие-либо действия в ответ на события времени?
5. Существует ли процесс мониторинга, как система перезапускается при сбое?
6. Кто контролирует деятельность и производительность системы?
7. Как выполняется обновление программного обеспечения?
8. Кто анализирует журналы регистрации?
9. Могут ли в качестве исполнителей выступать внешние программы или автоматические системы?
10. Кого следует уведомлять при ошибках или сбоях в системе?

# Как составить перечень исполнителей и их задач

Существуют следующие подходы к решению этой задачи.

1. При построении диаграммы прецедентов названия прецедентов рассматриваются как задачи системы.
2. Можно сначала составить перечень исполнителей, уточнить его, а затем строить диаграмму прецедентов.
3. Анализ внешних событий.

Роль	Кассир	Менеджер	Системный администратор
Задачи	<ul style="list-style-type: none"><li>• Оформляет продажи</li><li>• Оформляет кредиты</li><li>• Выполняет возврат товара</li><li>• Регистрирует выручку</li></ul>	<ul style="list-style-type: none"><li>• Включает систему</li><li>• Выключает систему</li></ul>	<ul style="list-style-type: none"><li>• Добавляет пользователей</li><li>• Изменяет параметры пользователей</li><li>• Удаляет пользователей</li><li>• Управляет безопасностью</li></ul>

# Анализ внешних событий

Зачастую к одному и тому же уровню EBP (EBP — Elementary Business Processes) или прецеденту, относится целая группа событий

Событие	Инициатор	УС/Задача
Регистрация обращения пользователя	Оператор поддержки	Оформление заявки
Обращение за услугой	Пользователь	Оформление заявки
Ввод причины обращения	Оператор поддержки или пользователь	Оформление заявки
Ввод информации о решении заявки	Оператор поддержки	Оформление заявки



# Варианты использования CRUD

## CRUD – Create, Read, Update, Delete

Обычно, каждой задаче пользователя соответствует один прецедент

Типичным исключением из правила соответствия задач и прецедентов является прецедент, решающий четыре задачи — создание, поиск (извлечение, чтение), обновление и удаление.

Обычно такой прецедент называется Управление <чем-либо>.

## Кандидаты в УС:

1. Переговоры с поставщиком
2. Обработка возврата товара
3. Регистрация в системе
4. Перемещение фишки по игровой доске

Какой из пунктов  
можно считать  
прецедентом?



# 3 критерия для выделения УС

- ✓ Критерий бизнес-ценности (аппрув шефа)
- ✓ Критерий EBP
- ✓ Критерий размера

## Применение критериев для кандидатов в УС:

- ✓ Обработка возврата товара
- ✗ Регистрация
- ✗ Переговоры с поставщиком
- ✗ Перемещение фишки по игровой доске

Возможны обоснованные отклонения!

# Итоги:

# 3

## Методы выявления УС

- ▶ УС выявляются по процессам и требованиям
- ▶ 4 шага: определить границы системы, исполнителей, их задачи и создать УС. Далее – груминг и детализация
- ▶ Одной задаче соответствует один прецедент, исключение – **CRUD**
- ▶ Критерии выявления УС: бизнес-ценность, EBR, размер
- ▶ Возможны обоснованные отклонения

Вопросы?

