Унифицированный процесс разработки ПО

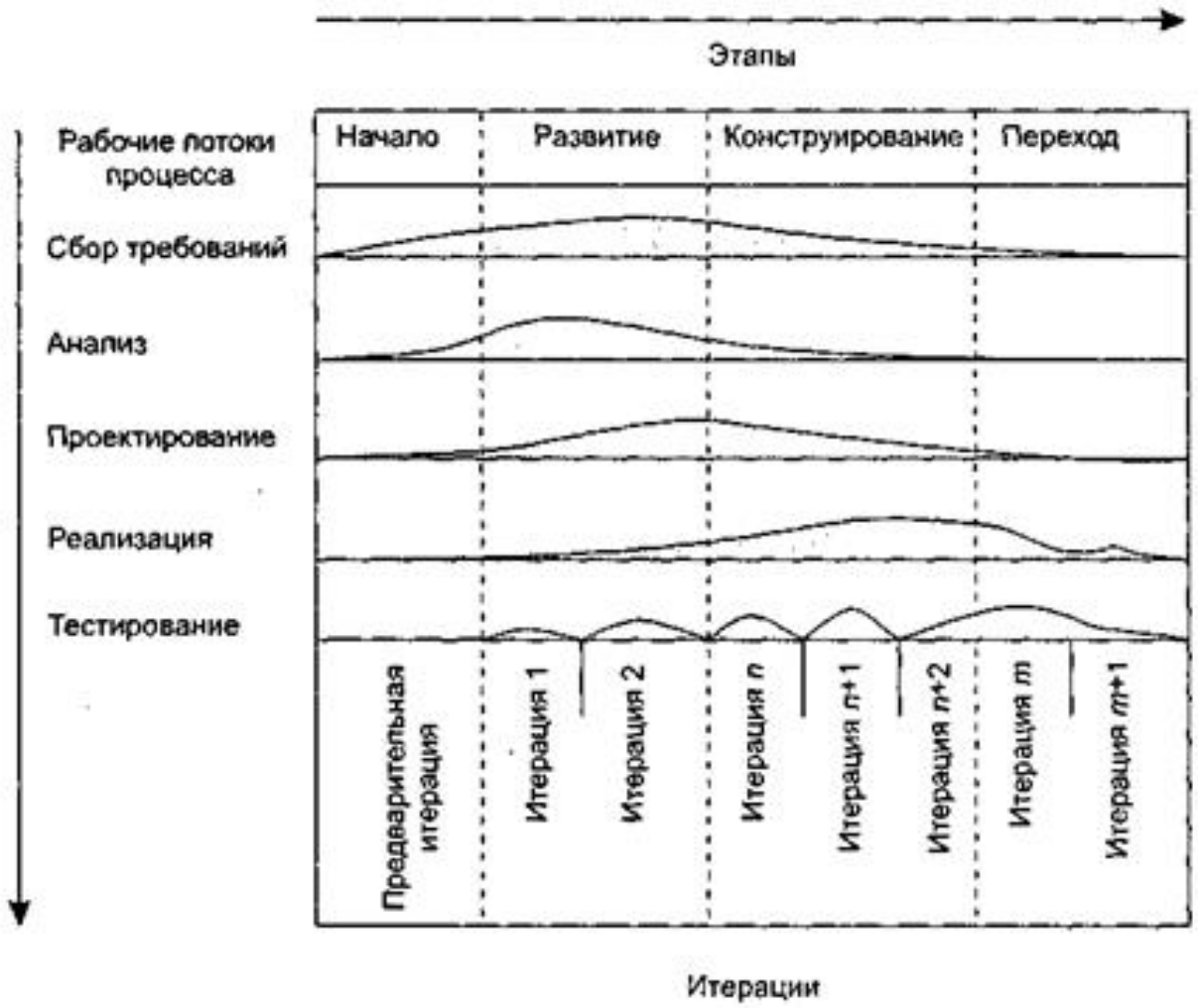
Технологии разработки программного обеспечения

Виноградова М.В.
МГТУ им. Н.Э. Баумана
Кафедра СОИУ (ИУ5)

RUP - Rational Unified Process

- Использует UML (визуальное проектирование);
- Поддерживается Rational / IBM;
- Управляется прецедентами
- Ориентирован на архитектуру
- Итеративность и инкрементность

График RUP


Этапы
Рабочие потоки процесса
Начало
Развитие
Конструирование
Переход
Сбор требований
Анализ
Проектирование
Реализация
Тестирование
Предварительная итерация
Итерация 1
Итерация 2
Итерация n
Итерация n+1
Итерация n+2
Итерация m
Итерация m+1
Итерации

Этапы RUP

- **Начало** - спецификация представления продукта;
- **Развитие** - планирование действий и требуемых решений;
- **Конструирование** - построение ПО в виде серии инкрементных итераций;
- **Переход** - внедрение ПО в среду пользователя (промышленное производство, доставка и применение)

