

Лекция 1. Введение в проектирование

Что такое проектирование?

Жизненный цикл ПО

RUP и UML

Немного о курсовом проекте

Литература

- **Крэг Ларман, «Применение UML 2.0 и шаблонов проектирования»**
- Гради Буч, Роберт А Максимчук, «Объектно-ориентированный анализ и проектирование с примерами приложений» (3-е издание)
- Иан Соммервилл, «Инженерия программного обеспечения»

Проектирование

Проектирование (от лат. *projectus* – брошенный вперед) – это процесс создания проекта – прототипа, прообраза предлагаемого или возможного объекта, состояния.

Проектирование программного обеспечения

Проектирование программного обеспечения — процесс создания проекта программного обеспечения (ПО), а также дисциплина, изучающая методы проектирования.

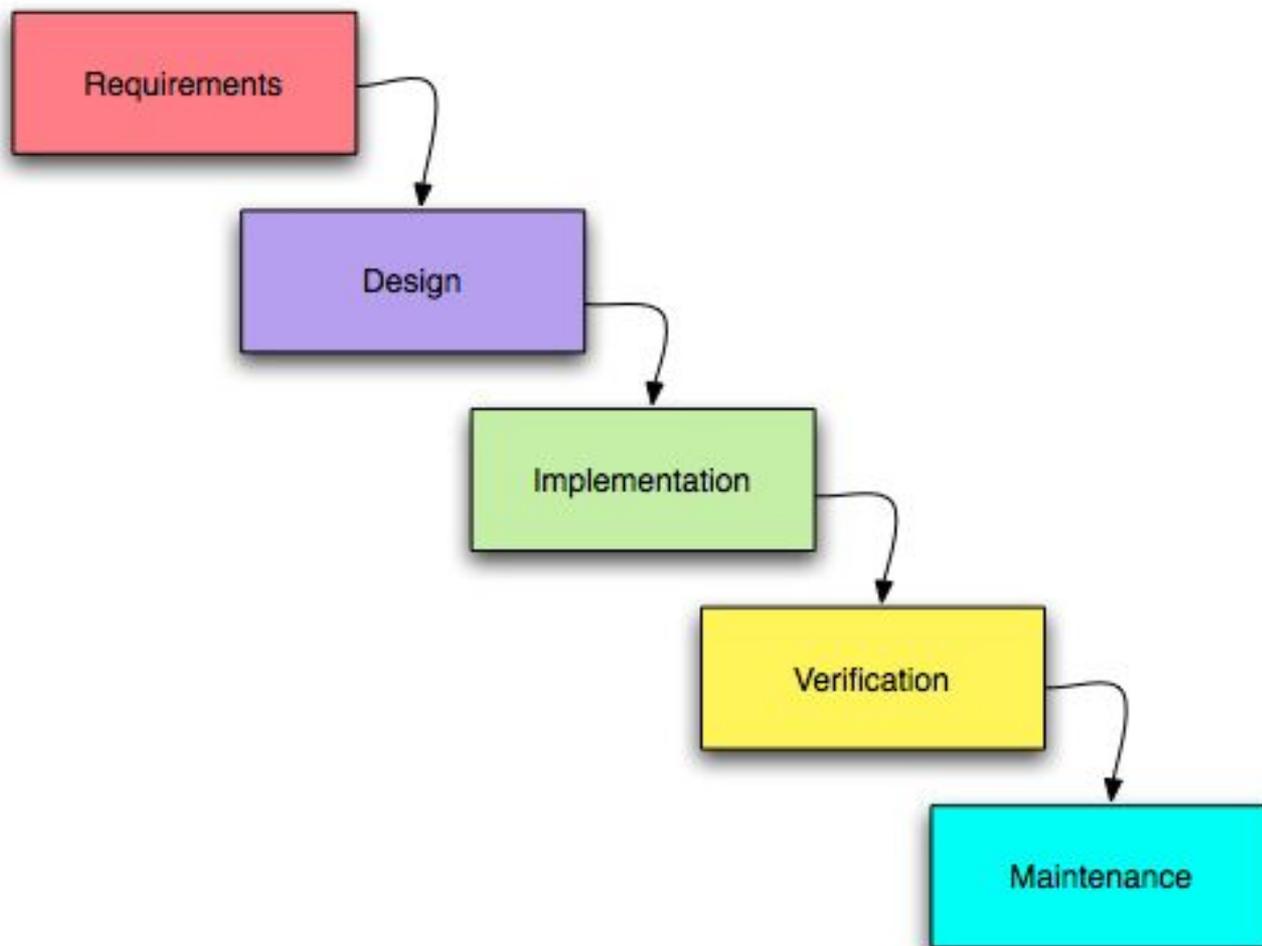
Объекты, подлежащие проектированию:

- Архитектура ПО;
- Устройство компонентов ПО;
- Пользовательские интерфейсы.