# 4

## Синтез карты УС

# Ссылки на другие УС (1)

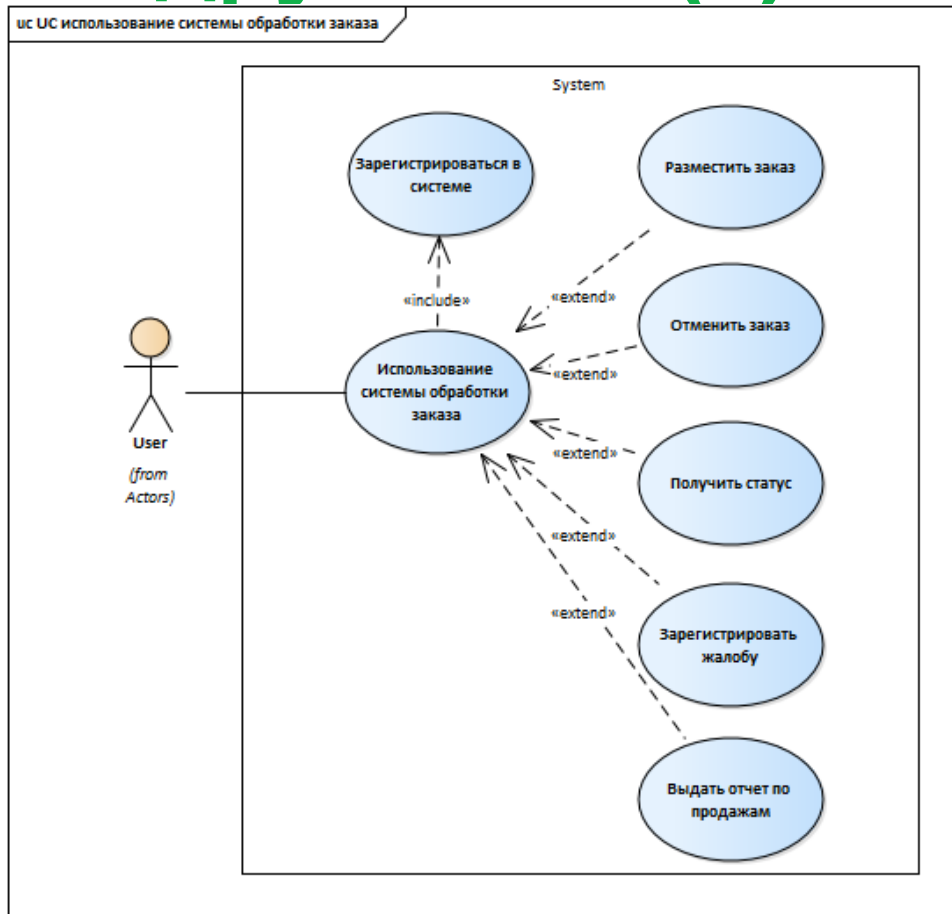
УС: Использовать систему обработки заказов

Основной сценарий:

1. Пользователь выполняет сценарий Зарегистрироваться в системе
2. Система предоставляет доступные функции. Пользователь выбирает и выполняет одну из них:
  - Разместить заказ
  - Отменить заказ
  - Получить статус
  - Зарегистрировать жалобу
  - Выдать отчет по продажам
3. Это повторяется, пока пользователь не решит выйти
4. Система регистрирует выход пользователя, когда пользователь укажет на выход

# Ссылки на другие УС (2)

Диаграмма  
вариантов  
использования



# Синтез карты Use cases

1. Добавить границы систем
2. Добавить сценарии
3. Добавить участников
4. Связать сценарии и участников, добавить отношения



# Отношения между УС (1)

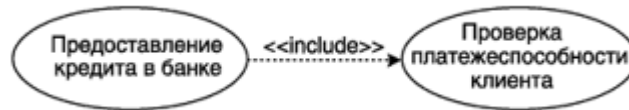
## Отношение взаимодействия:

- Связывает сценарии и участников



## Отношение содержания:

- Показывает включение функциональности одного сценария в другой



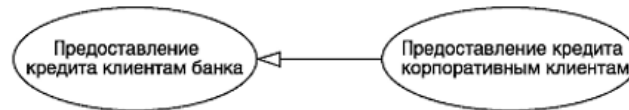
## Отношение расширения:

- Отражает связь включения опционального поведения



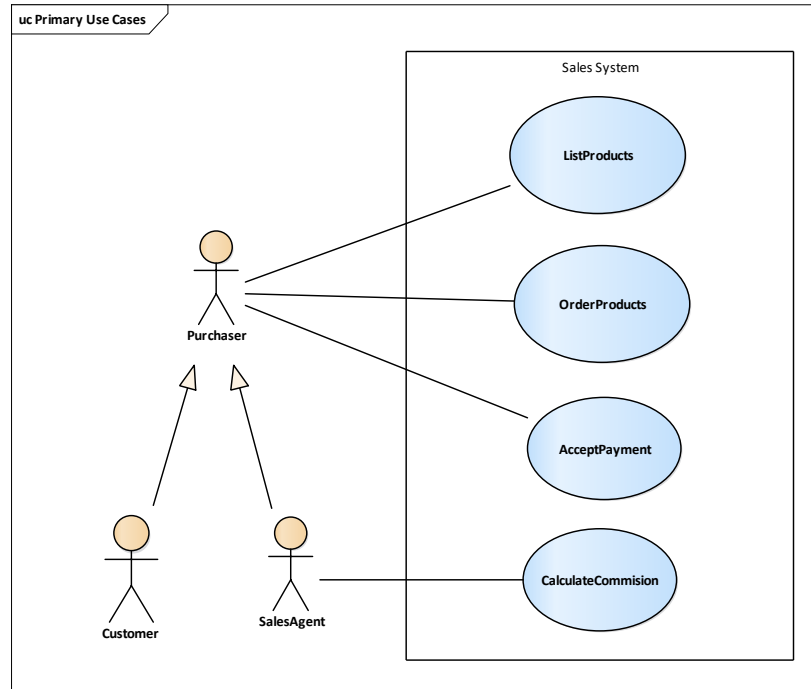
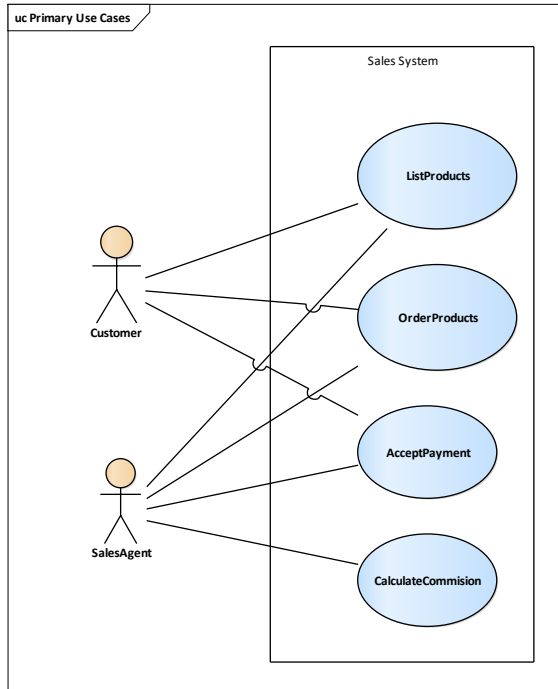
## Отношение обобщения:

- Отражает связь специализированного сценария к общему



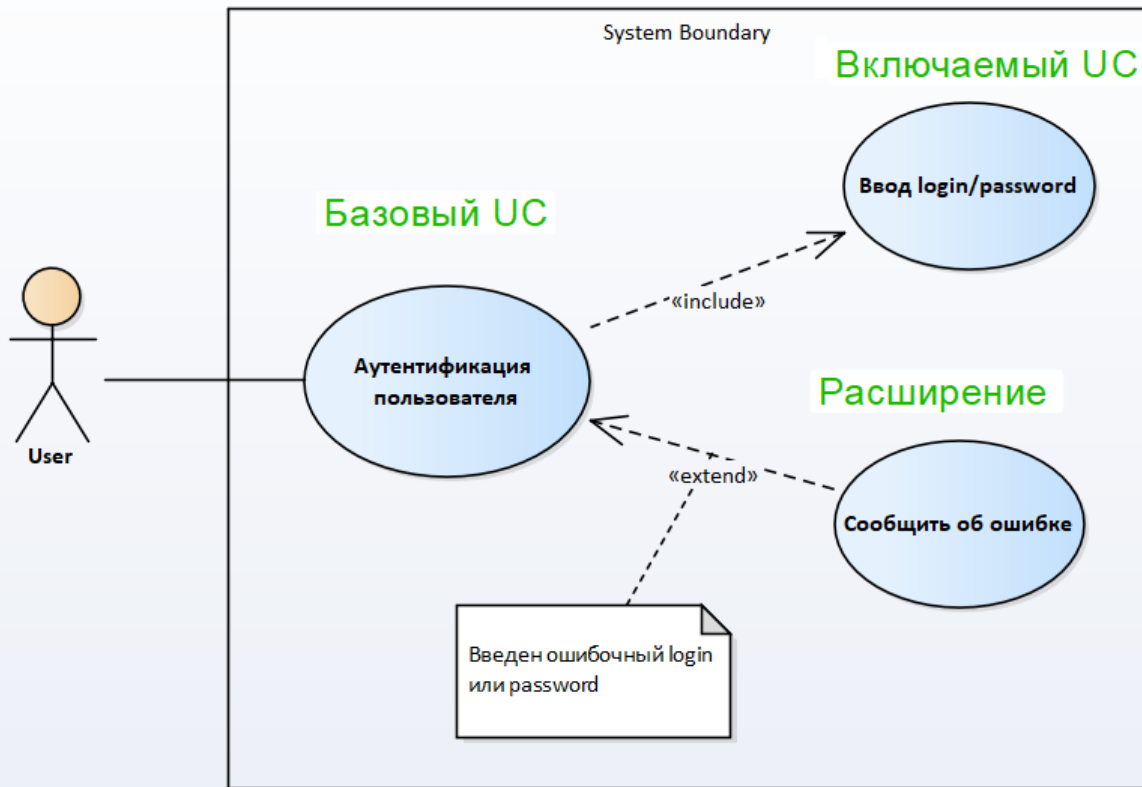
# Отношения между УС (2)

