

# ТИПИС 4 курс

1. Разработка требований к информационным процессам и системам. Уровни и типы требований. Требования в аспекте различных категорий проектов.
  1. Что охватывает
  2. Что включает
  3. Способы представления
  4. Цикл разработки
  5. \*\*Уровни требований
  6. **Типы требований**
2. Практические приемы разработки требований. Каркас процесса создания требований: выявление требований, анализ требований, спецификация требований, проверка требований, управление требованиями.
  1. Стадии
  2. Диаграмма (все со всем)
  3. Для чего каждый этап
3. Спецификация бизнес-требований. Концепция продукта и границы проекта. Способы представления границ проекта. Способы представления концепции проекта. Применение бизнес-целей для определения момента завершения проекта.
  1. Что такое концепция и границы продукта.
  2. Способы представления.
4. Выявление и документирование требований пользователей. Методы выявления требований пользователей и их особенности. Классификация предоставляемой пользователями информации и ее роль в проекте требований.
  1. Варианты использования
  2. Пользовательские истории
  3. Как их выявить?
5. Спецификация функциональных требований. Приемы разработки и управления функциональными требованиями.
  1. Что такое функциональные требования.
  2. Формат.
  3. Функции системы.
  4. Определение ГОСТ 34.
6. Моделирование требований. Варианты комбинации текстовых и визуальных способов представление требований на различных уровнях абстракции.
  1. Что такое моделирование требований.
  2. Три основных действия.
  3. Разные виды диаграмм.

7. Определение требований к данным. Приемы и механизмы разработки требований к данным: словарь данных, анализ данных, спецификация отчетов.
  1. Нормализация данных
  2. Нормальные формы
  3. Что такое модель данных
  4. Сущности, связи, атрибуты, мощности.
  5. Что такое словарь данных.
  6. Как делать анализ. CRUD таблица.
  7. Что такое спецификация отчетов.
8. Документирование требований. Спецификация требований к ПО: требования к именованию, шаблоны спецификации требований, уровни детализации требований, способы визуализации, приемы по проверке неполноты.
  1. Что должна и не должна содержать?
  2. Способы представления.
  3. Виды именований (по порядку, иерархии, текстовые иерархические теги).
  4. Уровни детализации
  5. Проверка по симметрии, сложной логике, отсутствию исключений
9. Прототипы как средство снижения риска. Модели и экспериментальные образцы. Одноразовые и эволюционные прототипы. Бумажные и электронные прототипы.
  1. Что такое окончательный продукт.
  2. Бумажные, электронные прототипы
  3. Модели, экспериментальные, одноразовые и эволюционные прототипы.
10. Требования в проектах определенных классов. Особенности различных типов проектов. Проекты автоматизации бизнес-процессов: особенности и отличия от проектов других классов.
  1. agile
  2. Проекты по доработке или замене систем
  3. Проекты с серийным продуктом
  4. Проекты выполняемые сторонними организациями
  5. Проекты бизнес-аналитики
  6. Проекты встроенных и других систем реального времени
  7. Проекты автоматизации бизнес-процессов
11. Требования к интерфейсам. Основные эргономические принципы создания интерактивных систем в общем виде.
  1. Приемлемость организации диалога для выполнения производственного задания
  2. Информативность
  3. Соответствие ожиданиям пользователей
  4. Пригодность для обучения
  5. Контролируемость

6. Устойчивость к ошибкам
7. Адаптируемость к индивидуальным особенностям пользователя
12. Визуализация различных уровней требований: диаграммы, модели, схемы.
  1. Что такое моделирование требований
  2. Стадии моделирования
  3. Методы моделирования
  4. Элементы моделирования
  5. Структура моделирования требований
13. Различные подходы в спецификации требований: примеры авторских методик, техническое задание по ГОСТ 34.
  1. Подход по Карлу Вигерсу
  2. Подход по Коберну Алистеру
  3. Подход по ГОСТ-34
  4. *Методика Volere*
14. Нотация моделирования бизнес-процессов (BPMN): область применения. Основные категории элементов нотации. Категории диаграмм бизнес-процессов в BPMN. Уровни моделирования бизнес-процессов.
  1. Что такое?
  2. 5 основных категорий: (Элементы потока, Данные, Соединяющие элементы, Зоны ответственности,Arteфакты)
  3. Способы группировки.
  4. Виды диаграмм
  5. Уровни моделирования:
    1. Описательное
    2. Аналитическое
    3. Исполняемое
15. Диаграммы Оркестровки - схемы потоков работ (Orchestration) в BPMN. Семантика диаграммы оркестровки. Типы Процессов. Элементы Процесса. Правила соединения потоков операций.
  1. Что такое.
  2. Виды процессов: Линейные, Интерактивные, Условные, Итеративные, Параллельные.
16. Диаграммы Взаимодействия (Collaboration) в BPMN. Элементы Взаимодействия. Семантика Взаимодействия. Правила соединения потоков сообщений.
  1. Что такое?
17. Диаграммы Диалогов (Conversation) в BPMN. Семантика Диалогов. Элементы Диалогов.
  1. Что такое?
  2. Как выглядит?
  3. Зачем?

18. Диаграммы Хореографии (Choreography) в BPMN. Типы Хореографий. Элементы Хореографии. Семантика Хореографии.
    1. Что такое?
    2. Как регулирует подпроцессы?
    3. Элементы.
    4. Семантика.
  19. Назначение Подпроцессов (Sub-Process) в BPMN. Классификация. Правила соединения с другими элементами моделей. Маркеры подпроцессов.
    1. Что такое подпроцесс
    2. Маркеры подпроцессов
    3. Транзакция
    4. Классификация
  20. Паттерны (шаблоны) различных аспектов BPMN моделей, организующие: потоки управления, данные, ресурсы, обработку исключений, события процессов.
    1. Паттерны соединения, синхронизации, параллельного исполнения.
    2. Структурированные паттерны
- 

# 1. Разработка требований к информационным процессам и системам. Уровни и типы требований. Требования в аспекте различных категорий проектов.

## Разработка требований к информационным процессам и системам

Разработка требований к информационным процессам и системам включает в себя определение и документирование спецификаций, которые должны быть реализованы. Этот процесс важен для обеспечения того, чтобы конечная система соответствовала ожиданиям пользователей и требованиям бизнеса.

## Уровни и типы требований

### 1. Уровни требований:

- **Бизнес-требования:** Описывают цели и задачи, которые система должна решать.
- **Пользовательские требования:** Описывают, что система должна делать с точки зрения пользователя.
- **Системные требования:** Описывают функциональные и нефункциональные аспекты системы.

- **Технические требования:** Описывают конкретные технические спецификации и ограничения.

## 2. Типы требований:

- **Функциональные требования:** Описывают, что система должна делать (например, функции, задачи).
- **Нефункциональные требования:** Описывают, как система должна работать (например, производительность, безопасность, удобство использования).

# Требования в аспекте различных категорий проектов

## 1. Проекты разработки новых систем:

- Требуют детального анализа и документирования всех уровней требований.
- Включают бизнес-требования, пользовательские требования, системные и технические требования.

## 2. Проекты модернизации существующих систем:

- Фокусируются на изменении или улучшении существующих функций.
- Включают анализ текущих требований и определение новых или измененных требований.

## 3. Проекты интеграции систем:

- Требуют анализа требований к взаимодействию между различными системами.
- Включают интерфейсные требования, требования к обмену данными и совместимости.

## 4. Проекты миграции данных:

- Фокусируются на переносе данных из одной системы в другую.
- Включают требования к качеству данных, безопасности и целостности данных.

## 5. Проекты по улучшению производительности:

- Требуют анализа текущей производительности и определения требований к улучшению.
- Включают нефункциональные требования, такие как время отклика, пропускная способность и масштабируемость.

# Способы представления требований

## 1. Документация:

- Используется четко структурированный естественный язык.
- Включает спецификации требований, пользовательские истории, сценарии использования.

## 2. Графические модели:

- Иллюстрируют процессы преобразования, состояния системы и их изменения.

- Включают диаграммы потоков данных, диаграммы состояний, диаграммы классов.

### 3. Формальные спецификации:

- Требования определены с помощью математически точных, формальных логических языков.
- Включают использование языков спецификаций, таких как Z, VDM, Alloy.

Эти аспекты помогают обеспечить полное и точное понимание требований, что является ключевым для успешной разработки информационных систем.

## 2. Практические приемы разработки требований. Каркас процесса создания требований: выявление требований, анализ требований, спецификация требований, проверка требований, управление требованиями.

### Каркас процесса создания требований

#### 1. Выявление требований

- **Сбор требований:** Проводится через интервью, анкетирование, воркшопы и анализ документов.
- **Идентификация стейкхолдеров:** Определение всех заинтересованных сторон, включая пользователей, заказчиков, разработчиков и других участников проекта.
- **Мозговой штурм:** Групповые сессии для генерации идей и требований.
- **Анализ бизнес-процессов:** Изучение текущих бизнес-процессов для выявления потребностей и проблем.