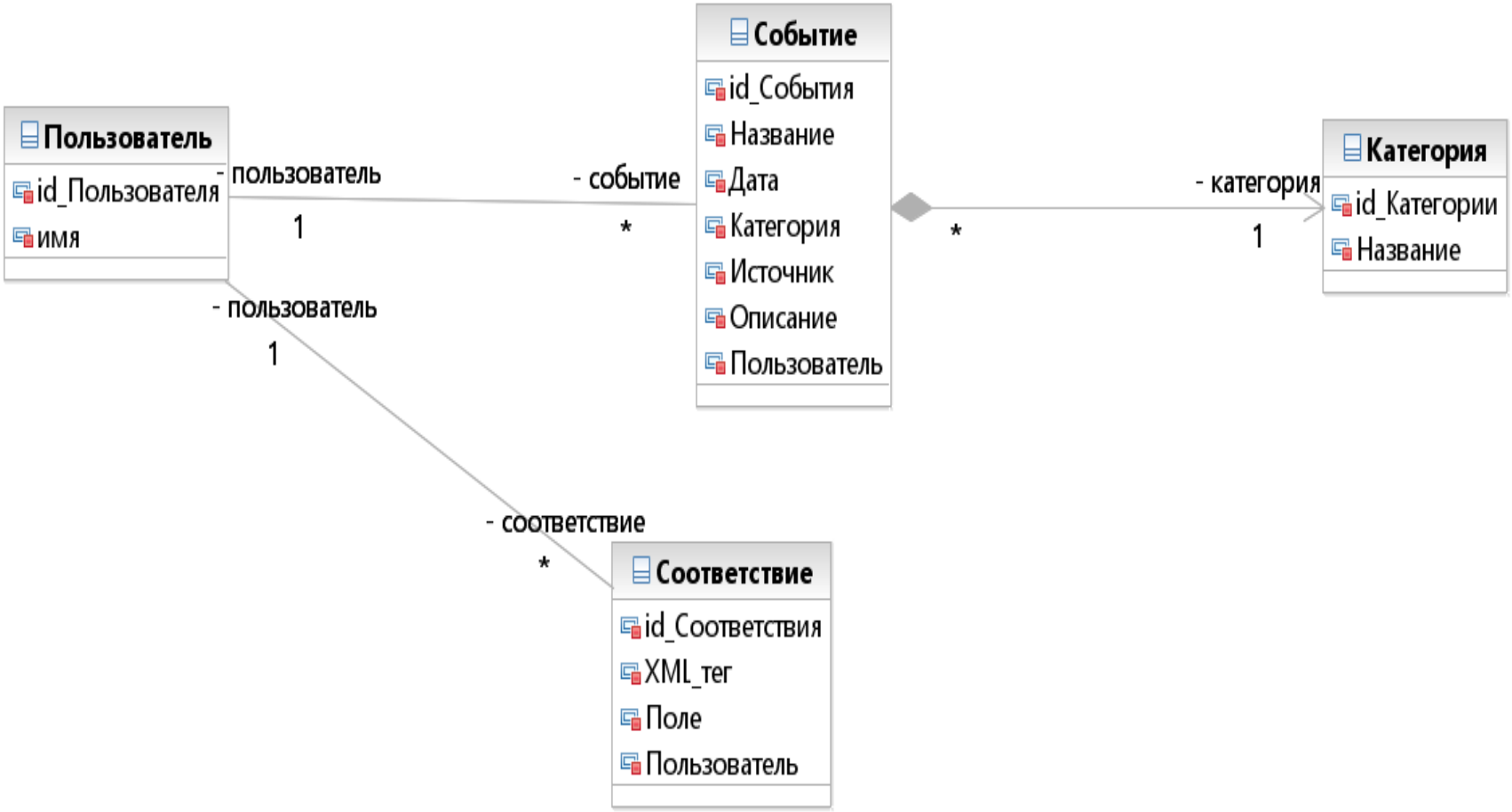
Рабочие процессы RUP

- Сбор требований - что делает система;
- Анализ - преобразование требований в классы и объекты предметной области;
- Проектирование - создание статического и динамического представления системы для выполнения требований;
- Реализация - производство программного кода;
- Тестирование - проверка системы в целом.