## Отношение обобщения

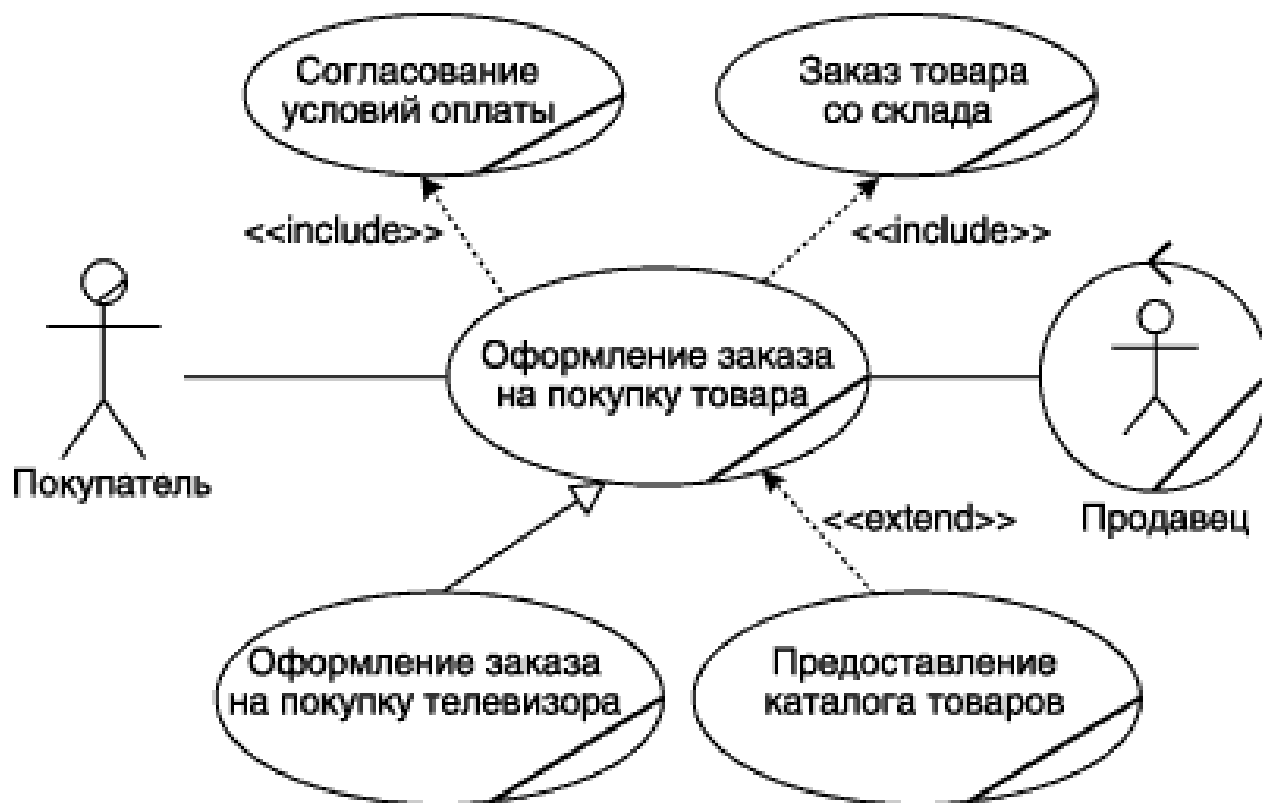


# Отношения между УС (3)

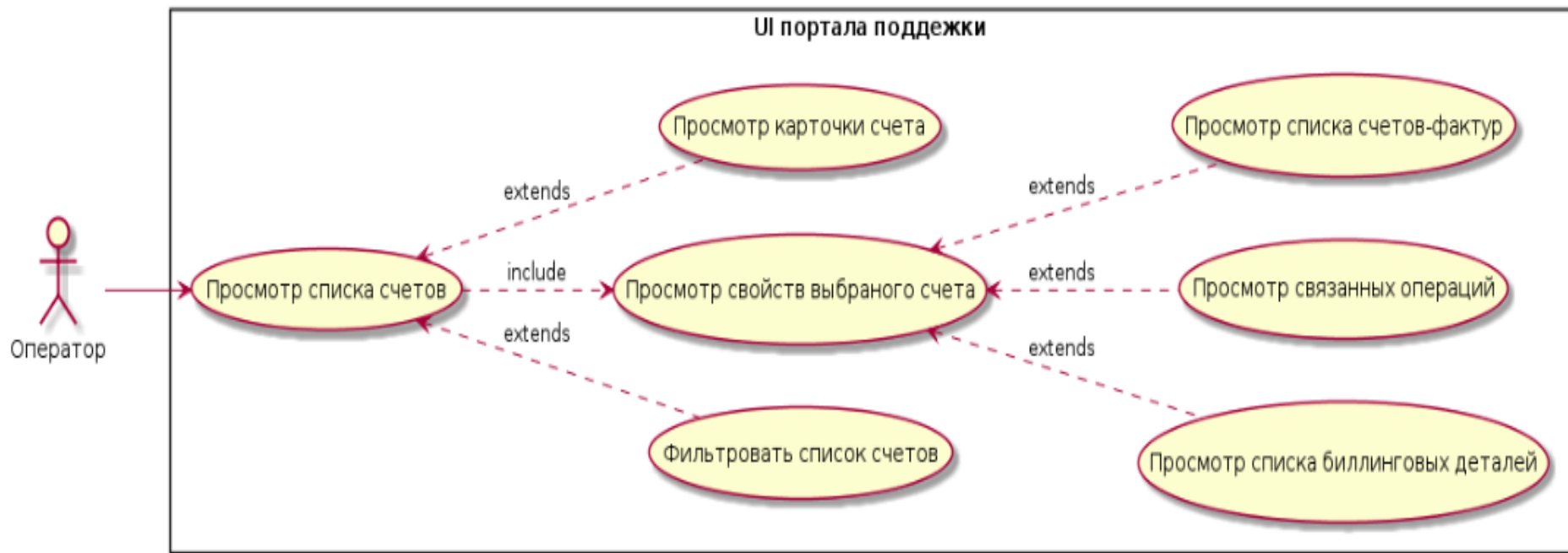