#### 2. Анализ требований

- **Приоритизация требований:** Определение важности и срочности каждого требования.
- **Моделирование требований:** Использование диаграмм и моделей для визуализации требований (например, диаграммы потоков данных, диаграммы состояний).
- **Анализ затрагиваемых систем:** Оценка влияния новых требований на существующие системы и процессы.
- **Анализ рисков:** Идентификация и оценка рисков, связанных с реализацией требований.

#### 3. Спецификация требований

- **Документирование требований:** Создание спецификаций требований, включающих функциональные и нефункциональные требования.
- **Использование шаблонов:** Применение стандартных шаблонов для документирования требований.
- **Формальные спецификации:** Использование формальных языков для точного описания требований.
- **Прототипирование:** Создание прототипов для визуализации и уточнения требований.

#### 4. Проверка требований

- **Ревью и инспекции:** Проведение ревью и инспекций документов требований для выявления ошибок и недочетов.
- **Тестирование требований:** Разработка тест-кейсов для проверки корректности и полноты требований.
- **Валидация с заинтересованными сторонами:** Подтверждение требований с участием стейкхолдеров для убеждения в их правильности и полноте.
- **Симуляции и моделирование:** Использование симуляций для проверки поведения системы в различных сценариях.

#### 5. Управление требованиями

- **Трейнинг требований:** Отслеживание изменений и статуса требований на протяжении всего жизненного цикла проекта.
- **Управление изменениями:** Процесс управления изменениями требований, включая оценку влияния и утверждение изменений.
- **Инструменты управления требованиями:** Использование специализированных инструментов для управления требованиями (например, JIRA, DOORS).
- **Отчетность и мониторинг:** Регулярное создание отчетов и мониторинг статуса требований для обеспечения прозрачности и контроля.

### 3. Спецификация бизнес-требований.

**Концепция продукта и границы проекта.**

**Способы представления границ проекта.**

**Способы представления концепции проекта.**

**Применение бизнес-целей для определения момента завершения проекта.**

**Спецификация бизнес-требований**

**Концепция продукта и границы проекта**

**Концепция продукта:**

- Описывает основную идею и цель продукта.
- Включает видение продукта, его функции и преимущества.
- Определяет, какие проблемы решает продукт и какую ценность он приносит.

### **Границы проекта:**

- Определяют, что включено и что исключено из проекта.
- Устанавливают рамки для ожиданий и ограничений.
- Помогают избежать расширения объема работ (scope creep).

## **Способы представления границ проекта**

### **1. Диаграммы контекста:**

- Показывают взаимодействие системы с внешними элементами.
- Определяют границы системы и ее окружение.

### **2. Матрицы трейсинга требований:**

- Связывают бизнес-требования с функциональными и нефункциональными требованиями.
- Помогают отслеживать изменения и их влияние.

### **3. Документы требований:**

- Включают разделы, описывающие границы проекта.
- Определяют, что входит в проект, а что нет.

## **Способы представления концепции проекта**

### **1. Документы видения:**

- Описывают цели, задачи и ожидаемые результаты проекта.
- Включают бизнес-цели и стратегические направления.

### **2. Презентации и питчи:**

- Визуальные и устные представления концепции проекта.
- Используются для привлечения стейкхолдеров и получения поддержки.

### **3. Прототипы и макеты:**

- Визуализация концепции продукта.
- Помогают стейкхолдерам лучше понять идею и дать обратную связь.

## **Применение бизнес-целей для определения момента завершения проекта**

### **1. Определение критериев успеха:**

- Установление конкретных, измеримых критериев, которые должны быть достигнуты.
- Включают ключевые показатели эффективности (KPI).



## 2. План завершения проекта:

- Определение этапов и задач, которые должны быть выполнены для завершения проекта.
- Включает тестирование, валидацию и подтверждение требований.

## 3. Оценка достижения целей:

- Регулярная проверка прогресса по достижению бизнес-целей.
- Использование отчетов и метрик для мониторинга и корректировки плана.

## 4. Формальное закрытие проекта:

- Проведение финальных ревью и инспекций.
- Подготовка отчетов о завершении и передача результатов стейкхолдерам.

# 4. Выявление и документирование требований пользователей. Методы выявления требований пользователей и их особенности. Классификация предоставляемой пользователями информации и ее роль в проекте требований.

## Выявление и документирование требований пользователей

## Методы выявления требований пользователей и их особенности

### 1. Интервью:

- **Особенности:** Личные или групповые беседы с пользователями для сбора информации.
- **Преимущества:** Позволяют получить глубокое понимание потребностей и ожиданий пользователей.
- **Недостатки:** Могут быть трудоемкими и требовать значительного времени.

### 2. Анкетирование:

- **Особенности:** Использование опросников и анкет для сбора данных от большого числа пользователей.
- **Преимущества:** Быстрое и массовое получение информации.
- **Недостатки:** Могут не дать полного понимания контекста и деталей.

### 3. Воркшопы:

- **Особенности:** Групповые сессии с участием пользователей и стейкхолдеров для совместного обсуждения и выявления требований.
- **Преимущества:** Способствуют коллективному мышлению и генерации идей.

- **Недостатки:** Требуют тщательной подготовки и модерации.

#### 4. Наблюдение:

- **Особенности:** Изучение поведения пользователей в реальных условиях.
- **Преимущества:** Позволяет выявить реальные потребности и проблемы.
- **Недостатки:** Может быть трудоемким и требовать значительного времени.

#### 5. Анализ документов:

- **Особенности:** Изучение существующих документов, отчетов и материалов.
- **Преимущества:** Позволяет использовать уже имеющуюся информацию.
- **Недостатки:** Может не дать полного понимания текущих потребностей.

## Классификация предоставляемой пользователями информации и ее роль в проекте требований

### 1. Функциональные требования:

- **Описание:** Определяют, что система должна делать.
- **Роль:** Формируют основу для разработки функциональных спецификаций и архитектуры системы.

### 2. Нефункциональные требования:

- **Описание:** Определяют, как система должна работать (например, производительность, безопасность).
- **Роль:** Обеспечивают качество и надежность системы.

### 3. Бизнес-требования:

- **Описание:** Определяют цели и задачи, которые система должна решать.
- **Роль:** Формируют стратегическое направление проекта и критерии успеха.

### 4. Пользовательские истории:

- **Описание:** Краткие описания функциональности с точки зрения пользователя.
- **Роль:** Помогают в понимании контекста и приоритетов пользователей.

### 5. Сценарии использования:

- **Описание:** Детальные описания взаимодействия пользователя с системой.
- **Роль:** Способствуют разработке интерфейсов и тестированию.

### 6. Ограничения:

- **Описание:** Определяют рамки и ограничения, в которых должна работать система.
- **Роль:** Обеспечивают соответствие требований реальным условиям и возможностям.

## Документирование требований пользователей

### 1. Спецификации требований:

- **Описание:** Формальные документы, описывающие все требования.

- **Роль:** Основной источник информации для разработчиков и тестировщиков.

## 2. Диаграммы и модели:

- **Описание:** Графические представления требований (например, диаграммы потоков данных, диаграммы состояний).
- **Роль:** Визуализация и упрощение понимания требований.

## 3. Прототипы и макеты:

- **Описание:** Визуальные модели интерфейсов и функциональности.
- **Роль:** Помогают в уточнении и валидации требований.

## 4. Таблицы и матрицы:

- **Описание:** Табличные представления требований и их связей.
- **Роль:** Обеспечивают структурированное и удобное управление требованиями.

# 5. Спецификация функциональных требований. Приемы разработки и управления функциональными требованиями.

## 1. Сбор и анализ требований:

- **Интервью и анкетирование:** Сбор информации от пользователей и стейкхолдеров.
- **Воркшопы и мозговые штурмы:** Групповые сессии для генерации и уточнения требований.
- **Анализ документов:** Изучение существующих документов и материалов.

## 2. Документирование требований:

- **Спецификации требований:** Формальные документы, описывающие все функциональные требования.
- **Пользовательские истории:** Краткие описания функциональности с точки зрения пользователя.
- **Сценарии использования:** Детальные описания взаимодействия пользователя с системой.

## 3. Моделирование требований:

- **Диаграммы потоков данных (DFD):** Визуализация потоков данных и процессов.
- **Диаграммы состояний:** Описание состояний системы и переходов между ними.
- **Диаграммы классов:** Моделирование структуры данных и их взаимодействий.

## 4. Прототипирование:

- **Прототипы интерфейсов:** Визуальные модели интерфейсов для уточнения требований.