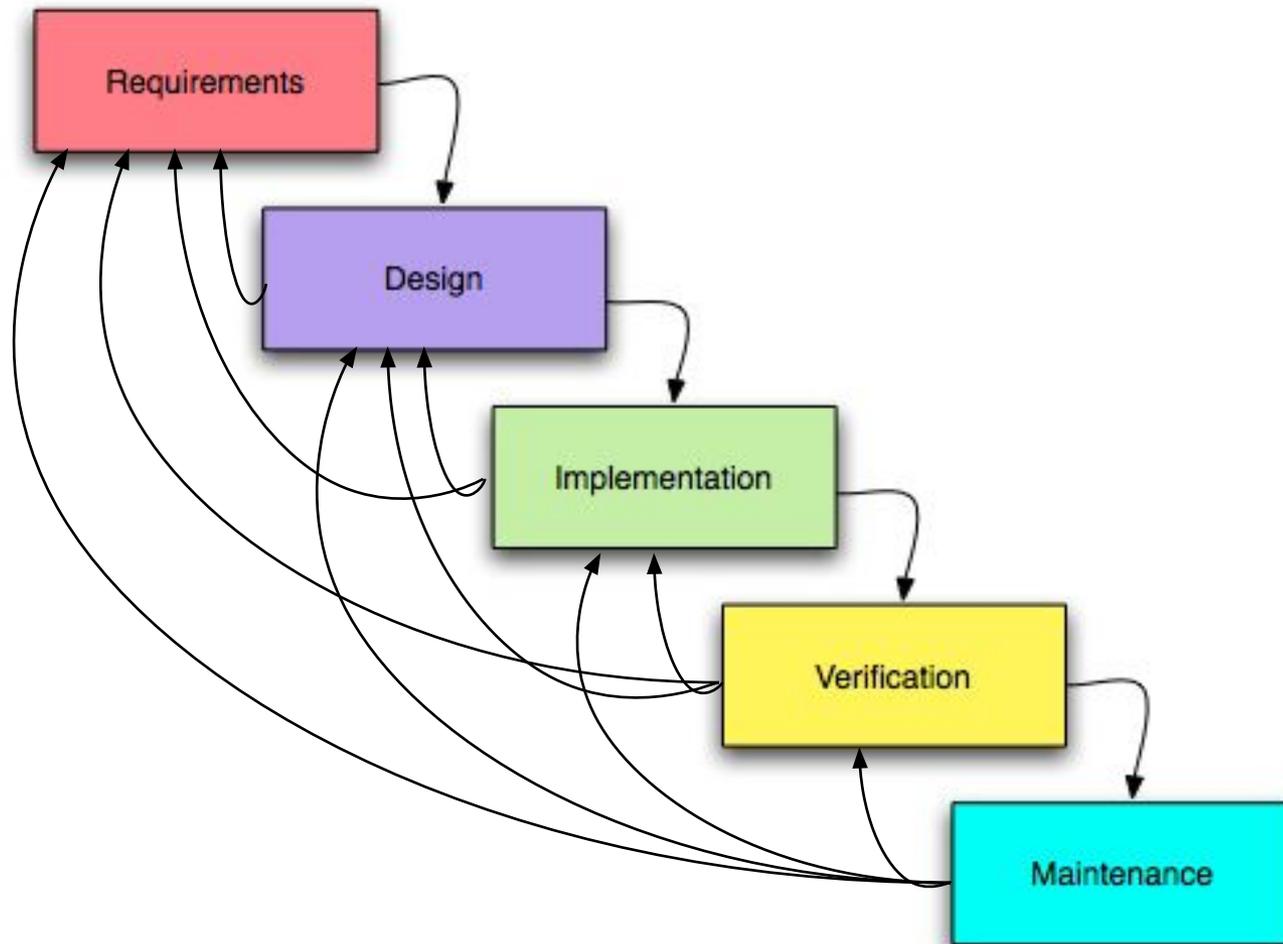
Место проектирования в жизненном цикле ПО

Жизненный цикл ПО — это непрерывный процесс, который начинается с момента принятия решения о необходимости его создания и заканчивается в момент его полного изъятия из эксплуатации.