Отношения включения и расширения



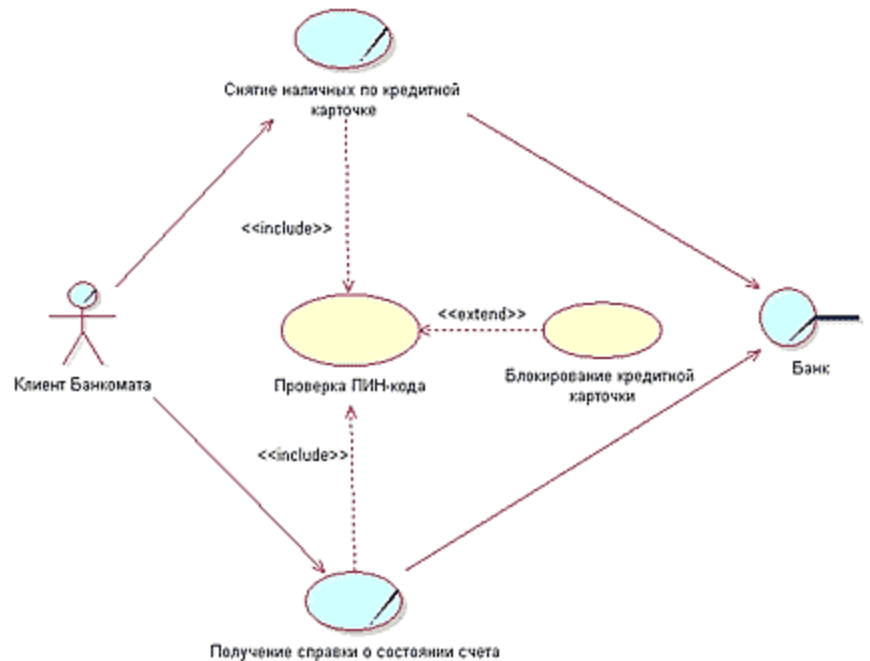
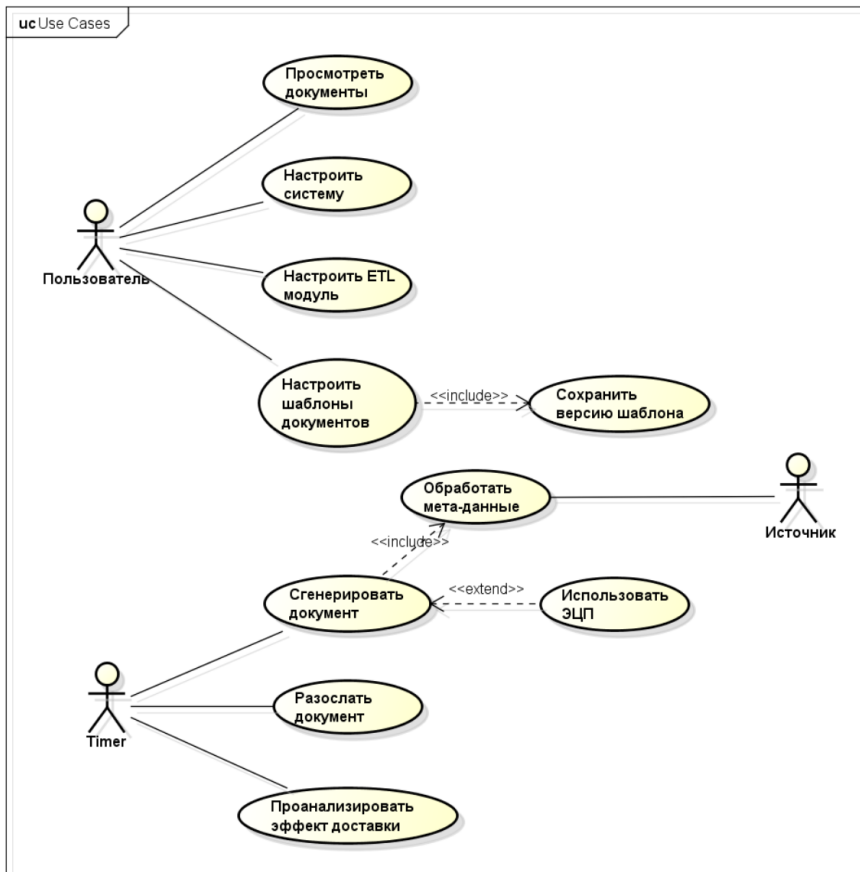
# Отношения между УС (4)