- **Макеты:** Статические или интерактивные модели функциональности.
5. **Приоритизация требований:**
- **Метод MoSCoW:** Классификация требований по категориям Must have, Should have, Could have, Won't have.
  - **Матрица приоритетов:** Оценка важности и срочности требований.
6. **Управление требованиями:**
- **Трейсинг требований:** Отслеживание изменений и статуса требований.
  - **Матрицы трейсинга:** Связывание бизнес-требований с функциональными и нефункциональными требованиями.
  - **Инструменты управления требованиями:** Использование специализированных инструментов (например, JIRA, DOORS).
7. **Проверка и валидация требований:**
- **Ревью и инспекции:** Проверка документов требований для выявления ошибок и недочетов.
  - **Тестирование требований:** Разработка тест-кейсов для проверки корректности и полноты требований.
  - **Валидация с заинтересованными сторонами:** Подтверждение требований с участием стейкхолдеров.
8. **Управление изменениями:**
- **Процесс управления изменениями:** Оценка влияния и утверждение изменений.
  - **Документирование изменений:** Обновление спецификаций и документов при изменении требований.
9. **Отчетность и мониторинг:**
- **Регулярные отчеты:** Создание отчетов о статусе требований и прогрессе проекта.
  - **Мониторинг:** Постоянное отслеживание и контроль выполнения требований.

## **6. Моделирование требований. Варианты комбинации текстовых и визуальных способов представления требований на различных уровнях абстракции.**

### **Варианты комбинации текстовых и визуальных способов представления требований**

1. **Высокий уровень абстракции (Концептуальный уровень):**
- **Текстовые способы:**
    - **Документы видения:** Описание целей, задач и ожидаемых результатов проекта.

- **Бизнес-требования:** Определение целей и задач, которые система должна решать.
  - **Визуальные способы:**
    - **Диаграммы контекста:** Показывают взаимодействие системы с внешними элементами.
    - **Диаграммы высокого уровня:** Общие схемы и блок-диаграммы для представления основных компонентов системы.
2. **Средний уровень абстракции (Функциональный уровень):**
- **Текстовые способы:**
    - **Спецификации требований:** Формальные документы, описывающие функциональные требования.
    - **Пользовательские истории:** Краткие описания функциональности с точки зрения пользователя.
  - **Визуальные способы:**
    - **Диаграммы потоков данных (DFD):** Визуализация потоков данных и процессов.
    - **Диаграммы состояний:** Описание состояний системы и переходов между ними.
    - **Диаграммы классов:** Моделирование структуры данных и их взаимодействий.
3. **Низкий уровень абстракции (Технический уровень):**
- **Текстовые способы:**
    - **Технические спецификации:** Детальные описания технических требований и ограничений.
    - **Сценарии использования:** Детальные описания взаимодействия пользователя с системой.
  - **Визуальные способы:**
    - **Диаграммы последовательностей:** Описание взаимодействий между объектами в хронологическом порядке.
    - **Диаграммы активности:** Моделирование потоков управления и данных.
    - **Диаграммы компонентов:** Описание физических и логических компонентов системы.

## Примеры комбинаций

1. **Концептуальный уровень:**
- **Документ видения + Диаграмма контекста:** Описание целей проекта и взаимодействия системы с внешними элементами.
  - **Бизнес-требования + Диаграмма высокого уровня:** Определение целей и задач, которые система должна решать, и общая схема компонентов системы.

## 2. Функциональный уровень:

- **Спецификация требований + Диаграмма потоков данных (DFD):** Формальное описание функциональных требований и визуализация потоков данных.
- **Пользовательские истории + Диаграмма состояний:** Краткие описания функциональности и моделирование состояний системы.

## 3. Технический уровень:

- **Техническая спецификация + Диаграмма последовательностей:** Детальное описание технических требований и описание взаимодействий между объектами.
- **Сценарии использования + Диаграмма активности:** Детальные описания взаимодействия пользователя с системой и моделирование потоков управления и данных.

# 7. Определение требований к данным.

## Приемы и механизмы разработки требований к данным: словарь данных, анализ данных, спецификация отчетов.

### Приемы и механизмы разработки требований к данным

#### 1. Словарь данных:

- **Описание:** Централизованный реестр всех данных, используемых в системе.
- **Содержание:** Включает определения данных, их типы, форматы, ограничения и связи.
- **Преимущества:** Обеспечивает единообразие и стандартизацию данных, упрощает их управление и использование.
- **Пример:** Определение полей таблицы, их типов данных (например, строка, число) и ограничений (например, обязательное поле, уникальность).

#### 2. Анализ данных:

- **Описание:** Процесс изучения существующих данных и их структуры.
- **Методы:**
  - **Анализ существующих баз данных:** Изучение текущих схем баз данных и таблиц.
  - **Анализ бизнес-процессов:** Определение данных, необходимых для поддержки бизнес-процессов.
  - **Анализ источников данных:** Идентификация внешних и внутренних источников данных.
- **Преимущества:** Помогает понять текущие данные и их использование, выявить пробелы и необходимость новых данных.

- **Пример:** Анализ текущей базы данных для определения, какие данные уже существуют и какие новые данные необходимы.

### 3. Спецификация отчетов:

- **Описание:** Определение требований к отчетам и аналитическим данным, которые система должна генерировать.
- **Содержание:** Включает описание типов отчетов, их структуры, форматов и частоты генерации.
- **Методы:**
  - **Определение пользовательских потребностей:** Сбор требований от пользователей и стейкхолдеров.
  - **Прототипирование отчетов:** Создание макетов отчетов для уточнения требований.
  - **Спецификация форматов:** Определение форматов отчетов (например, PDF, Excel) и их структуры.
- **Преимущества:** Обеспечивает точное и полезное представление данных для принятия решений.
- **Пример:** Спецификация ежемесячного отчета о продажах, включающая структуру отчета, формат (PDF) и частоту генерации (ежемесячно).

## Примеры приемов и механизмов

### 1. Словарь данных:

- **Поле:** "CustomerID"
- **Тип данных:** Число (INT)
- **Ограничения:** Обязательное поле, уникальное значение
- **Описание:** Уникальный идентификатор клиента

### 2. Анализ данных:

- **Источник данных:** Текущая база данных клиентов
- **Анализ:** Изучение таблицы "Customers" для определения существующих полей и их использования
- **Вывод:** Необходимо добавить новое поле "CustomerEmail" для хранения адресов электронной почты клиентов

### 3. Спецификация отчетов:

- **Тип отчета:** Ежемесячный отчет о продажах
- **Структура:** Таблица с колонками "ProductID", "ProductName", "QuantitySold", "TotalRevenue"
- **Формат:** PDF
- **Частота:** Ежемесячно, 1-е число каждого месяца

## 8. Документирование требований.

### Спецификация требований к ПО: требования к

**именованию, шаблоны спецификации требований, уровни детализации требований, способы визуализации, приемы по проверке неполноты.**

## **Спецификация требований к ПО**

### **Требования к именованию**

#### **1. Единообразие:**

- Использование стандартных соглашений для именования требований.
- Пример: "REQ-001", "FUNC-002", "NONFUNC-003".

#### **2. Ясность и краткость:**

- Имена должны быть понятными и лаконичными.
- Пример: "UserAuthentication" вместо "RequirementForUserAuthenticationProcess".

#### **3. Уникальность:**

- Каждое требование должно иметь уникальное имя для избежания путаницы.
- Пример: "REQ-001" для первого требования, "REQ-002" для второго.

## **Шаблоны спецификации требований**

#### **1. Общий шаблон:**

- **Идентификатор:** Уникальный номер или имя требования.
- **Название:** Краткое описание требования.
- **Описание:** Детальное описание требования.
- **Приоритет:** Уровень важности требования.
- **Источник:** Откуда поступило требование (например, пользователь, бизнес-аналитик).
- **Зависимости:** Связи с другими требованиями.
- **Критерии приемки:** Условия, при которых требование считается выполненным.

#### **2. Пример шаблона:**

Идентификатор: REQ-001

Название: Пользовательская аутентификация

Описание: Система должна предоставлять возможность аутентификации пользователей с использованием логина и пароля.

Приоритет: Высокий

Источник: Пользователь

Зависимости: REQ-002 (Регистрация пользователей)



Критерии приемки: Пользователь должен успешно аутентифицироваться при вводе правильного логина и пароля.

## Уровни детализации требований

### 1. Высокий уровень (Общие требования):

- Описание основных целей и задач системы.
- Пример: "Система должна обеспечивать управление заказами клиентов."

### 2. Средний уровень (Функциональные требования):

- Детальное описание функций и возможностей системы.
- Пример: "Система должна позволять пользователям создавать, редактировать и удалять заказы."

### 3. Низкий уровень (Технические требования):

- Описание конкретных технических спецификаций и ограничений.
- Пример: "Система должна поддерживать не менее 1000 одновременных пользователей."

## Способы визуализации

### 1. Диаграммы потоков данных (DFD):

- Визуализация потоков данных и процессов в системе.
- Пример: Диаграмма, показывающая, как данные перемещаются между различными компонентами системы.