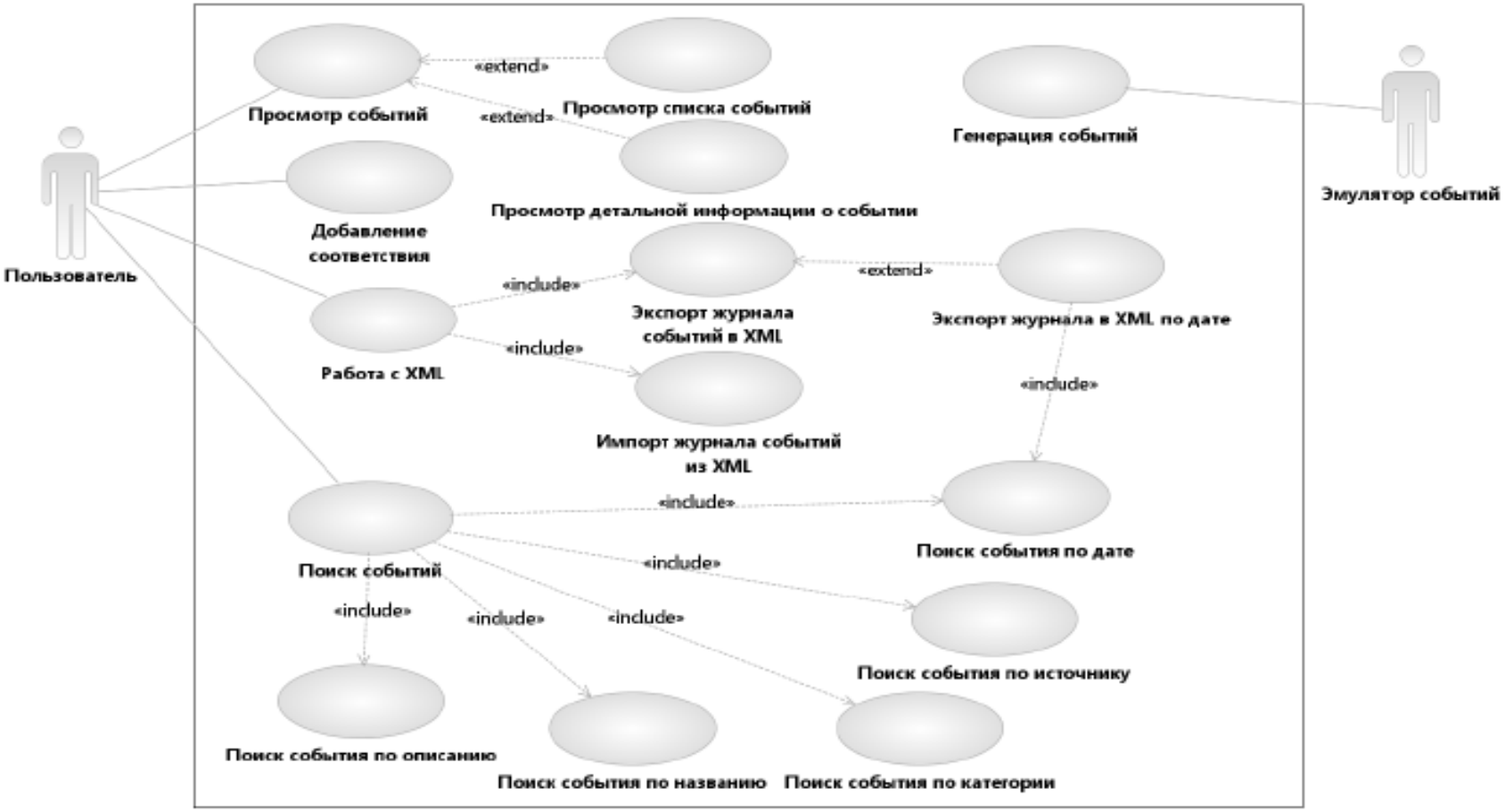
Рабочий процесс Определение требований

- Перечисление кандидатов в требования
- Осознание контекста системы
- Определение функциональных требований (в виде прецедентов)
- Определение нефункциональных требований

Модель предметной области – пример ЖСС



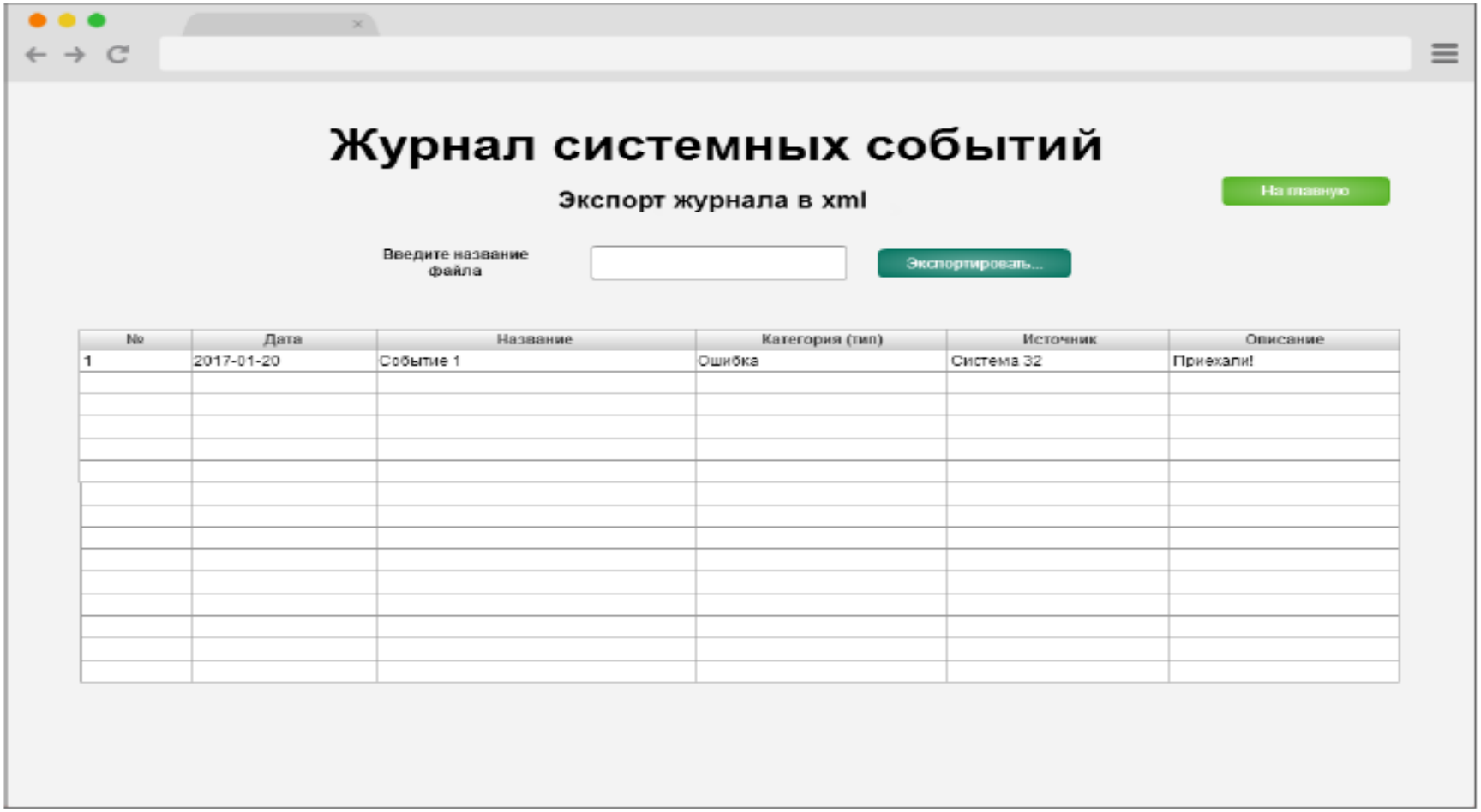
Модель прецедентов – пример ЖСС



Спецификация прецедента – пример ЖСС

Краткое описание: Предназначен для экспорта журнала событий в документ с расширением XML
Главные актеры: Пользователь
Второстепенные актеры: Нет
Предусловие: 1. Выполнен прецедент Работа с XML
Основной поток: 1. Прецедент начинается после выбора пользователем 2. Система проверяет введенные параметры пользователем 3. Система формирует пользователю документ в формате XML с информацией о событиях из БД 4. Система загружает документ в формате XML пользователю 5. Система отображает документ в формате XML пользователю
Постусловие: Нет
Альтернативные потоки: 1. Прецедент начинается после выбора пользователем 2. Система проверяет введенные параметры пользователем 3. Система выдает ошибку пользователю
Прецедент: Импорт журнала событий из XML

Прототип ПИ – пример ЖСС



Рабочий процесс Анализ (требований)