# Use case map (1)

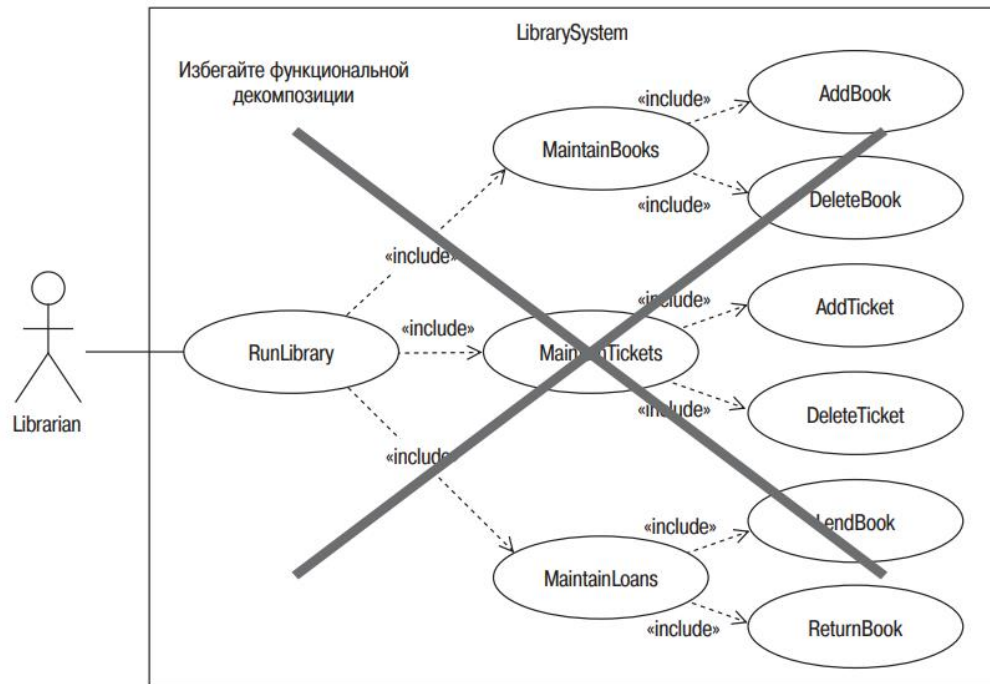


# Use case map (2)



# Типовая ошибка моделирования

Функциональная декомпозиция не подходит для моделей прецедентов.



# Итоги:

# 4

## Синтез карты УС

- ▶ Обобщение обеспечивает возможность вынести общее поведение на родительский уровень
- ▶ «Include» позволяет вынести повторяющиеся шаги в отдельный прецедент
- ▶ «Extend» вводит новое поведение в существующий (базовый) прецедент
- ▶ Следует избегать функциональной декомпозиции

Вопросы?