### 2. Диаграммы состояний:

- Описание состояний системы и переходов между ними.
- Пример: Диаграмма, показывающая состояния заказа (создан, оплачен, отправлен).

### 3. Диаграммы классов:

- Моделирование структуры данных и их взаимодействий.
- Пример: Диаграмма, показывающая классы "Заказ", "Клиент", "Продукт" и их связи.

### 4. Диаграммы последовательностей:

- Описание взаимодействий между объектами в хронологическом порядке.
- Пример: Диаграмма, показывающая последовательность действий при создании заказа.

## Приемы по проверке неполноты

### 1. Ревью и инспекции:

- Проведение регулярных ревью и инспекций документов требований для выявления пробелов и недочетов.

- Пример: Еженедельные встречи команды для обсуждения и проверки требований.

## **2. Тестирование требований:**

- Разработка тест-кейсов для проверки корректности и полноты требований.
- Пример: Создание тест-кейсов для каждого функционального требования.

## **3. Валидация с заинтересованными сторонами:**

- Подтверждение требований с участием стейкхолдеров для убеждения в их правильности и полноте.
- Пример: Проведение встреч с пользователями для обсуждения и подтверждения требований.

## **4. Симуляции и моделирование:**

- Использование симуляций для проверки поведения системы в различных сценариях.
- Пример: Моделирование процесса создания заказа для проверки всех возможных состояний и переходов.

# **9. Прототипы как средство снижения риска. Модели и экспериментальные образцы. Одноразовые и эволюционные прототипы. Бумажные и электронные прототипы.**

## **Модели и экспериментальные образцы**

### **1. Модели:**

- **Описание:** Абстрактные представления системы или ее компонентов.
- **Примеры:** Диаграммы, схемы, макеты.
- **Преимущества:** Помогают визуализировать и понять структуру и поведение системы.

### **2. Экспериментальные образцы:**

- **Описание:** Функциональные версии системы или ее компонентов, предназначенные для тестирования и оценки.
- **Примеры:** Прототипы, демонстрационные версии.
- **Преимущества:** Позволяют протестировать реальные функции и взаимодействия, выявить проблемы на ранних стадиях.

## **Одноразовые и эволюционные прототипы**

### **1. Одноразовые прототипы:**

- **Описание:** Прототипы, которые создаются для одной конкретной цели и затем отбрасываются.

- **Примеры:** Прототипы для демонстрации концепции, прототипы для тестирования пользовательского интерфейса.
- **Преимущества:** Быстрое создание и тестирование идей, минимальные затраты на разработку.

## 2. Эволюционные прототипы:

- **Описание:** Прототипы, которые постепенно развиваются и улучшаются, становясь основой для конечной системы.
- **Примеры:** Прототипы, которые постепенно дополняются новыми функциями и улучшениями.
- **Преимущества:** Позволяют постепенно улучшать систему на основе обратной связи и тестирования.

# Бумажные и электронные прототипы

## 1. Бумажные прототипы:

- **Описание:** Прототипы, созданные на бумаге или других физических носителях.
- **Примеры:** Рисунки, схемы, макеты на бумаге.
- **Преимущества:** Быстрое и дешевое создание, легкость в изменении и обсуждении.
- **Недостатки:** Ограниченная интерактивность, сложность в демонстрации динамических элементов.

## 2. Электронные прототипы:

- **Описание:** Прототипы, созданные с использованием программного обеспечения.
- **Примеры:** Интерактивные макеты, демонстрационные версии программ.
- **Преимущества:** Возможность демонстрации динамических элементов и взаимодействий, легкость в изменении и обновлении.
- **Недостатки:** Могут требовать больше времени и ресурсов для создания.

# Примеры использования прототипов

## 1. Одноразовый бумажный прототип:

- **Цель:** Демонстрация концепции пользовательского интерфейса.
- **Пример:** Рисунок интерфейса приложения на бумаге для обсуждения с пользователями.

## 2. Эволюционный электронный прототип:

- **Цель:** Постепенное развитие функциональности системы.
- **Пример:** Интерактивный макет веб-приложения, который постепенно дополняется новыми функциями и улучшениями.

## 3. Экспериментальный образец:

- **Цель:** Тестирование технических решений.

- **Пример:** Прототип системы для тестирования производительности и масштабируемости.

## **10. Требования в проектах определенных классов. Особенности различных типов проектов. Проекты автоматизации бизнес-процессов: особенности и отличия от проектов других классов.**

### **Модели и экспериментальные образцы**

#### **1. Модели:**

- **Описание:** Абстрактные представления системы или ее компонентов.
- **Примеры:** Диаграммы, схемы, макеты.
- **Преимущества:** Помогают визуализировать и понять структуру и поведение системы.

#### **2. Экспериментальные образцы:**

- **Описание:** Функциональные версии системы или ее компонентов, предназначенные для тестирования и оценки.
- **Примеры:** Прототипы, демонстрационные версии.
- **Преимущества:** Позволяют протестировать реальные функции и взаимодействия, выявить проблемы на ранних стадиях.

### **Одноразовые и эволюционные прототипы**

#### **1. Одноразовые прототипы:**

- **Описание:** Прототипы, которые создаются для одной конкретной цели и затем отбрасываются.
- **Примеры:** Прототипы для демонстрации концепции, прототипы для тестирования пользовательского интерфейса.
- **Преимущества:** Быстрое создание и тестирование идей, минимальные затраты на разработку.

#### **2. Эволюционные прототипы:**

- **Описание:** Прототипы, которые постепенно развиваются и улучшаются, становясь основой для конечной системы.
- **Примеры:** Прототипы, которые постепенно дополняются новыми функциями и улучшениями.
- **Преимущества:** Позволяют постепенно улучшать систему на основе обратной связи и тестирования.

### **Бумажные и электронные прототипы**

## 1. Бумажные прототипы:

- **Описание:** Прототипы, созданные на бумаге или других физических носителях.
- **Примеры:** Рисунки, схемы, макеты на бумаге.
- **Преимущества:** Быстрое и дешевое создание, легкость в изменении и обсуждении.
- **Недостатки:** Ограниченная интерактивность, сложность в демонстрации динамических элементов.

## 2. Электронные прототипы:

- **Описание:** Прототипы, созданные с использованием программного обеспечения.
- **Примеры:** Интерактивные макеты, демонстрационные версии программ.
- **Преимущества:** Возможность демонстрации динамических элементов и взаимодействий, легкость в изменении и обновлении.
- **Недостатки:** Могут требовать больше времени и ресурсов для создания.

# Примеры использования прототипов

## 1. Одноразовый бумажный прототип:

- **Цель:** Демонстрация концепции пользовательского интерфейса.
- **Пример:** Рисунок интерфейса приложения на бумаге для обсуждения с пользователями.

## 2. Эволюционный электронный прототип:

- **Цель:** Постепенное развитие функциональности системы.
- **Пример:** Интерактивный макет веб-приложения, который постепенно дополняется новыми функциями и улучшениями.

## 3. Экспериментальный образец:

- **Цель:** Тестирование технических решений.
- **Пример:** Прототип системы для тестирования производительности и масштабируемости.

# 11. Требования к интерфейсам. Основные эргономические принципы создания интерактивных систем в общем виде.

## Особенности различных типов проектов

### 1. Проекты разработки новых систем:

- **Особенности:** Требуют детального анализа и документирования всех уровней требований.

- **Требования:** Включают бизнес-требования, пользовательские требования, системные и технические требования.
  - **Пример:** Разработка новой CRM-системы для управления взаимоотношениями с клиентами.
2. **Проекты модернизации существующих систем:**
- **Особенности:** Фокусируются на изменении или улучшении существующих функций.
  - **Требования:** Включают анализ текущих требований и определение новых или измененных требований.
  - **Пример:** Обновление ERP-системы для поддержки новых бизнес-процессов.
3. **Проекты интеграции систем:**
- **Особенности:** Требуют анализа требований к взаимодействию между различными системами.
  - **Требования:** Включают интерфейсные требования, требования к обмену данными и совместимости.
  - **Пример:** Интеграция системы управления складом с системой управления заказами.
4. **Проекты миграции данных:**
- **Особенности:** Фокусируются на переносе данных из одной системы в другую.
  - **Требования:** Включают требования к качеству данных, безопасности и целостности данных.
  - **Пример:** Миграция данных с устаревшей системы на новую облачную платформу.
5. **Проекты по улучшению производительности:**
- **Особенности:** Требуют анализа текущей производительности и определения требований к улучшению.
  - **Требования:** Включают нефункциональные требования, такие как время отклика, пропускная способность и масштабируемость.
  - **Пример:** Оптимизация производительности веб-приложения для улучшения пользовательского опыта.