- Цель:
  - уточнить спецификацию требований;
  - перейти от языка заказчика к языку разработчика, анализ внутренних механизмов системы;
  - структурирование требований для дальнейшей разработки ПО

- Шаги:
  - Анализ архитектуры
  - Анализ кооперации
  - Анализ классов
  - Анализ пакетов

Анализ архитектуры

- Идентификация пакетов анализа
- Идентификация сервисных пакетов
- Определение зависимости между пакетами анализа
- Определение очевидных классов сущностей
- Определение общих специальных требований

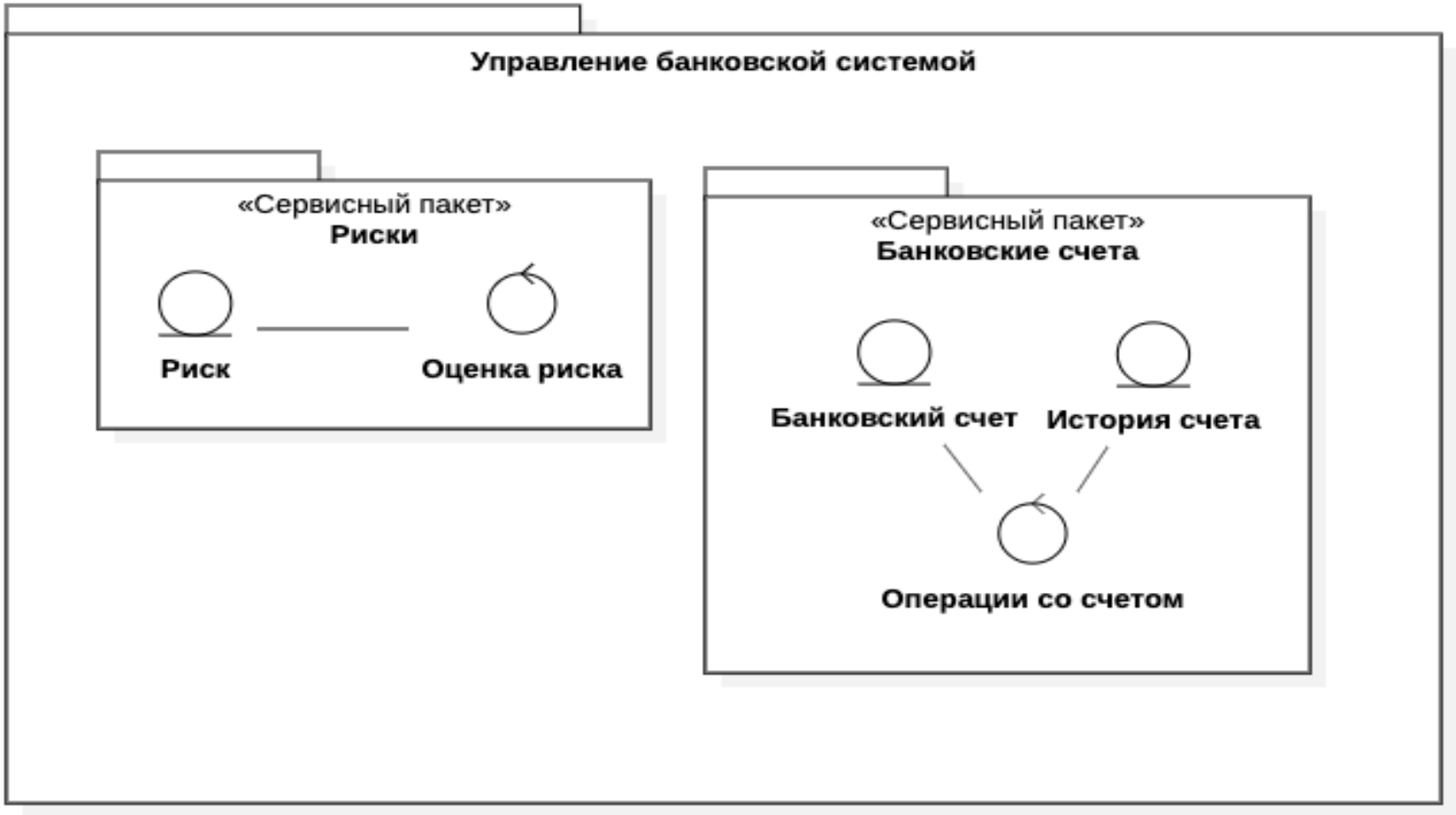
Пакеты анализа

- Пакет анализа:
  - сильная связность и слабое сцепление
  - на основе функциональных требований бизнеса, множество прецедентов для 1 актера / бизнес-процесса / обобщения и расширения.
  - для 2-х верхних уровней приложения.

- Если >= 1 пакета используют общий класс анализа, то класс надо вынести в отдельный пакет анализа или просто отдельно

- Сервисный пакет:
  - набор функциональности для клиента;
  - содержит множество функционально связных классов;
  - неделим при приобретении;
  - может повторно использоваться;
  - взаимозаменяем.