# 5

Инструменты

Литература

Практика

# Инструменты (1)

## Простая визуализация и текст

- ✓ Microsoft Visio + Word
- ✓ Google Documents
- ✓ Gliffy
- ✓ Draw.io
- ✓ Confluence
- ✓ ... тысячи их!

# Инструменты (2)

## Plantuml

<https://plantuml.com/>

```
@startuml
```

```
actor customer
```

```
actor clerk
```

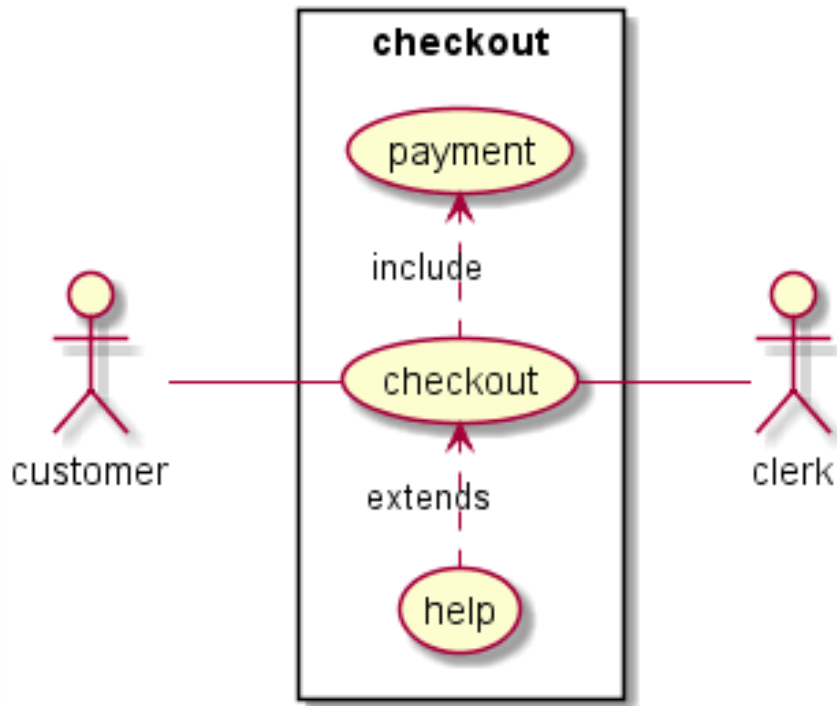
```
rectangle checkout
```

```
{
```

```
    customer -- (checkout)  
    (checkout) .> (payment) : include  
    (help) .> (checkout) : extends  
    (checkout) -- clerk
```

```
}
```

```
@enduml
```



# Инструменты (3)



# Инструменты (3)

UseCase: CreateAccount

Properties

- General
- Rules
  - Requirements
  - Constraints
  - Scenarios
- Related
  - Files
  - Links

Scenario: Basic Path

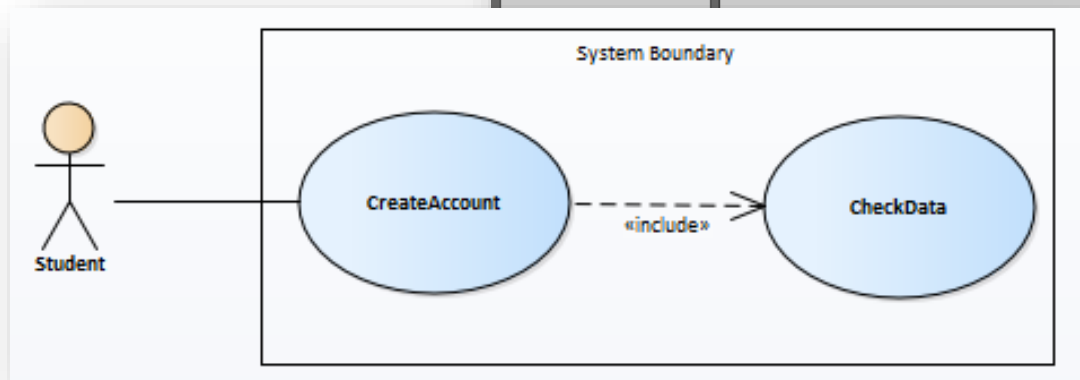
Type: Basic Path

Description: Structured Specification

Step	Action	Uses	Results	State
1	Student запрашивает создание аккаунта в LIS	Student	LIS создает черновик	
2	LIS создает черновик заявления			
3	LIS отображает форму ввода данных Student	LIS		
4	Student заполняет данные в LIS	Student		