## **Проекты автоматизации бизнес-процессов: особенности и отличия от проектов других классов**

**Особенности проектов автоматизации бизнес-процессов:**

1. **Фокус на бизнес-процессы:**
  - **Описание:** Основное внимание уделяется автоматизации и оптимизации существующих бизнес-процессов.
  - **Пример:** Автоматизация процесса обработки заказов в интернет-магазине.
2. **Взаимодействие с пользователями:**

- **Описание:** Требуется тесного взаимодействия с пользователями для понимания текущих процессов и ожиданий.
- **Пример:** Проведение воркшопов с сотрудниками для выявления текущих проблем и потребностей.

### 3. Интеграция с существующими системами:

- **Описание:** Часто требует интеграции с уже существующими системами и процессами.
- **Пример:** Интеграция новой системы управления заказами с существующей системой управления складом.

### 4. Требования к данным:

- **Описание:** Включают требования к сбору, хранению и обработке данных, необходимых для автоматизации процессов.
- **Пример:** Требования к хранению истории заказов и отчетов о выполнении.

### 5. Требования к безопасности и соответствию:

- **Описание:** Включают требования к безопасности данных и соответствию нормативным требованиям.
- **Пример:** Обеспечение безопасности данных клиентов и соответствие GDPR.

## Отличия от проектов других классов:

### 1. Фокус на процессы, а не на функции:

- **Отличие:** В проектах автоматизации бизнес-процессов основное внимание уделяется процессам, а не отдельным функциям системы.
- **Пример:** Автоматизация процесса обработки заказов включает в себя несколько функций, таких как прием заказа, обработка платежа и отправка товара.

### 2. Тесное взаимодействие с пользователями:

- **Отличие:** Требуется более тесного взаимодействия с пользователями для понимания текущих процессов и ожиданий.
- **Пример:** Проведение интервью и воркшопов с сотрудниками для выявления текущих проблем и потребностей.

### 3. Интеграция с существующими системами:

- **Отличие:** Часто требует интеграции с уже существующими системами и процессами.
- **Пример:** Интеграция новой системы управления заказами с существующей системой управления складом.

### 4. Требования к данным:

- **Отличие:** Включают требования к сбору, хранению и обработке данных, необходимых для автоматизации процессов.
- **Пример:** Требования к хранению истории заказов и отчетов о выполнении.

### 5. Требования к безопасности и соответствию:

- **Отличие:** Включают требования к безопасности данных и соответствию нормативным требованиям.
- **Пример:** Обеспечение безопасности данных клиентов и соответствие GDPR.

## 12. Визуализация различных уровней требований: диаграммы, модели, схемы.

### Высокий уровень абстракции (Концептуальный уровень)

#### 1. Диаграммы контекста:

- **Описание:** Показывают взаимодействие системы с внешними элементами.
- **Пример:** Диаграмма, показывающая, как система управления заказами взаимодействует с клиентами, поставщиками и внутренними системами.

#### 2. Диаграммы высокого уровня:

- **Описание:** Общие схемы и блок-диаграммы для представления основных компонентов системы.
- **Пример:** Блок-диаграмма, показывающая основные модули системы управления заказами (например, модуль обработки заказов, модуль управления складом).

### Средний уровень абстракции (Функциональный уровень)

#### 1. Диаграммы потоков данных (DFD):

- **Описание:** Визуализация потоков данных и процессов в системе.
- **Пример:** DFD, показывающая, как данные о заказе перемещаются между модулями системы (например, от модуля приема заказов к модулю обработки платежей).

#### 2. Диаграммы состояний:

- **Описание:** Описание состояний системы и переходов между ними.
- **Пример:** Диаграмма состояний заказа, показывающая состояния "создан", "оплачен", "отправлен" и переходы между ними.

#### 3. Диаграммы классов:

- **Описание:** Моделирование структуры данных и их взаимодействий.
- **Пример:** Диаграмма классов, показывающая классы "Заказ", "Клиент", "Продукт" и их связи.

### Низкий уровень абстракции (Технический уровень)

#### 1. Диаграммы последовательностей:

- **Описание:** Описание взаимодействий между объектами в хронологическом порядке.



- **Пример:** Диаграмма последовательностей, показывающая последовательность действий при создании заказа.

## 2. Диаграммы активности:

- **Описание:** Моделирование потоков управления и данных.
- **Пример:** Диаграмма активности, показывающая процесс обработки заказа, включая действия и решения.

## 3. Диаграммы компонентов:

- **Описание:** Описание физических и логических компонентов системы.
- **Пример:** Диаграмма компонентов, показывающая, как различные модули системы взаимодействуют друг с другом.

# Примеры использования диаграмм, моделей и схем

## 1. Диаграмма контекста:

- **Пример:** Система управления заказами взаимодействует с клиентами (через веб-интерфейс), поставщиками (через API) и внутренними системами (например, система управления складом).

## 2. Диаграмма потоков данных (DFD):

- **Пример:** DFD показывает, как данные о заказе перемещаются от модуля приема заказов к модулю обработки платежей, а затем к модулю управления складом.

## 3. Диаграмма состояний:

- **Пример:** Диаграмма состояний заказа показывает состояния "создан", "оплачен", "отправлен" и переходы между ними, такие как "оплатить заказ" и "отправить заказ".

## 4. Диаграмма классов:

- **Пример:** Диаграмма классов показывает классы "Заказ", "Клиент", "Продукт" и их связи, такие как "Заказ содержит Продукты" и "Клиент создает Заказ".

## 5. Диаграмма последовательностей:

- **Пример:** Диаграмма последовательностей показывает последовательность действий при создании заказа, включая взаимодействия между клиентом, системой управления заказами и системой управления складом.

## 6. Диаграмма активности:

- **Пример:** Диаграмма активности показывает процесс обработки заказа, включая действия "принять заказ", "проверить наличие товара", "оплатить заказ" и "отправить заказ".

## 7. Диаграмма компонентов:

- **Пример:** Диаграмма компонентов показывает, как модули системы управления заказами взаимодействуют друг с другом, включая модули приема заказов, обработки платежей и управления складом.

# 13. Различные подходы в спецификации требований: примеры авторских методик, техническое задание по ГОСТ 34.

## Примеры авторских методик

### 1. Методика Volere:

- **Описание:** Разработана Сюзанной Робертс. Фокусируется на структурированном подходе к сбору и документированию требований.
- **Особенности:** Включает шаблоны для документирования требований, методы анализа и проверки требований.
- **Пример:** Использование шаблона Volere для документирования функциональных и нефункциональных требований к системе управления проектами.

### 2. Методика Use Case Driven:

- **Описание:** Основана на использовании сценариев использования (use cases) для описания требований.
- **Особенности:** Фокусируется на описании взаимодействий между пользователями и системой.
- **Пример:** Разработка сценариев использования для системы управления заказами, включая сценарии "Создание заказа", "Оплата заказа", "Отмена заказа".

### 3. Методика Agile (Scrum):

- **Описание:** Используется в Agile-разработке, фокусируется на итеративном подходе к сбору и документированию требований.
- **Особенности:** Включает использование пользовательских историй (user stories) и бэклога продукта (product backlog).
- **Пример:** Создание пользовательских историй для функциональности системы управления задачами, таких как "Как пользователь, я хочу создавать задачи, чтобы управлять своим временем".

## Техническое задание по ГОСТ 34

**ГОСТ 34** (ГОСТ 34.003-90, ГОСТ 34.602-89 и другие стандарты серии ГОСТ 34) регламентирует требования к документации на автоматизированные системы (АС) и программные средства (ПС). Техническое задание (ТЗ) по ГОСТ 34 включает следующие разделы:

### 1. Введение:

- **Описание:** Общие сведения о проекте, цели и задачи разработки.
- **Пример:** Описание цели создания системы управления заказами, основные задачи и ожидаемые результаты.

## 2. Назначение разработки:

- **Описание:** Описание функционального назначения системы.
- **Пример:** Система управления заказами предназначена для автоматизации процесса приема, обработки и выполнения заказов клиентов.

## 3. Требования к системе:

- **Описание:** Описание функциональных и нефункциональных требований.
- **Пример:** Система должна поддерживать создание, редактирование и удаление заказов, обеспечивать безопасность данных и высокую производительность.

## 4. Требования к программному обеспечению:

- **Описание:** Спецификации требований к ПО, включая языки программирования, платформы и инструменты.
- **Пример:** Система должна быть разработана на языке Java, работать на платформе Windows и использовать базу данных MySQL.

## 5. Требования к информационной и программной совместимости:

- **Описание:** Требования к интеграции с другими системами и программным обеспечением.
- **Пример:** Система должна интегрироваться с существующей системой управления складом через API.

## 6. Требования к надежности:

- **Описание:** Требования к надежности и отказоустойчивости системы.
- **Пример:** Система должна обеспечивать доступность 99.9% и восстановление после сбоев в течение 1 часа.

## 7. Требования к безопасности информации:

- **Описание:** Требования к защите данных и обеспечению безопасности.
- **Пример:** Система должна обеспечивать шифрование данных, аутентификацию пользователей и защиту от несанкционированного доступа.

