Основные практики архитектора ПО

Преподаватель:

Валентин Степанов

Модуль 7. Архитектура в жизненном цикле проекта разработки ПО

- Применение изученных архитектурных практик в жизненном цикле проектов разработки ПО, их сочетание с разными проектными методологиями, в т. ч. гибкими (Agile) методологиями разработки
- Разновидности роли архитектора
- Взаимодействие с ролями аналитика и менеджера проекта

Жизненный цикл ПО

- Сбор и анализ требований
- **Проектирование**
- Разработка
- Тестирование
- Сопровождение

Входные данные для проектирования

- Концепция проекта
- Бизнес-цели
- Ограничения проекта
- Требования к ПО
- Модель разработки
- Команда проекта

Этапы процесса проектирования

- Определение целей архитектуры
  - Начальное определение задач архитектуры
  - Определение потребителей архитектуры
  - Определение ограничений
- Основные сценарии
- Общее представление приложения
- Потенциальные проблемы
- Варианты решений

Упрощения в Agile

- Отсутствие документации
- Совмещение ролей
- Тесное взаимодействие
- Слияние с разработкой

Аналитик vs Архитектор

- Размытые границы
- Совмещение документации
- Уровень детализации

Пользовательские требования

- **Варианты использования** и сценарии (Use Cases and Scenarios) описывают, как человек или система взаимодействуют с моделируемым решением для достижения цели.

- **История пользователя** (User Stories) представляет собой небольшое, краткое описание функциональности или качества, необходимые для обеспечения ценности для конкретной заинтересованной стороны.

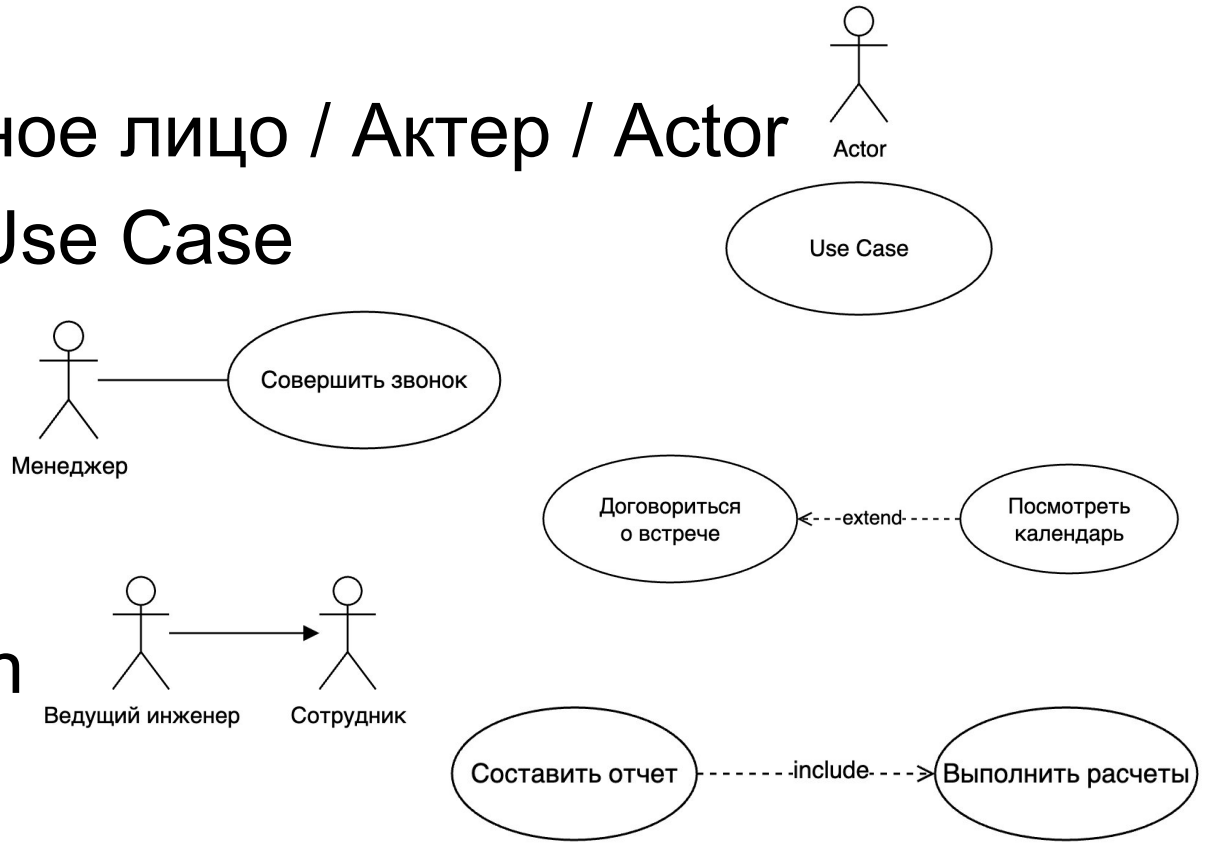
Диаграммы вариантов использования (use-case diagrams)

Структурные элементы:

- Участник / Заинтересованное лицо / Актер / Actor
- Вариант использования / Use Case

Взаимосвязи:

- Ассоциация / Association
- Расширение / Extend
- Обобщение / Generalization
- Включение / Include