OK Cancel Apply Help

Type	Join
Basic Path	-
Exception	11
Alternate	3



# Рекомендация литературы

1. Современные методы описания функциональных требований. Алистер Коберн
2. Применение UML 2.0 и шаблонов проектирования. Крэг Ларман
3. UML 2 и унифицированный процесс. Джим Арлоу, Айла Нейштадт



# Что дальше? (1)

## УС дополняют:

- ▶ Словарь терминов, вижен, бизнес-ограничения
- ▶ Специальные (нефункциональные) требования
- ▶ Диаграммы бизнес-процессов, деятельности, последовательностей, классов, статусов, компонентов и развертывания
- ▶ Мокапы, варфреймы и прототипы пользовательских экранов
- ▶ Спецификации системных операций, API и интеграционные
- ▶ Матрица трассировки требований
- ▶ Тест-кейсы (сценарии проверки)

# Что дальше? (3)

## REQUIREMENTS TRACEABILITY MATRIX

Project Name: Online Flight Booking Application

Business Requirements Document BRD		Functional Specification Document FSD			Use Cases		Test Case Document
Business Requirement ID#	Business Requirement / Business Use case	Functional Requirement ID#	Functional Requirement	Priority	Use Case ID#	Use Case	Test Case ID#
BR_1	Reservation Module (It should allow user to book one or more tickets, one way or round way for future dates)	FR_1	One Way Ticket booking (It should allow user to book one way ticket)	High	UC#001	User is able to book one way ticket	TC#001 (Verify if user is able to book one way ticket)
BR_1		FR_1		High	UC#002	User is able to book multiple way tickets	TC#002 (Verify if user is able to book multiple way tickets)
BR_1		FR_2	Round Way Ticket (It should allows user to book round way ticket)	High	UC#003	User is able to book round way ticket	TC#003 (Verify if user is able to book round way ticket)
				High	UC#004	User is able to book multiple round way tickets	TC#004 (Verify if user is able to book multiple round way tickets)
BR_1		FR_3	Multicity Ticket booking (It should allows user to book one way or round way ticket for multiple cities)	High	UC#005	User is able to book one way ticket for multiple cities	TC#005 (Verify if user is able to book one way ticket for multiple cities)
				High	UC#006	User is able to book book round way ticket for multiple cities	TC#006 (Verify if user is able to book round way ticket for multiple cities)
BR_2	Payment Module (User should able to make payment for booked tickets via Credit / Debit Card)	FR_4	By Credit Card (It should allows user to make payment by Credit Cards)	High	UC#007	User is able to pay by Master Card	TC#007 (Verify if user is able to pay by Master Card)
				High	UC#008	User is able to pay by Visa Card	TC#008 (Verify if user is able to pay by Visa Card)
BR_2		FR_5	By Debit Card (It should allows user to make payment by Debit Cards)	High	UC#009	User is able to pay by Debit Card	TC#009 (Verify if user is able to pay by Debit Card)



# Практика.

## Создаем варианты использования...

An aerial photograph of a city with numerous skyscrapers and buildings. Overlaid on the image are several large, semi-transparent green geometric shapes, including triangles and parallelograms, creating a modern, tech-oriented aesthetic.

**NX Bootcamp**

# Выход из дома по QR-коду

# Задание

Определить основное действующее лицо и написать основной поток сценария для кейсов:

- ✓ «Оформить разовый пропуск на выход из дома»
- ✓ «Проверка наличия основания для выхода на улицу»

Результаты присылать на [Aleksandr.Zakharov@nexign.com](mailto:Aleksandr.Zakharov@nexign.com)

В ответ пришлю шаблон УС и вторую часть задания

# Преимущества Use Cases

- ▶ Базируются на целях участников
- ▶ Описывают взаимодействие со стороны (без деталей реализации)
- ▶ Читабельны
- ▶ Описывают функциональные требования
- ▶ Документируют бизнес-процессы
- ▶ Содержат расширения основного сценария
- ▶ Могут записываться в свободной и полной форме
- ▶ Простота и наглядность использования диаграмм UC
- ▶ Являются основой для других артефактов

# Итоги

- ▶ UC – инструмент описания функциональных требований, UC ориентированы на взаимодействие участников
- ▶ Познакомились с терминологией и структурой UC
- ▶ Изучили основные способы выявления UC
- ▶ Обсудили лучшие практики и типовые ошибки при создании UC
- ▶ Рассмотрели примерами UC из практики
- ▶ Рассмотрели создание карты UC
- ▶ Ознакомились с отношениями между вариантами использования
- ▶ Ознакомились с инструментами моделирования и литературой для дальнейшего изучения

# 5

## Q&A

# Thank you