## 8. Требования к эргономике и удобству эксплуатации:

- **Описание:** Требования к пользовательскому интерфейсу и удобству использования.
- **Пример:** Интерфейс системы должен быть интуитивно понятным и удобным для пользователей с различным уровнем подготовки.

## 9. Требования к сопровождающей документации:

- **Описание:** Требования к документации, сопровождающей систему.
- **Пример:** Документация должна включать руководство пользователя, техническую документацию и инструкции по установке и настройке.

## 10. Требования к организационно-техническим мероприятиям:

- **Описание:** Требования к организационным и техническим мероприятиям, необходимым для разработки и внедрения системы.
- **Пример:** Требуется проведение обучения пользователей, тестирование системы и подготовка плана внедрения.

# 14. Нотация моделирования бизнес-процессов (BPMN): область применения. Основные категории элементов нотации. Категории диаграмм бизнес-процессов в BPMN. Уровни моделирования бизнес-процессов.

## Нотация моделирования бизнес-процессов (BPMN): область применения

**BPMN (Business Process Model and Notation)** — это стандартная нотация для моделирования бизнес-процессов, которая предоставляет графическое представление потоков работ в бизнес-процессах. BPMN используется для визуализации и анализа бизнес-процессов, а также для их автоматизации и оптимизации.

### Область применения BPMN:

- **Анализ и оптимизация бизнес-процессов:** Визуализация текущих процессов для выявления узких мест и возможностей для улучшения.
- **Документирование бизнес-процессов:** Создание документации для обучения новых сотрудников и обеспечения единообразия в выполнении процессов.
- **Автоматизация бизнес-процессов:** Использование BPMN для разработки и внедрения автоматизированных систем управления бизнес-процессами (BPM-систем).
- **Коммуникация между стейкхолдерами:** Обеспечение общего понимания бизнес-процессов между различными участниками проекта, включая бизнес-аналитиков, разработчиков и менеджеров.

## Основные категории элементов нотации BPMN

### 1. Потоки (Flow Objects):

- **События (Events):** Начало, конец или промежуточные события в процессе (например, начало процесса, завершение процесса).
- **Действия (Activities):** Задачи или подпроцессы, выполняемые в рамках процесса (например, задача "Обработка заказа").
- **Шлюзы (Gateways):** Управление потоком процесса, включая ветвления, слияния и условия (например, разветвление на основе условия).

### 2. Связи (Connecting Objects):

- **Последовательные потоки (Sequence Flows):** Определяют последовательность выполнения действий.
- **Сообщения (Message Flows):** Определяют обмен сообщениями между участниками процесса.

- **Ассоциации (Associations):** Связывают артефакты с потоками или действиями.

### 3. Плавающие артефакты (Swimlanes):

- **Пулы (Pools):** Представляют участников процесса (например, отделы или организации).
- **Дорожки (Lanes):** Подразделения внутри пула, представляющие роли или функции.

### 4. Артефакты (Artifacts):

- **Данные (Data Objects):** Информация, используемая или создаваемая в процессе (например, документы, данные).
- **Группы (Groups):** Визуальное группирование элементов для улучшения читаемости диаграммы.
- **Аннотации (Annotations):** Дополнительная информация или комментарии к элементам диаграммы.

## Категории диаграмм бизнес-процессов в BPMN

### 1. Диаграммы высокого уровня (High-Level Diagrams):

- **Описание:** Представляют общий обзор бизнес-процессов, включая основные этапы и участников.
- **Пример:** Диаграмма, показывающая основные этапы процесса обработки заказа, такие как "Прием заказа", "Обработка платежа", "Отправка товара".

### 2. Детализированные диаграммы (Detailed Diagrams):

- **Описание:** Представляют подробное описание каждого этапа процесса, включая задачи, условия и потоки данных.
- **Пример:** Диаграмма, показывающая детализированные шаги процесса "Обработка платежа", включая проверку платежных данных, подтверждение платежа и обновление статуса заказа.

### 3. Диаграммы взаимодействия (Collaboration Diagrams):

- **Описание:** Представляют взаимодействие между различными участниками процесса, включая обмен сообщениями и данными.
- **Пример:** Диаграмма, показывающая взаимодействие между отделом продаж, отделом логистики и клиентом в процессе обработки заказа.

## Уровни моделирования бизнес-процессов

### 1. Высокий уровень (High-Level):

- **Описание:** Общий обзор бизнес-процессов, включая основные этапы и участников.
- **Пример:** Диаграмма, показывающая основные этапы процесса обработки заказа.

### 2. Средний уровень (Mid-Level):

- **Описание:** Детализированное описание каждого этапа процесса, включая задачи и условия.
- **Пример:** Диаграмма, показывающая детализированные шаги процесса "Обработка платежа".

### 3. Низкий уровень (Low-Level):

- **Описание:** Подробное описание каждой задачи, включая конкретные действия, условия и потоки данных.
- **Пример:** Диаграмма, показывающая конкретные действия и условия в процессе "Проверка платежных данных".

## 15. Диаграммы Оркестровки - схемы потоков работ (Orchestration) в BPMN. Семантика диаграммы оркестровки. Типы Процессов. Элементы Процесса. Правила соединения потоков операций.

### Диаграммы Оркестровки - схемы потоков работ (Orchestration) в BPMN

Диаграммы оркестровки в BPMN представляют собой схемы потоков работ, которые описывают последовательность действий и взаимодействий в бизнес-процессе. Эти диаграммы помогают визуализировать и управлять сложными процессами, включая взаимодействие между различными участниками и системами.

### Семантика диаграммы оркестровки

Семантика диаграммы оркестровки в BPMN включает в себя следующие ключевые элементы:

#### 1. Пулы (Pools):

- **Описание:** Представляют участников процесса, такие как отделы или организации.
- **Пример:** Пул "Отдел продаж" и пул "Отдел логистики".

#### 2. Дорожки (Lanes):

- **Описание:** Подразделения внутри пула, представляющие роли или функции.
- **Пример:** Дорожка "Менеджер по продажам" внутри пула "Отдел продаж".

#### 3. События (Events):

- **Описание:** Начало, конец или промежуточные события в процессе.
- **Пример:** Событие "Начало процесса" или событие "Завершение процесса".

#### 4. Действия (Activities):

- **Описание:** Задачи или подпроцессы, выполняемые в рамках процесса.

- **Пример:** Задача "Обработка заказа" или подпроцесс "Проверка платежа".

#### 5. Шлюзы (Gateways):

- **Описание:** Управление потоком процесса, включая ветвления, слияния и условия.
- **Пример:** Шлюз "Разветвление на основе условия" или шлюз "Слияние потоков".

#### 6. Последовательные потоки (Sequence Flows):

- **Описание:** Определяют последовательность выполнения действий.
- **Пример:** Поток от задачи "Прием заказа" к задаче "Обработка платежа".

#### 7. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между участниками процесса.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

## Типы Процессов

#### 1. Процессы (Processes):

- **Описание:** Основные бизнес-процессы, которые моделируются и управляются.
- **Пример:** Процесс "Обработка заказа".

#### 2. Подпроцессы (Sub-Processes):

- **Описание:** Части процесса, которые могут быть выделены и моделированы отдельно.
- **Пример:** Подпроцесс "Проверка платежа" внутри процесса "Обработка заказа".

## Элементы Процесса

#### 1. События (Events):

- **Начальные события (Start Events):** Иницируют начало процесса.
- **Промежуточные события (Intermediate Events):** Происходят в середине процесса.
- **Конечные события (End Events):** Завершают процесс.

#### 2. Действия (Activities):

- **Задачи (Tasks):** Основные единицы работы в процессе.
- **Подпроцессы (Sub-Processes):** Части процесса, которые могут быть выделены и моделированы отдельно.

#### 3. Шлюзы (Gateways):

- **Разветвления (Diverging Gateways):** Управление ветвлениями потока.
- **Слияния (Converging Gateways):** Управление слиянием потоков.

## Правила соединения потоков операций

### 1. Последовательные потоки (Sequence Flows):

- **Описание:** Определяют последовательность выполнения действий.
- **Правила:**
  - Поток должен начинаться с события или действия и заканчиваться событием или действием.
  - Поток не может начинаться или заканчиваться шлюзом.

### 2. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между участниками процесса.
- **Правила:**
  - Сообщение должно начинаться и заканчиваться в разных пулах.
  - Сообщение не может начинаться или заканчиваться шлюзом.

### 3. Ассоциации (Associations):

- **Описание:** Связывают артефакты с потоками или действиями.
- **Правила:**
  - Ассоциация может связывать артефакт с любым элементом потока (событие, действие, шлюз).
  - Ассоциация не может связывать два артефакта.