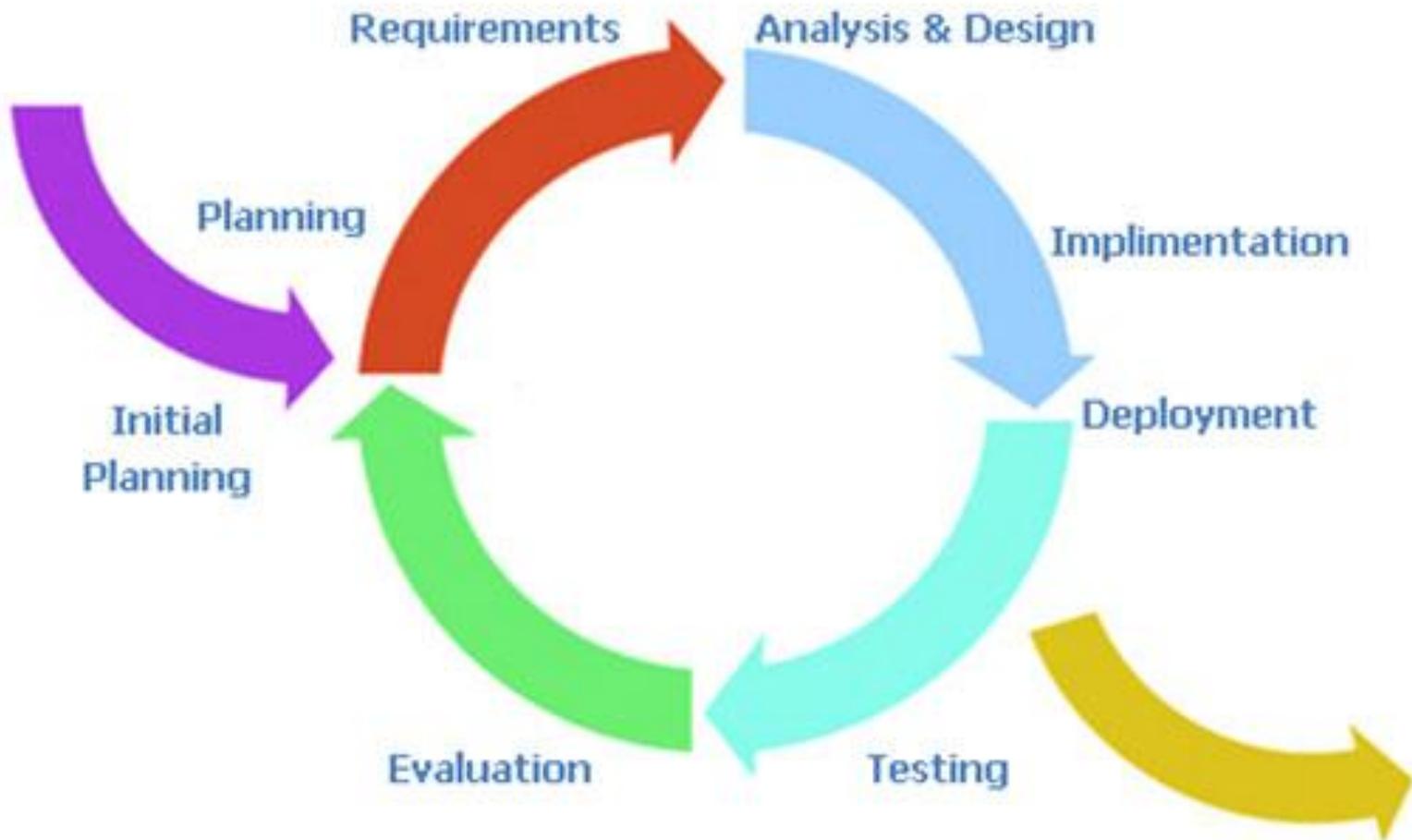
Каскадный жизненный цикл



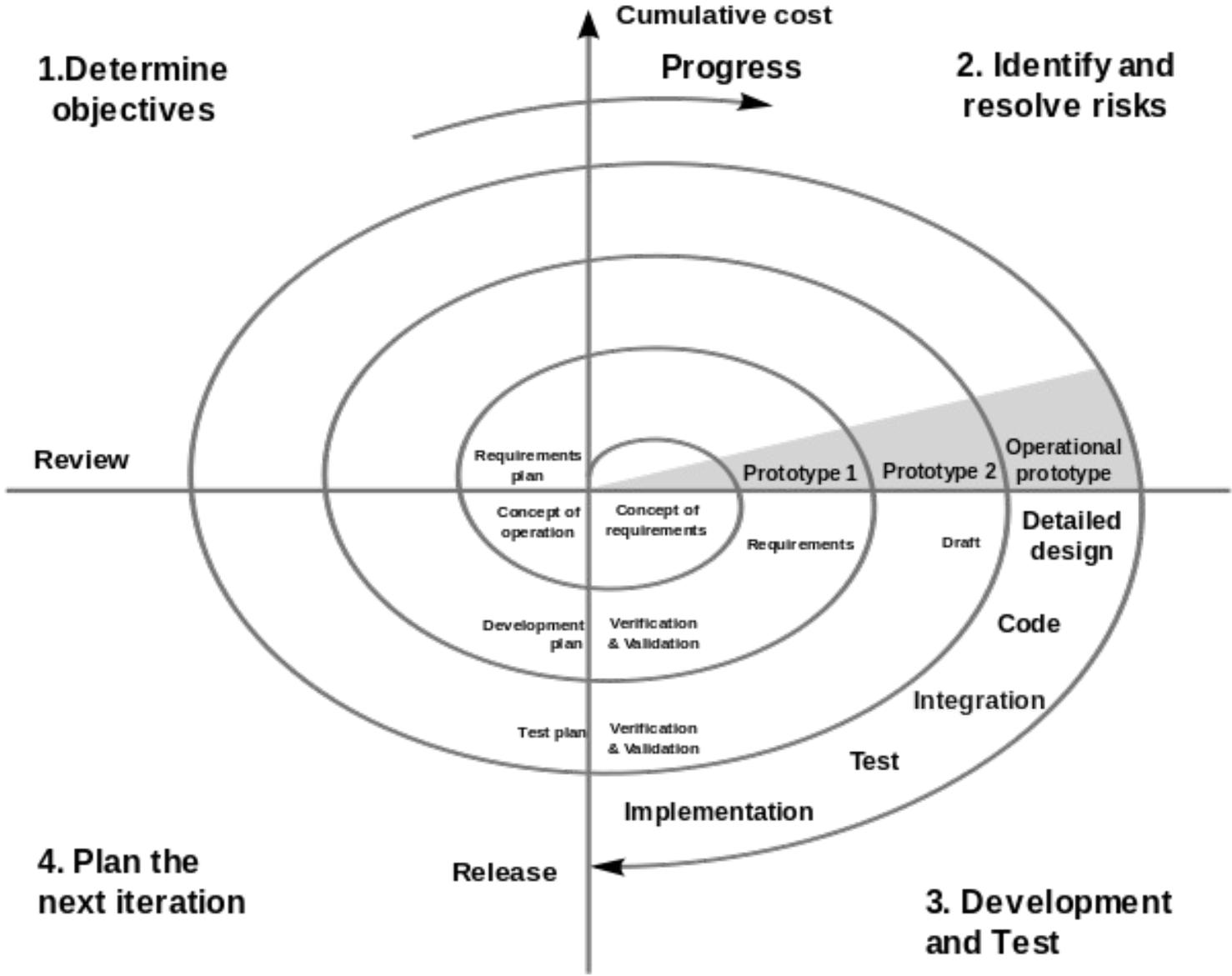
Каскадно-возвратный цикл



Итерационный жизненный цикл



Спиральный жизненный цикл



Экстремальное программирование

- **Короткие обратные связи**
 - Разработка через тестирование
 - Парное программирование
 - Заказчик всегда рядом
- **Непрерывный процесс**
 - Частые релизы
 - Непрерывная интеграция
 - Постоянный рефакторинг
- **Общее понимание**
 - Простота
 - Метафора системы (единый язык)
 - Коллективное владение кодом
 - Стандарт оформления кода
- **Социальная защищенность программиста**
 - Строго 40-часовая рабочая неделя