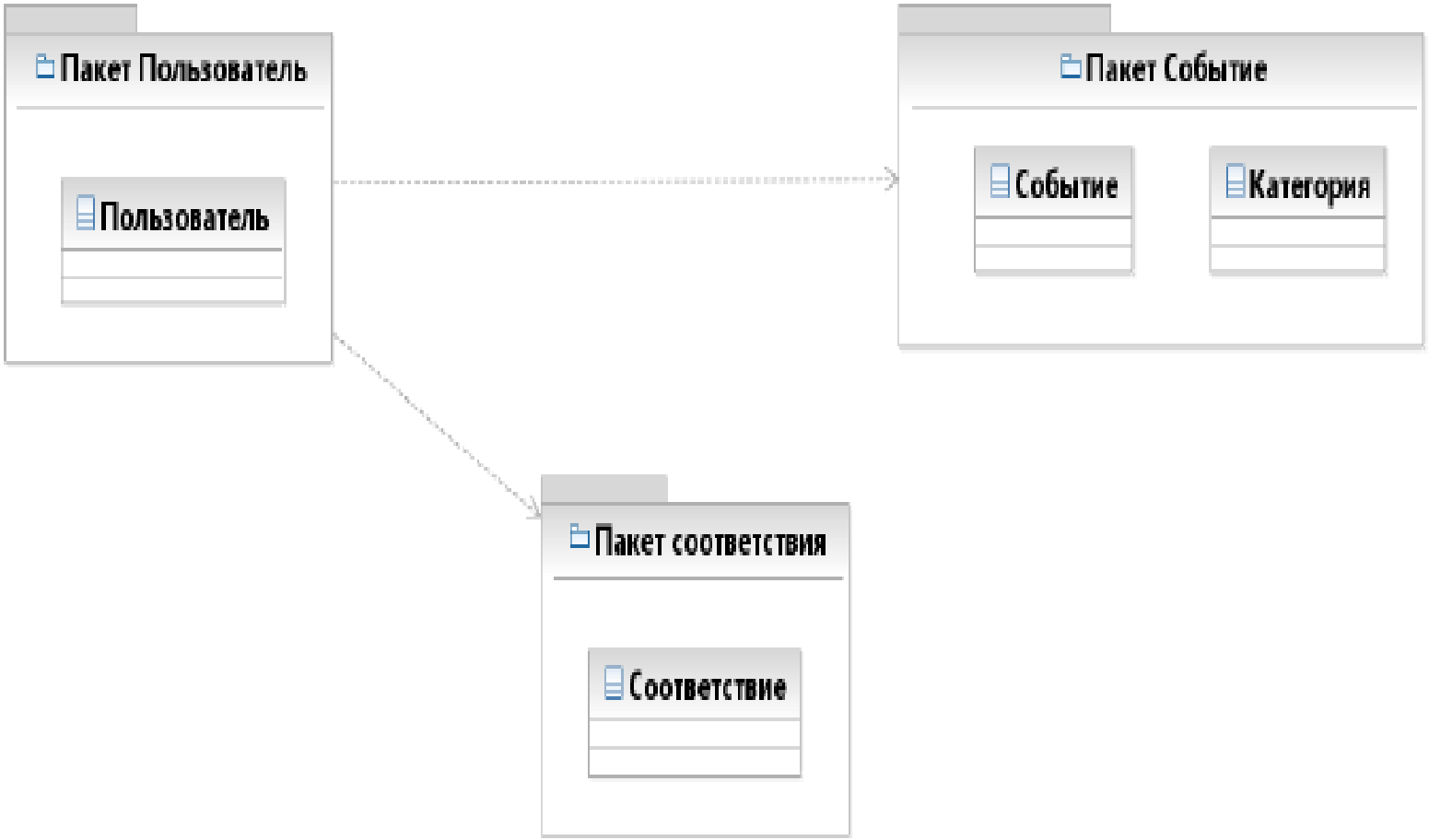
Анализ архитектуры - пример



Зависисмости между пакетами анализа

- связи зависимости;
- высокая связность и низкое сцепление;
- деление на уровни.

Пакеты анализа – пример ЖСС



Определение общих специальных требований

- хранение данных (скорость, частота, ...);
- распределение и распараллеливание разработки;
- безопасность;
- устойчивость к сбоям;
- упрвление транзакциями.

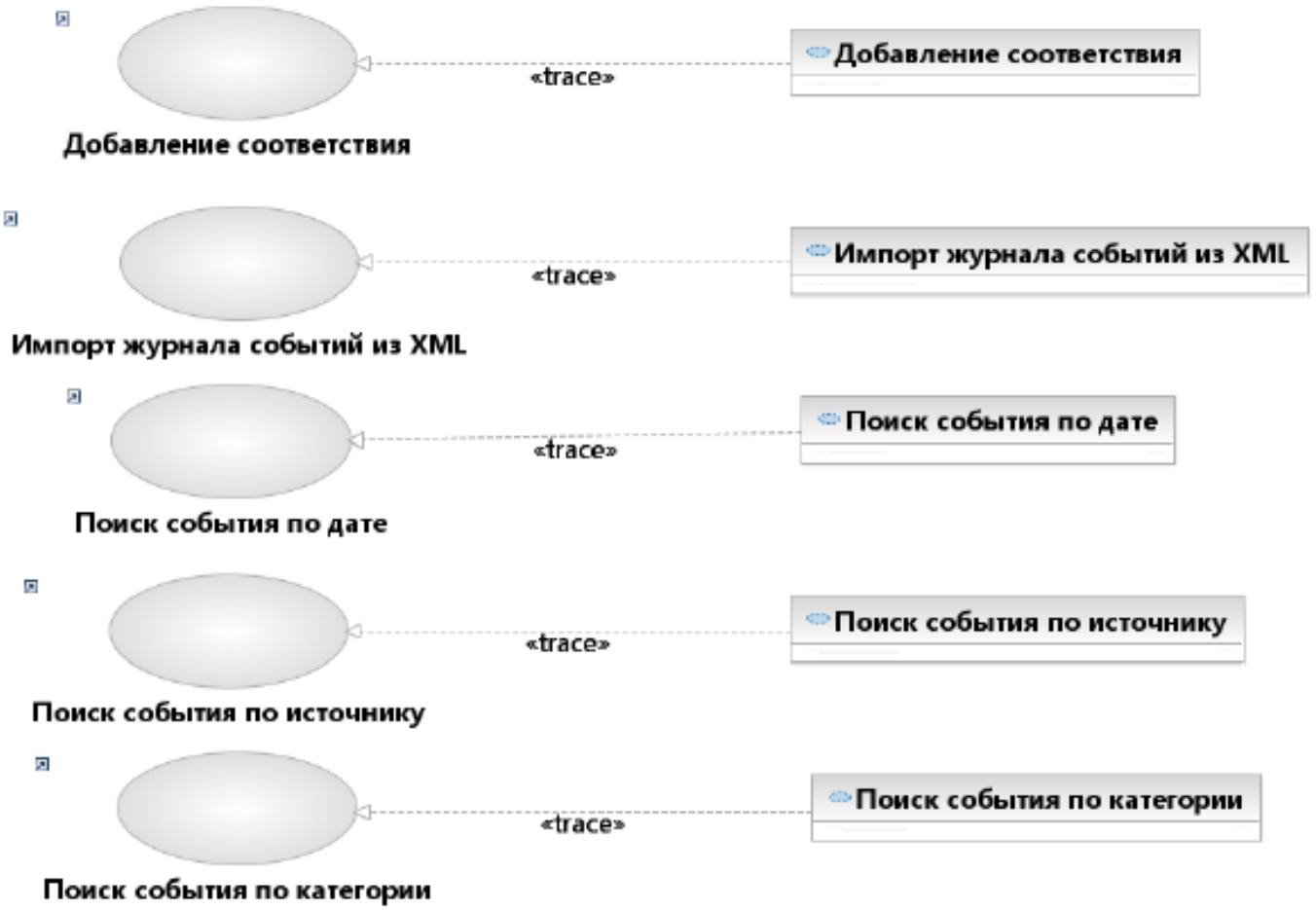
  – Влияют на проектирование и реализацию, характеристики требований приписаны к любому классу или кооперации, на которую ссылаются.