## Пример диаграммы оркестровки

```
[Start Event] --> [Task: Прием заказа] --> [Gateway: Проверка наличия  
товара]  
  
|  
|----> [Task: Уведомление  
о нехватке товара] --> [End Event]  
  
|  
|----> [Task: Обработка  
платежа] --> [Task: Отправка товара] --> [End Event]
```

В этом примере:

- **Start Event:** Начало процесса.
- **Task: Прием заказа:** Задача по приему заказа.
- **Gateway: Проверка наличия товара:** Шлюз для проверки наличия товара.
- **Task: Уведомление о нехватке товара:** Задача по уведомлению о нехватке товара.
- **Task: Обработка платежа:** Задача по обработке платежа.
- **Task: Отправка товара:** Задача по отправке товара.
- **End Event:** Завершение процесса.

Эти элементы и правила помогают создать четкую и структурированную диаграмму оркестровки, которая визуализирует и управляет сложными бизнес-процессами.



# 16. Диаграммы Взаимодействия (Collaboration) в BPMN. Элементы Взаимодействия. Семантика Взаимодействия. Правила соединения потоков сообщений.

Диаграммы взаимодействия (Collaboration) в BPMN представляют собой схемы, которые описывают взаимодействие между различными участниками бизнес-процесса. Эти диаграммы помогают визуализировать, как различные участники и системы взаимодействуют друг с другом в рамках процесса.

## Элементы Взаимодействия

### 1. Пулы (Pools):

- **Описание:** Представляют участников процесса, такие как отделы или организации.
- **Пример:** Пул "Отдел продаж" и пул "Отдел логистики".

### 2. Дорожки (Lanes):

- **Описание:** Подразделения внутри пула, представляющие роли или функции.
- **Пример:** Дорожка "Менеджер по продажам" внутри пула "Отдел продаж".

### 3. События (Events):

- **Описание:** Начало, конец или промежуточные события в процессе.
- **Пример:** Событие "Начало процесса" или событие "Завершение процесса".

### 4. Действия (Activities):

- **Описание:** Задачи или подпроцессы, выполняемые в рамках процесса.
- **Пример:** Задача "Обработка заказа" или подпроцесс "Проверка платежа".

### 5. Шлюзы (Gateways):

- **Описание:** Управление потоком процесса, включая ветвления, слияния и условия.
- **Пример:** Шлюз "Разветвление на основе условия" или шлюз "Слияние потоков".

### 6. Последовательные потоки (Sequence Flows):

- **Описание:** Определяют последовательность выполнения действий внутри одного пула.
- **Пример:** Поток от задачи "Прием заказа" к задаче "Обработка платежа".

### 7. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными пулами.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

## Семантика Взаимодействия

Семантика взаимодействия в BPMN включает в себя следующие ключевые элементы:

### 1. Пулы (Pools):

- **Описание:** Представляют участников процесса.
- **Пример:** Пул "Отдел продаж" и пул "Отдел логистики".

### 2. Дорожки (Lanes):

- **Описание:** Подразделения внутри пула, представляющие роли или функции.
- **Пример:** Дорожка "Менеджер по продажам" внутри пула "Отдел продаж".

### 3. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными пулами.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

## Правила соединения потоков сообщений

### 1. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными пулами.
- **Правила:**
  - Сообщение должно начинаться и заканчиваться в разных пулах.
  - Сообщение не может начинаться или заканчиваться шлюзом.
  - Сообщение может связывать события, задачи или подпроцессы в разных пулах.

## Пример диаграммы взаимодействия

```
[Pool: Отдел продаж]                                [Pool: Отдел логистики]
[Start Event] --> [Task: Прием заказа] --> [Task: Обработка платежа] -->
[Message Flow] --> [Task: Подготовка к отправке] --> [End Event]
```

В этом примере:

- **Pool: Отдел продаж:** Пул, представляющий отдел продаж.
- **Pool: Отдел логистики:** Пул, представляющий отдел логистики.
- **Start Event:** Начало процесса в отделе продаж.
- **Task: Прием заказа:** Задача по приему заказа в отделе продаж.
- **Task: Обработка платежа:** Задача по обработке платежа в отделе продаж.
- **Message Flow:** Обмен сообщением между отделом продаж и отделом логистики о подтверждении заказа.
- **Task: Подготовка к отправке:** Задача по подготовке к отправке в отделе логистики.
- **End Event:** Завершение процесса в отделе логистики.

# 17. Диаграммы Диалогов (Conversation) в BPMN. Семантика Диалогов. Элементы Диалогов.

Диаграммы диалогов (Conversation) в BPMN представляют собой схемы, которые описывают взаимодействие между различными участниками бизнес-процесса на высоком уровне абстракции. Эти диаграммы фокусируются на обмене сообщениями между участниками и помогают визуализировать, как различные участники и системы взаимодействуют друг с другом в рамках процесса.

## Семантика Диалогов

Семантика диалогов в BPMN включает в себя следующие ключевые элементы:

### 1. Пулы (Pools):

- **Описание:** Представляют участников процесса, такие как отделы или организации.
- **Пример:** Пул "Отдел продаж" и пул "Отдел логистики".

### 2. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными пулами.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

### 3. Диалоги (Conversations):

- **Описание:** Группируют сообщения, которые относятся к одному и тому же взаимодействию или теме.
- **Пример:** Диалог "Обработка заказа" включает все сообщения, связанные с обработкой заказа.

## Элементы Диалогов

### 1. Пулы (Pools):

- **Описание:** Представляют участников процесса.
- **Пример:** Пул "Отдел продаж" и пул "Отдел логистики".

### 2. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными пулами.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

### 3. Диалоги (Conversations):

- **Описание:** Группируют сообщения, которые относятся к одному и тому же взаимодействию или теме.
- **Пример:** Диалог "Обработка заказа" включает все сообщения, связанные с обработкой заказа.

## Пример диаграммы диалогов

```
[Pool: Отдел продаж]                                [Pool: Отдел  
логистики]  
[Conversation: Обработка заказа]  
[Message Flow: Запрос на подтверждение заказа] --> [Message Flow:  
Подтверждение заказа]  
[Message Flow: Уведомление о готовности к отправке] --> [Message Flow:  
Подтверждение отправки]
```

В этом примере:

- **Pool: Отдел продаж:** Пул, представляющий отдел продаж.
- **Pool: Отдел логистики:** Пул, представляющий отдел логистики.
- **Conversation: Обработка заказа:** Диалог, группирующий все сообщения, связанные с обработкой заказа.
- **Message Flow: Запрос на подтверждение заказа:** Сообщение от отдела продаж к отделу логистики с запросом на подтверждение заказа.
- **Message Flow: Подтверждение заказа:** Сообщение от отдела логистики к отделу продаж с подтверждением заказа.
- **Message Flow: Уведомление о готовности к отправке:** Сообщение от отдела продаж к отделу логистики с уведомлением о готовности к отправке.
- **Message Flow: Подтверждение отправки:** Сообщение от отдела логистики к отделу продаж с подтверждением отправки.

Эти элементы и правила помогают создать четкую и структурированную диаграмму диалогов, которая визуализирует, как различные участники и системы взаимодействуют друг с другом в рамках бизнес-процесса на высоком уровне абстракции.

## 18. Диаграммы Хореографии (Choreography) в BPMN. Типы Хореографий. Элементы Хореографии. Семантика Хореографии.

### Диаграммы Хореографии (Choreography) в BPMN

Диаграммы хореографии (Choreography) в BPMN представляют собой схемы, которые описывают взаимодействие между различными участниками бизнес-процесса на уровне сообщений. В отличие от диаграмм оркестровки, которые фокусируются на внутренних процессах одного участника, диаграммы хореографии описывают, как участники взаимодействуют друг с другом через обмен сообщениями.

### Типы Хореографий

## 1. Глобальная хореография:

- **Описание:** Описывает взаимодействие между всеми участниками процесса на высоком уровне.
- **Пример:** Хореография, описывающая взаимодействие между отделом продаж, отделом логистики и клиентом в процессе обработки заказа.

## 2. Локальная хореография:

- **Описание:** Описывает взаимодействие между ограниченным числом участников или в рамках одного участника.
- **Пример:** Хореография, описывающая взаимодействие между отделом продаж и клиентом в процессе подтверждения заказа.

# Элементы Хореографии

## 1. Участники (Participants):

- **Описание:** Представляют участников процесса, такие как отделы или организации.
- **Пример:** Участник "Отдел продаж", участник "Отдел логистики".

## 2. События (Events):

- **Описание:** Начало, конец или промежуточные события в процессе.
- **Пример:** Событие "Начало процесса" или событие "Завершение процесса".

## 3. Действия (Activities):

- **Описание:** Задачи или подпроцессы, выполняемые в рамках процесса.
- **Пример:** Задача "Обработка заказа" или подпроцесс "Проверка платежа".

## 4. Шлюзы (Gateways):

- **Описание:** Управление потоком процесса, включая ветвления, слияния и условия.
- **Пример:** Шлюз "Разветвление на основе условия" или шлюз "Слияние потоков".