Rational Unified Process

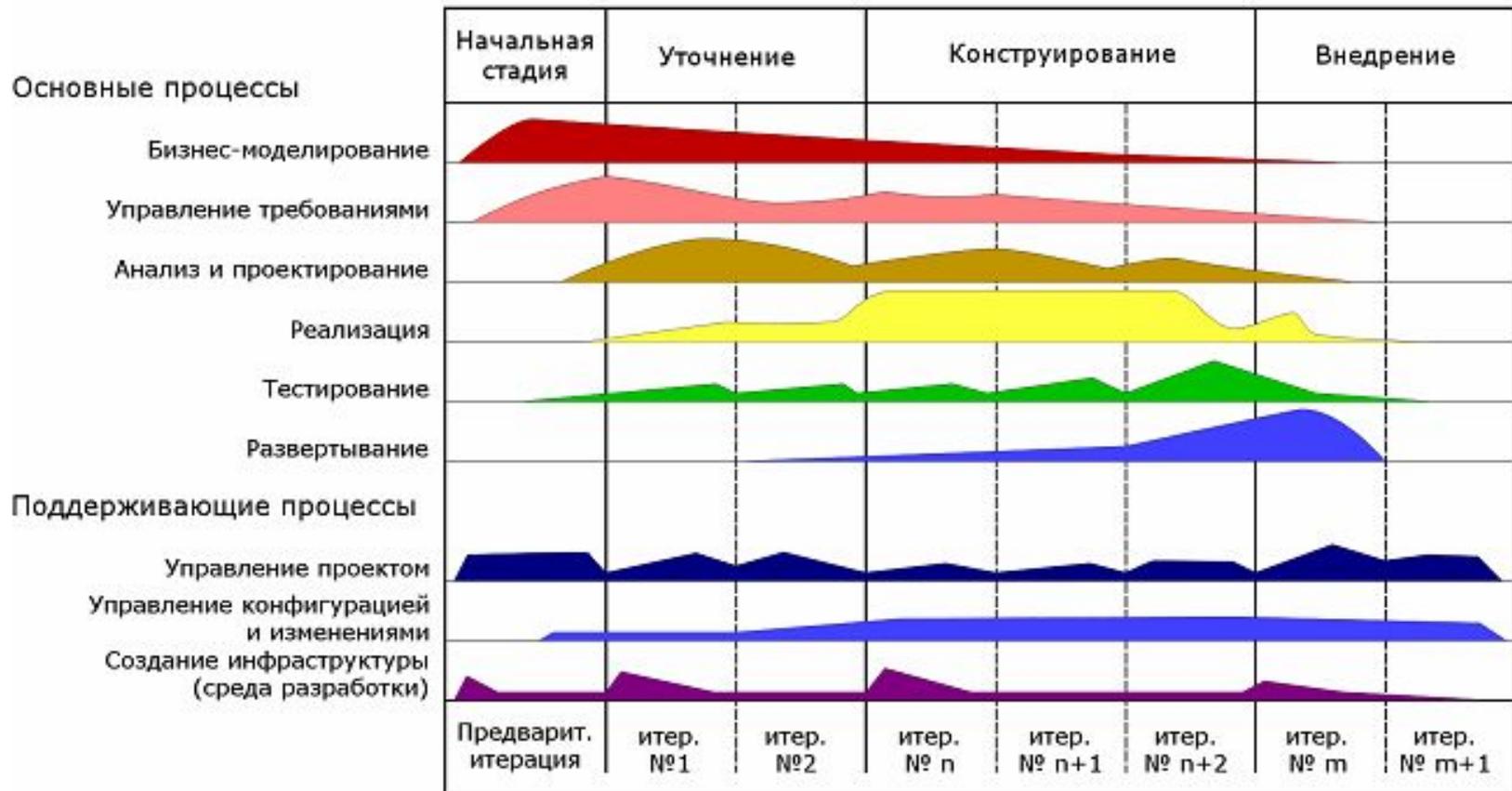
Rational Unified Process (RUP) — методология разработки программного обеспечения, созданная компанией Rational Software.

- **Итеративная разработка.**
- **UML (Unified Modeling Language).**
- **Ожидание изменений в требованиях, проектных решениях и реализации в процессе разработки.**
- **Компонентная архитектура.**
- **Постоянное обеспечение качества.**