Анализ кооперации

- Кооперация – реализация прецедента
  - Статика – диаграмма классов-участников
  - Динамика – диаграмма взаимодействия или последовательностей

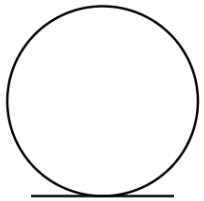
- Трассировка кооперации от прецедентов
- Определение классов анализа
- Описание взаимодействия объектов анализа
- Определение специальных требований

Трассировка кооперацій от прецедентов – пример ЖСС

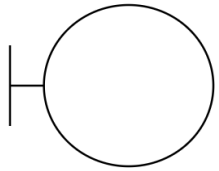


Классы анализа

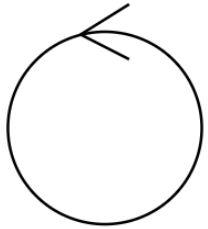
- Класс сущности - для моделирования долгоживущей системы, часто сохраняемой информацию о человеке / объекте / событии реального мира. Может иметь сложное поведение. Показывает логическую структуру данных.



- Граничный класс - для моделирования взаимодействия между системами и актерами:получение / передача информации / запросов. Соответствуют пользовательскому интерфейсу / устройству/ коммуникации / API на внешнем уровне без реализации.



- Управляющий класс - координация / последовательность /взаимодействие / управление другими объектами для прецедента. Обрабатывают и координируют действия и потоки управления; реализуют бизнес-логику.



Определение классов анализа

- определение классов сущностей (10 - 20 штук) на основе модели предметной области или бизнес-объектов, которые участвуют в прецедентах.
  Для них определяют атрибуты и ассоциации.

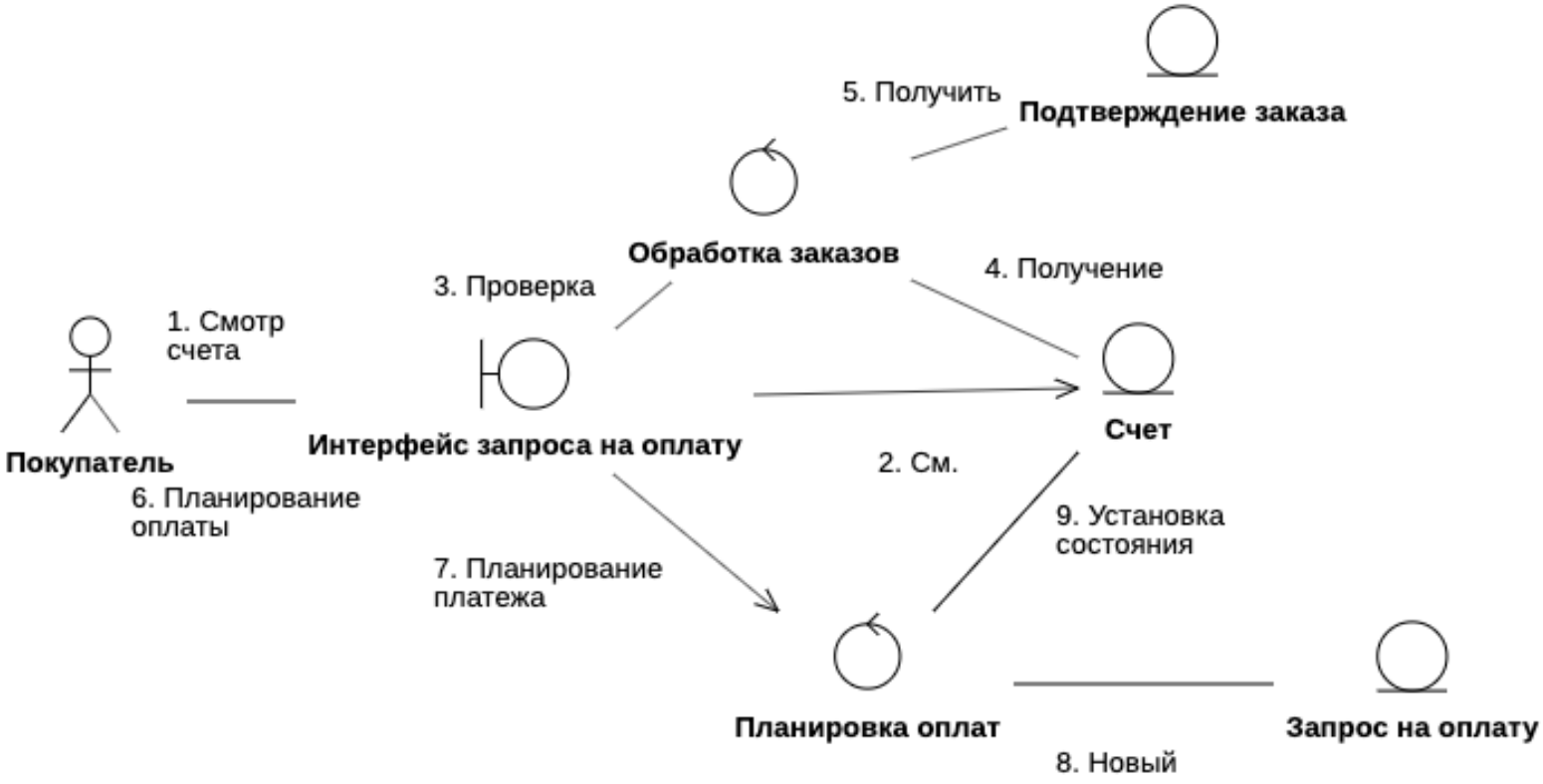
- определение по одному основному граничному классу на любого актера человека - это главная форма его интерфейса,
- определение по одному основному граничному классу на любую внешннюю систему - это коммуникационный интерфейс. Если N уровней коммуникации, то по одному граничному классу на уровень.