Описание варианта использования

- **Идентификатор и название**
- Автор
- Дата создания
- Основное действующее лицо
- Дополнительное действующее лицо
- Описание
- **Триггер**
- **Предварительные условия**
- **Выходные условия**
- **Нормальное направление**

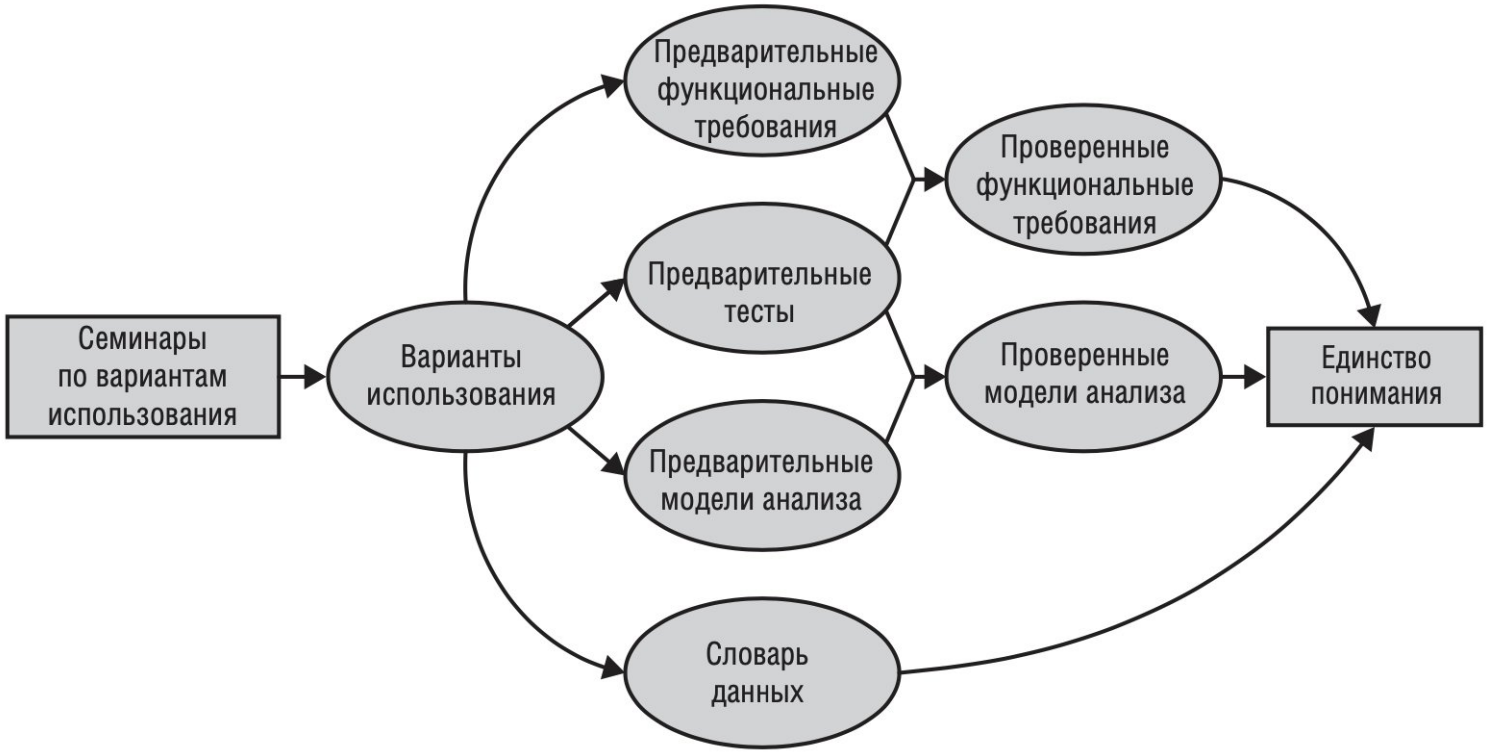
развития варианта использования

- Альтернативное направление развития варианта использования
- Исключения
- Приоритет
- Частота использования
- Бизнес-правила
- Специальные требования
- Предположения

Определение вариантов использования

- Определить действующих лиц, бизнес-процессы, поддерживаемые системой, и варианты использования для действий, в которых участвуют действующие лица и система
- Выразить бизнес-процессы в терминах определенных сценариев, обобщить сценарии в варианты использования и определить действующих лиц для каждого варианта
- Используя описание бизнес-процесса, задать вопрос: «Какие задачи должна система выполнить, чтобы реализовать этот процесс или преобразовать входные данные в выходные?» Эти задачи могут быть вариантами использования
- Определить внешние события, на которые система должна реагировать, затем соотнести эти события с участвующими лицами и конкретными вариантами использования
- Применить CRUD-анализ (Create, Read, Update, Delete – создание, чтение, обновление, удаление) для определения сущностей данных, которым нужны варианты использования для создания, чтения, обновления, удаления или других операций с данными
- Проанализировать контекстную диаграмму, ответив на вопрос: «Какие цели нужно каждой из внешних сущностей достичь с использованием системы?»

Анализ вариантов использования



Структура User Story

- Заголовок
- Роль
- Функционал
- Выгода

INVEST критерии написания User Story

- I — Independent. Независимая
- N — Negotiable. Договорная
- V — Valuable. Ценная
- E — Estimable. Доступная для оценки
- S — Small. Маленькая
- T — Testable. Тестируемая