Диаграмма RUP

Рабочие процессы

Стадии



Итерации

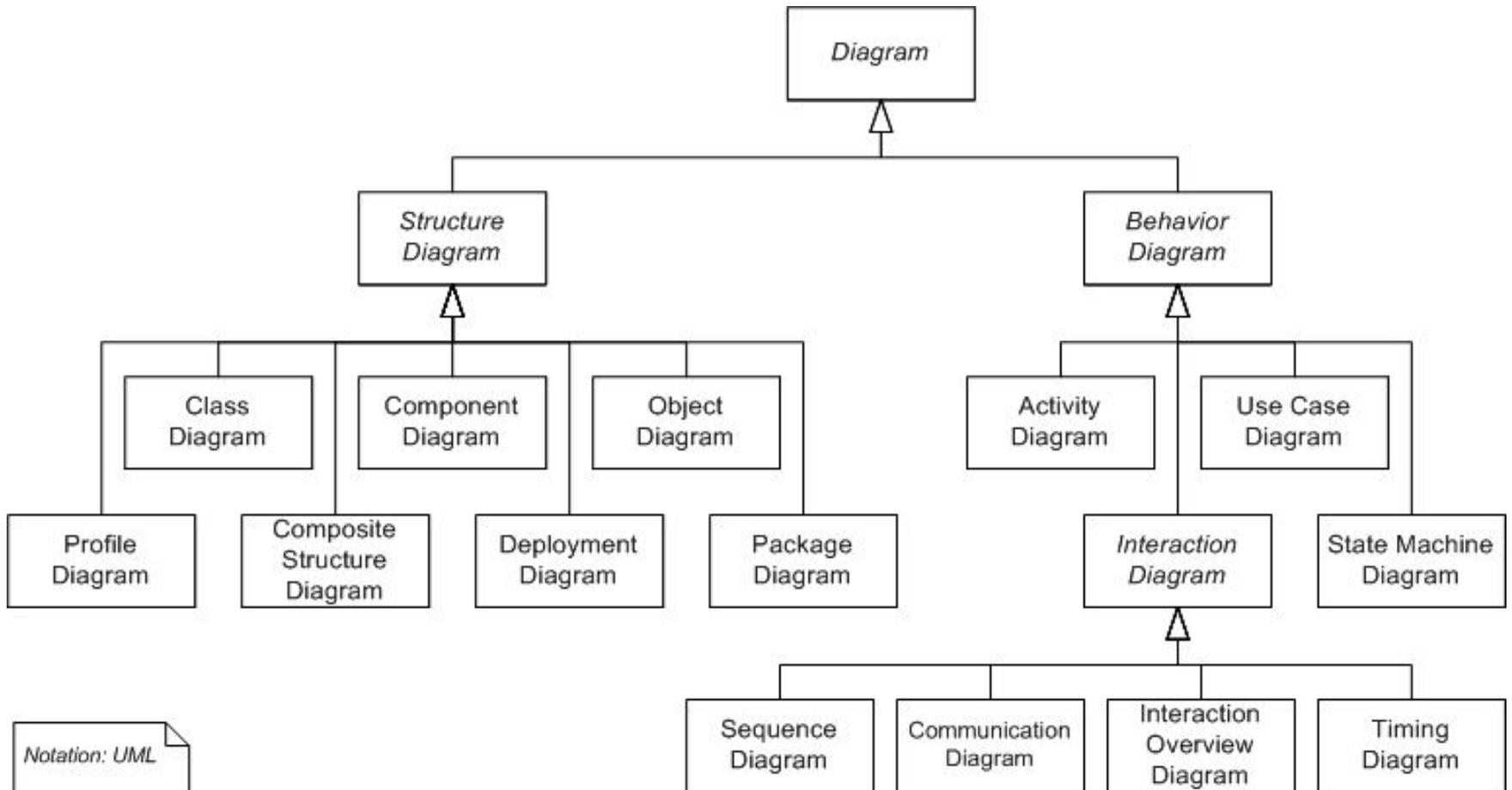
UML

UML (англ. Unified Modeling Language — унифицированный язык моделирования) — язык графического описания для объектного моделирования в области разработки программного обеспечения.

Преимущества UML

- объектно-ориентирован
- позволяет описать систему практически со всех возможных точек зрения
- диаграммы UML сравнительно просты для чтения
- расширяет и позволяет вводить собственные текстовые и графические стереотипы
- получил широкое распространение и динамично развивается

Классификация диаграмм UML 2.3

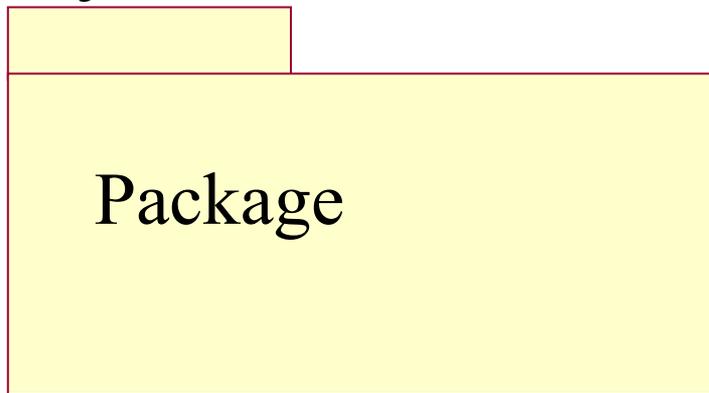


Диаграммы UML

- Требования
 - **Диаграмма прецедентов (Use Case Diagram)**
- Анализ
 - **Концептуальная диаграмма классов (Class Diagram, Domain Class Model)**
 - Диаграмма последовательностей (Sequence Diagram)
 - Диаграмма действий (Activity Diagram)
- Проектирование
 - **Диаграмма классов проекта (Class Diagram, Design Model)**
 - Диаграмма состояний (Statechart Diagram)
 - Диаграмма компонентов (Component Diagram)
 - Диаграмма развёртывания (Deployment Diagram)

Пакеты в UML

Пакеты - средство организации объектов и диаграмм. Являются логическими контейнерами сродни папкам с документами.



Пакеты в Rational XDE

The screenshot displays the Rational XDE interface for a project named "ISGB - XDE Modeling Project [design] - ISGBDesign.mdx [ISGBDesign::Main]". The main workspace shows a UML Package Diagram with four packages: «layer» Business, «layer» Integration, «layer» Common Elements, and «layer» Presentation. Dashed dependency arrows indicate relationships: «layer» Business depends on «layer» Integration and «layer» Common Elements; «layer» Presentation depends on «layer» Common Elements and «layer» Business. The Model Explorer on the right shows the project structure, including folders for "Исполнители", "Прецеденты", "График Бурения", and "Супервайзер", along with various use cases and logical views.