- Определение по одному управляющему классу на кооперацию (может быть управляющий класс в граничном или больше двух управляющих классов для сложных кооперациий)

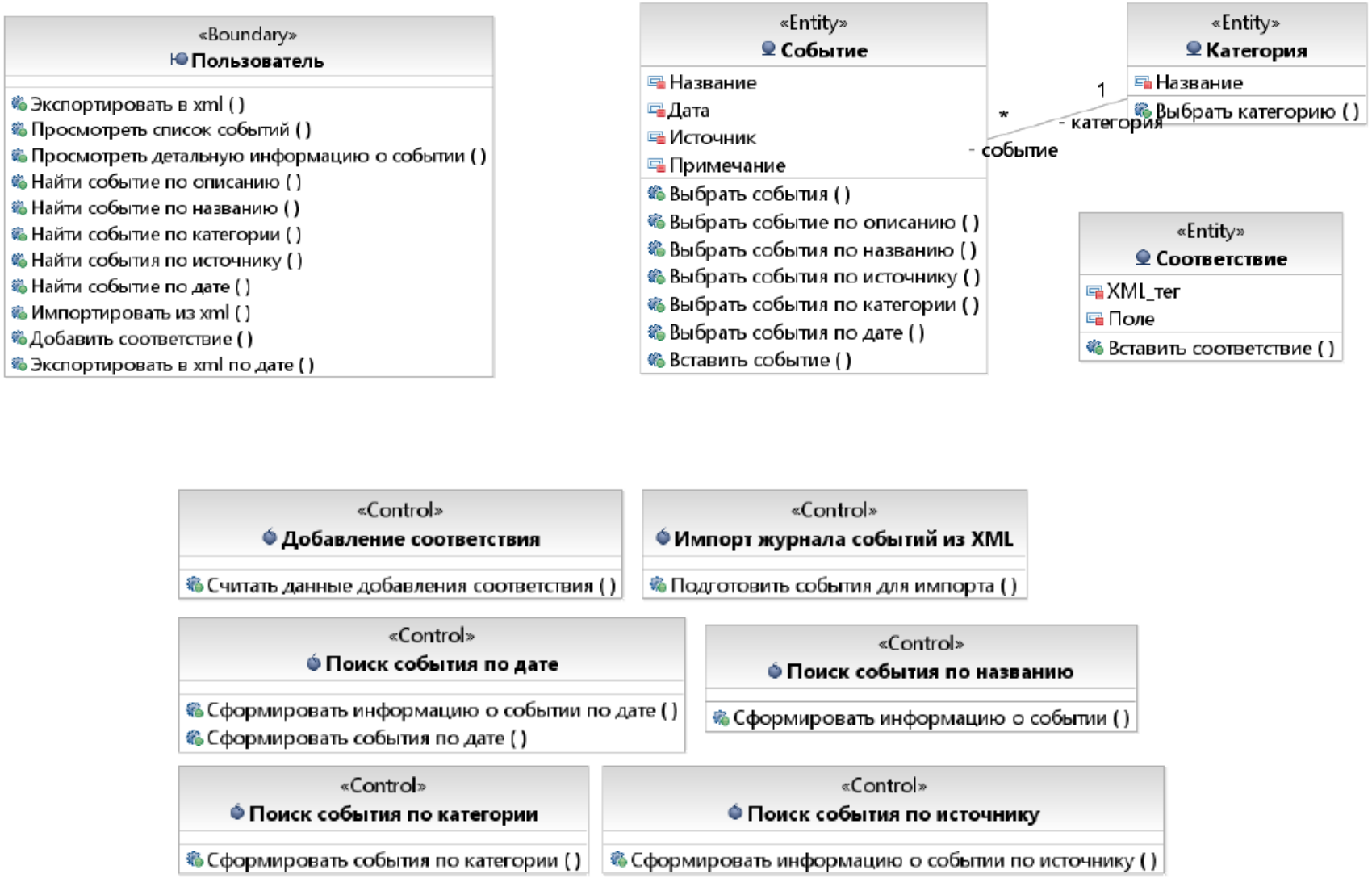
Описание взаимодействия объектов анализа

- Построение диаграммы кооперации для прецедента.
- Если существует N потоков или подпотоков, то по 1 диаграмме кооперацией на поток.
- Выбрать участников кооперации: актер(ы), граничный – управляющий – сущность(и).
- Кооперация создается событием от актера.
- Направленные ассоциации.