## 5. Последовательные потоки (Sequence Flows):

- **Описание:** Определяют последовательность выполнения действий внутри одного участника.
- **Пример:** Поток от задачи "Прием заказа" к задаче "Обработка платежа".

## 6. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными участниками.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

# Семантика Хореографии

Семантика хореографии в BPMN включает в себя следующие ключевые элементы:

### 1. Участники (Participants):

- **Описание:** Представляют участников процесса.
- **Пример:** Участник "Отдел продаж", участник "Отдел логистики".

### 2. События (Events):

- **Описание:** Начало, конец или промежуточные события в процессе.
- **Пример:** Событие "Начало процесса" или событие "Завершение процесса".

### 3. Действия (Activities):

- **Описание:** Задачи или подпроцессы, выполняемые в рамках процесса.
- **Пример:** Задача "Обработка заказа" или подпроцесс "Проверка платежа".

### 4. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между различными участниками.
- **Пример:** Сообщение от "Отдела продаж" к "Отделу логистики" о подтверждении заказа.

## 19. Назначение Подпроцессов (Sub-Process) в BPMN. Классификация. Правила соединения с другими элементами моделей. Маркеры подпроцессов.

### Назначение Подпроцессов (Sub-Process) в BPMN

Подпроцессы (Sub-Process) в BPMN используются для декомпозиции сложных процессов на более мелкие, управляемые части. Это позволяет упростить моделирование и анализ бизнес-процессов, а также способствует повторному использованию и модульности.

### Классификация Подпроцессов

#### 1. Встроенные подпроцессы (Embedded Sub-Processes):

- **Описание:** Подпроцессы, которые полностью включены в основной процесс и выполняются как часть этого процесса.
- **Пример:** Подпроцесс "Проверка платежа" внутри основного процесса "Обработка заказа".

#### 2. Ссылочные подпроцессы (Call Activities):

- **Описание:** Подпроцессы, которые вызываются из основного процесса и выполняются как отдельные процессы.
- **Пример:** Подпроцесс "Обработка возврата товара", который вызывается из основного процесса "Обработка заказа".

#### 3. Событийные подпроцессы (Event Sub-Processes):

- **Описание:** Подпроцессы, которые активируются на основе событий и выполняются параллельно с основным процессом.
- **Пример:** Подпроцесс "Обработка исключений", который активируется при возникновении ошибки в основном процессе.

## Правила соединения с другими элементами моделей

### 1. Последовательные потоки (Sequence Flows):

- **Описание:** Определяют последовательность выполнения действий.
- **Правила:**
  - Поток должен начинаться с события или действия и заканчиваться событием или действием.
  - Поток не может начинаться или заканчиваться шлюзом.
  - Подпроцесс может быть связан с другими элементами через последовательные потоки.

### 2. Сообщения (Message Flows):

- **Описание:** Определяют обмен сообщениями между участниками процесса.
- **Правила:**
  - Сообщение должно начинаться и заканчиваться в разных пулах.
  - Сообщение не может начинаться или заканчиваться шлюзом.
  - Подпроцесс может быть связан с другими элементами через сообщения.

### 3. Ассоциации (Associations):

- **Описание:** Связывают артефакты с потоками или действиями.
- **Правила:**
  - Ассоциация может связывать артефакт с любым элементом потока (событие, действие, шлюз).
  - Ассоциация не может связывать два артефакта.
  - Подпроцесс может быть связан с артефактами через ассоциации.

## Маркеры подпроцессов

### 1. Маркеры встроенных подпроцессов:

- **Описание:** Используются для обозначения встроенных подпроцессов.
- **Пример:** Прямоугольник с закругленными углами, содержащий иконку подпроцесса (маленький прямоугольник с плюсом внутри).

### 2. Маркеры ссылочных подпроцессов:

- **Описание:** Используются для обозначения ссылочных подпроцессов.
- **Пример:** Прямоугольник с закругленными углами, содержащий иконку вызова (маленький прямоугольник с плюсом внутри и стрелкой, указывающей на вызываемый процесс).

### 3. Маркеры событийных подпроцессов:

- **Описание:** Используются для обозначения событийных подпроцессов.
- **Пример:** Прямоугольник с закругленными углами, содержащий иконку события (маленький круг с плюсом внутри).

## Пример использования подпроцессов

```
[Start Event] --> [Task: Прием заказа] --> [Sub-Process: Проверка платежа]
--> [Task: Подготовка к отправке] --> [End Event]
```

В этом примере:

- **Start Event:** Начало процесса.
- **Task: Прием заказа:** Задача по приему заказа.
- **Sub-Process: Проверка платежа:** Подпроцесс, включающий задачи по проверке платежа.
- **Task: Подготовка к отправке:** Задача по подготовке к отправке.
- **End Event:** Завершение процесса.

## 20. Паттерны (шаблоны) различных аспектов BPMN моделей, организующие: потоки управления, данные, ресурсы, обработку исключений, события процессов.

### Паттерны (шаблоны) различных аспектов BPMN моделей

Паттерны в BPMN помогают организовать и структурировать различные аспекты бизнес-процессов, такие как потоки управления, данные, ресурсы, обработка исключений и события процессов. Эти паттерны обеспечивают повторное использование и упрощают моделирование сложных процессов.

### Потоки управления

#### 1. Последовательный поток (Sequence Flow):

- **Описание:** Определяет последовательность выполнения действий. Этот паттерн используется для моделирования линейных процессов, где каждое действие следует за предыдущим.
- **Применение:** Используется в процессах, где важна строгая последовательность выполнения задач.

#### 2. Параллельное выполнение (Parallel Execution):

- **Описание:** Определяет параллельное выполнение нескольких задач. Этот паттерн используется для моделирования процессов, где несколько задач могут выполняться одновременно.



- **Применение:** Используется в процессах, где необходимо выполнить несколько задач параллельно для ускорения выполнения процесса.

### 3. Ветвление и слияние (Branching and Merging):

- **Описание:** Управление ветвлениями и слияниями потоков на основе условий. Этот паттерн используется для моделирования процессов, где необходимо принять решение на основе условий и разветвить поток выполнения.
- **Применение:** Используется в процессах, где необходимо принять решение на основе условий и разветвить поток выполнения.

## Данные

### 1. Данные объекты (Data Objects):

- **Описание:** Представляют данные, используемые или создаваемые в процессе. Этот паттерн используется для моделирования данных, которые передаются между задачами.
- **Применение:** Используется для моделирования данных, которые передаются между задачами и используются в процессе.

### 2. Данные хранилища (Data Stores):

- **Описание:** Представляют хранилища данных, доступные для всего процесса. Этот паттерн используется для моделирования централизованных хранилищ данных, которые используются в процессе.
- **Применение:** Используется для моделирования централизованных хранилищ данных, которые используются в процессе.

### 3. Данные входа и выхода (Data Inputs and Outputs):

- **Описание:** Определяют входные и выходные данные для задач. Этот паттерн используется для моделирования данных, которые вводятся в задачу и выводятся из задачи.
- **Применение:** Используется для моделирования данных, которые вводятся в задачу и выводятся из задачи.

## Ресурсы

### 1. Пулы и дорожки (Pools and Lanes):

- **Описание:** Представляют участников процесса и их роли. Этот паттерн используется для моделирования участников процесса и их ролей.
- **Применение:** Используется для моделирования участников процесса и их ролей.

### 2. Ресурсы (Resources):

- **Описание:** Представляют ресурсы, необходимые для выполнения задач. Этот паттерн используется для моделирования ресурсов, которые используются в процессе.

- **Применение:** Используется для моделирования ресурсов, которые используются в процессе.

## Обработка исключений

### 1. Событийные подпроцессы (Event Sub-Processes):

- **Описание:** Подпроцессы, которые активируются на основе событий и выполняются параллельно с основным процессом. Этот паттерн используется для моделирования исключительных ситуаций, которые требуют обработки.
- **Применение:** Используется для моделирования исключительных ситуаций, которые требуют обработки.

### 2. Исключения (Exceptions):

- **Описание:** Определяют исключительные ситуации и их обработку. Этот паттерн используется для моделирования исключительных ситуаций и их обработки.
- **Применение:** Используется для моделирования исключительных ситуаций и их обработки.

## События процессов

### 1. Начальные события (Start Events):

- **Описание:** Иницируют начало процесса. Этот паттерн используется для моделирования начала процесса.
- **Применение:** Используется для моделирования начала процесса.

### 2. Промежуточные события (Intermediate Events):

- **Описание:** Происходят в середине процесса. Этот паттерн используется для моделирования событий, которые происходят в середине процесса.
- **Применение:** Используется для моделирования событий, которые происходят в середине процесса.

### 3. Конечные события (End Events):

- **Описание:** Завершают процесс. Этот паттерн используется для моделирования завершения процесса.
- **Применение:** Используется для моделирования завершения процесса.