ISGB - XDE Modeling Project [design] - ISGBDesign.mdx [ISGBDesign::Main]

File Edit View Project Build Debug Modeling Diagram Tools PurifyPlus Window Help

Model Explorer

Element Type

- (ISGB) Usecase
 - Исполнители
 - Прецеденты
 - График Бурения
 - График Бурения
 - Внести изменения в справочники
 - Получить баланс календарного времени
 - Получить ведомость израсходованных м
 - Получить ведомость расхода хим. матер
 - Получить данные инклинометрии
 - Получить отчёт по отработке долот
 - Получить план-график строительства ск
 - Получить статистические отчётные фор
 - Получить суточный рапорт по бурению
 - Супервайзер
 - (:)(:Начальник службы заказчика)
 - (:)(:Супервайзер)
 - (:Внести изменения в справочники)(:Администр
- (ISGB) ISGBDesign
 - «layer» Business
 - «layer» Common Elements
 - «layer» Integration
 - «layer» Presentation
 - Use-Case Realizations
 - Logical View: Architecturally Significant Design Eleme
 - Logical View: Architecturally Significant Use-Case Re
 - Logical View: Key Abstractions
 - Logical View: Overall Design Model Organization
 - Main
 - Process View
 - Key Abstraction 1

Требования. Диаграммы прецедентов

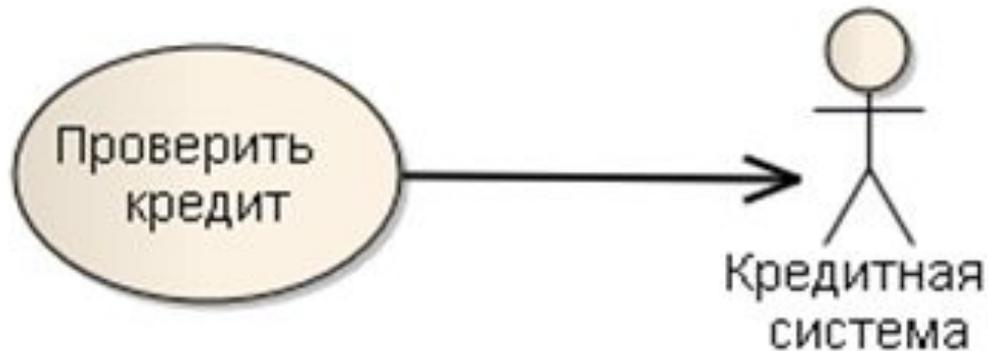
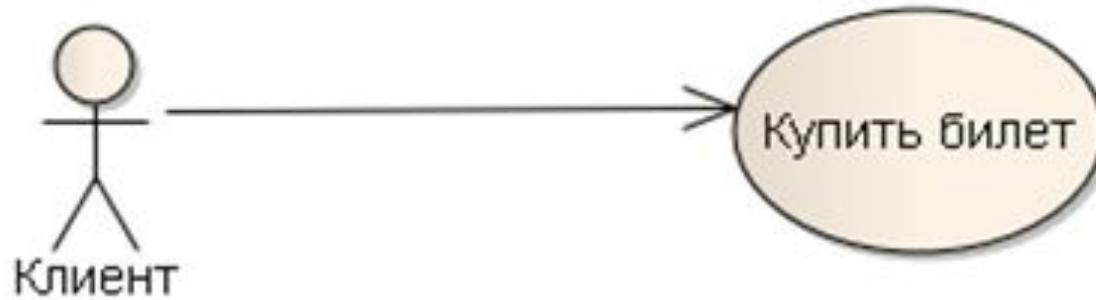
Прецедент – набор взаимосвязанных успешных и неуспешных сценариев, описывающий использование системы исполнителем для решения одной задачи



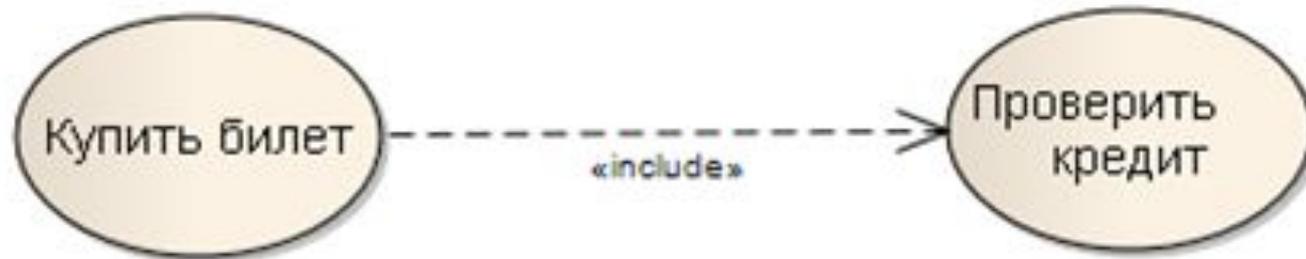
Примеры прецедентов

- Продать билет
- Изменить заказ
- Отменить заказ
- Проверить баланс
- Согласовать договор
- ...

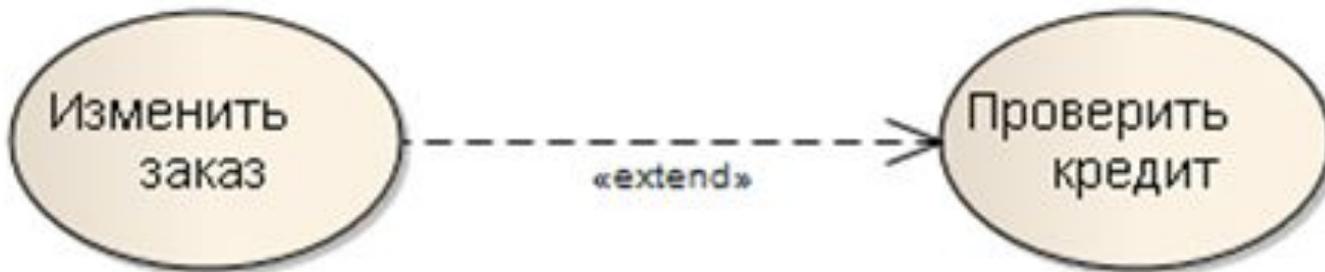
Use Case. Ассоциативные отношения



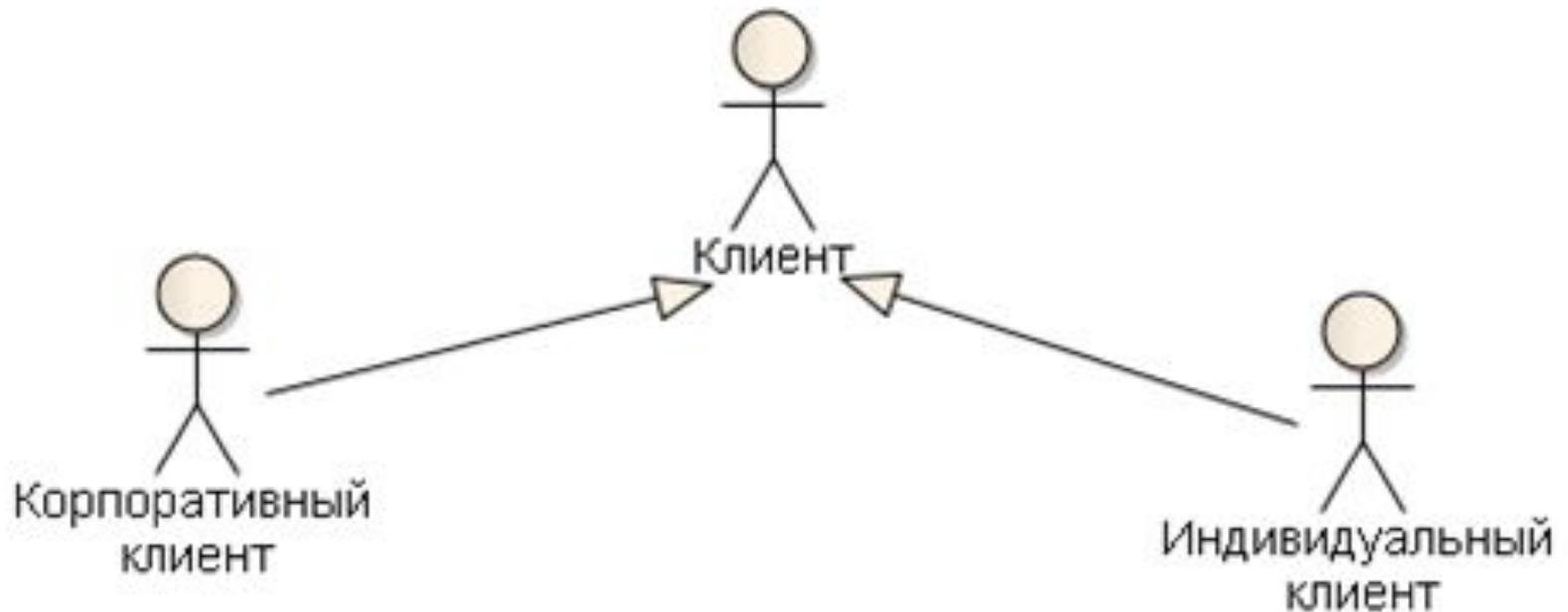
Use Case. Включающие отношения



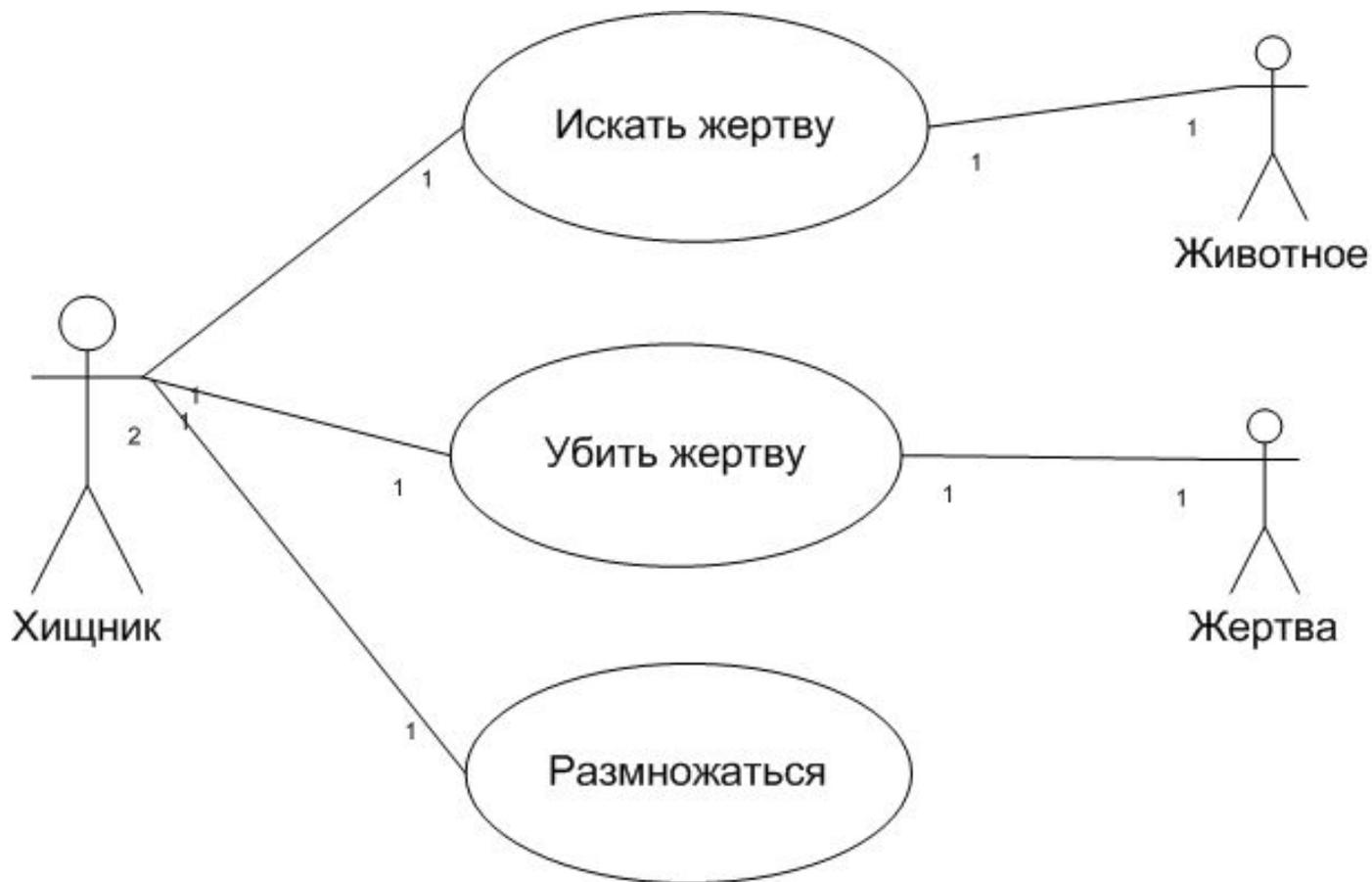
Use Case. Расширяющие отношения



Обобщающие отношения



Множественность связей (на примере Terrarium)



Спецификация прецедента

- Краткое описание
- Действующие лица
- Предусловия (Preconditions)
- Основной сценарий
- Альтернативные сценарии
- Постусловия (Postconditions)

Прецедент «Купить билет»

Описание: Позволяет клиенту получить информацию о рейсах, узнать о наличии свободных мест и произвести покупку билета с оплатой по кредитной карточке.

Прецедент «Купить билет»

Предусловия:

- Пользователь должен быть авторизован
- Соединение с сервером БД должно быть установлено

Прецедент «Купить билет»

Основной сценарий:

1. Прецедент начинается с выбора клиентом режима показа информации о рейсах.
2. Система показывает сведения об аэропортах отправления и назначения, а также датах вылета и возвращения (для билета в обратную сторону).
3. Клиент вводит название городов отправления и назначения, дату вылета и возвращения.
4. Система показывает список доступных рейсов, включая стоимость билетов.

A1: Нет нужного рейса.

5. Пользователь выбирает рейс, билет на который он хотел бы зарезервировать.

Прецедент «Купить билет»

6. Система показывает нее доступные варианты стоимости билетов на этот рейс.

7. Пользователь выбирает категорию билета для резервирования.

A2: Пользователь выбрал бесплатный билет, предоставляемый членам клуба постоянных клиентов.

8. Система показывает цену, которую должен заплатить пользователь.

9. Пользователь подтверждает цену.

10. Система запрашивает тип кредитной карточки, ее номер, имя владельца и дату завершения срока действия.

11. Пользователь вводит тип кредитной карточки, ее номер, имя владельца и дату завершения срока действия.

Прецедент «Купить билет»

12. Система подтверждает продажу по кредитной карточке.

A6: Счет пользователя не найден

A7: Недостаточно денег на счете

E1: Кредитная система недоступна

13. Система резервирует место для пользователя на выбранный рейс

14. Система генерирует и показывает пользователю код подтверждения

15. Пользователь подтверждает получение кода.

16. Прецедент завершается.

Прецедент «Купить билет»

Альтернативные сценарии:

A1: Нет нужного рейса

1. Система выводит сообщение, что для введенных города отправления и значения, а также для указанных дат нет мест на рейсах авиакомпании
2. Пользователь подтверждает просмотр сообщения.
3. Поток возвращается па этап 2 основного потока.

Прецедент «Купить билет»

A2: Пользователь выбрал бесплатный билет, предоставляемый членам клуба постоянных клиентов

1. Система запрашивает идентификационный номер в клубе постоянных клиентов.
2. Пользователь вводит этот номер.
3. Система подтверждает правильность введенного номера.

A3: Неправильный идентификационный номер

4. Система подтверждает, что расстояние, которое налетал пользователь на самолетах авиакомпании, позволяет предоставить бесплатный

A4: Недостаточное расстояние для предоставления бесплатного билета

A5: Нет бесплатных билетов

5. Цена билета устанавливается в \$0.
6. Поток возвращается на этап 8 основного потока.

Прецедент «Купить билет»

Потоки ошибок

E1: Кредитная система недоступна

1. Система выводит сообщение о недоступности кредитной системы.
2. Поток возвращается на этап 10 основного потока.

Прецедент «Купить билет»

Постусловия:

- Денежная транзакция завершена
- Билет забронирован