Диаграмма взаимодействия - пример



Классы анализа – пример ЖСС



Кооперация – пример ЖСС

Импорт журнала событий из XML

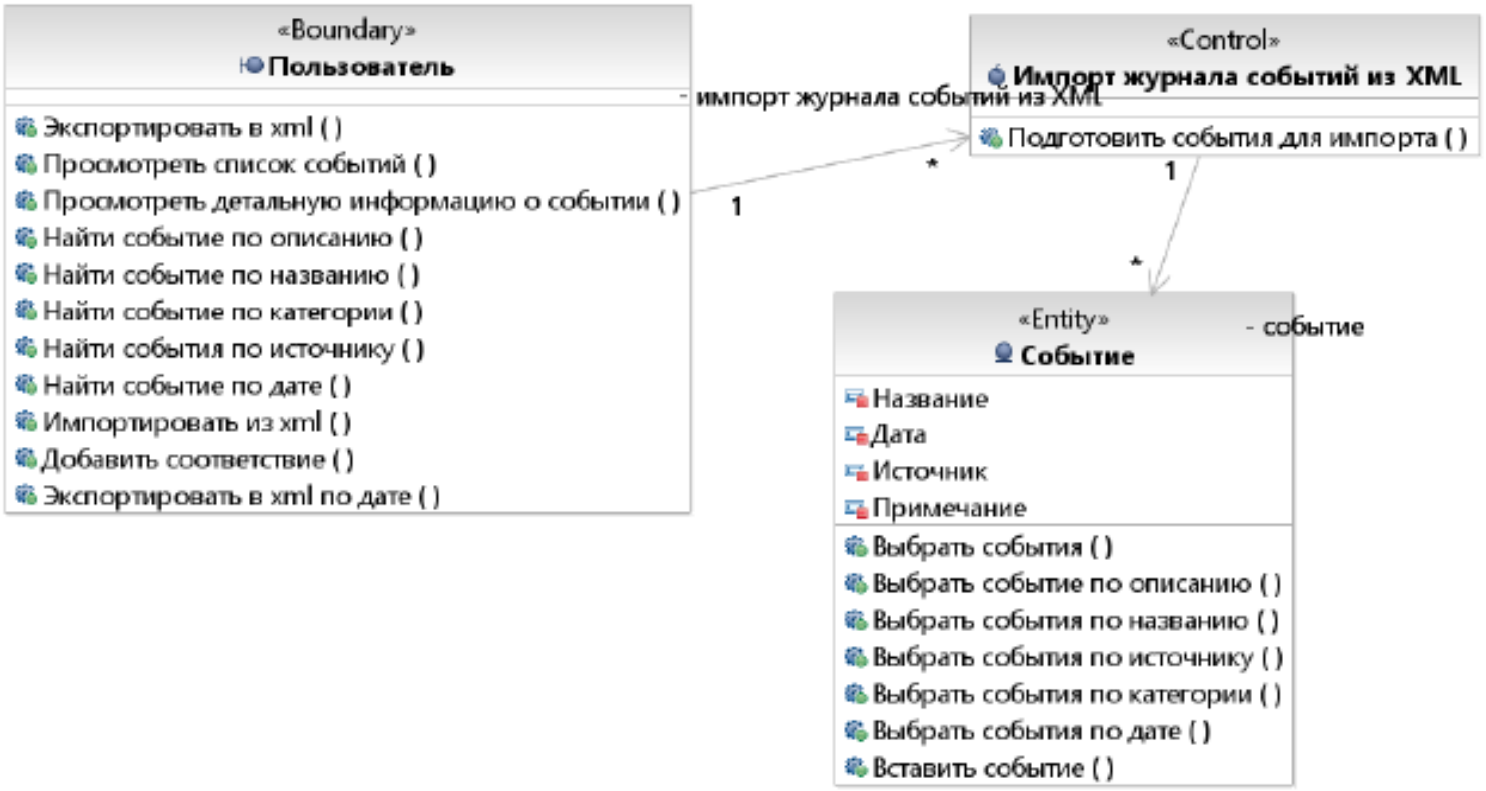
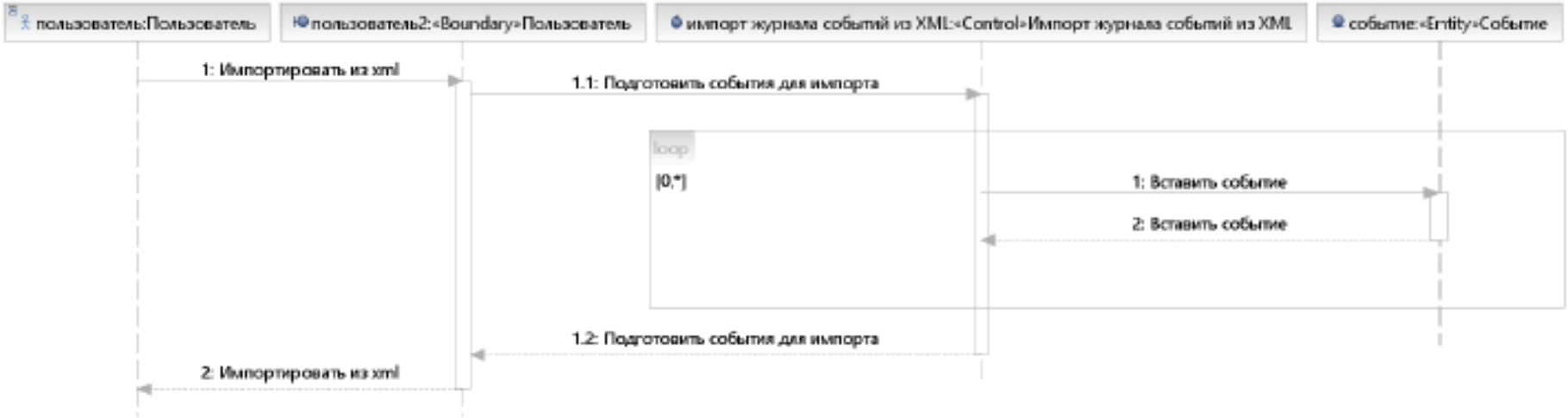


Диаграмма последовательностей кооперации – пример ЖСС



Анализ классов

1. Определение ответственности - из комбинации всех ролей класса во всех прецедентах
2. Определение атрибутов
   - концептуальные типы
   - если сложные и много атрибутов, то часть атрибутов выносят в отдельный класс
   - общие атрибуты у N классов выносят в отдельный класс

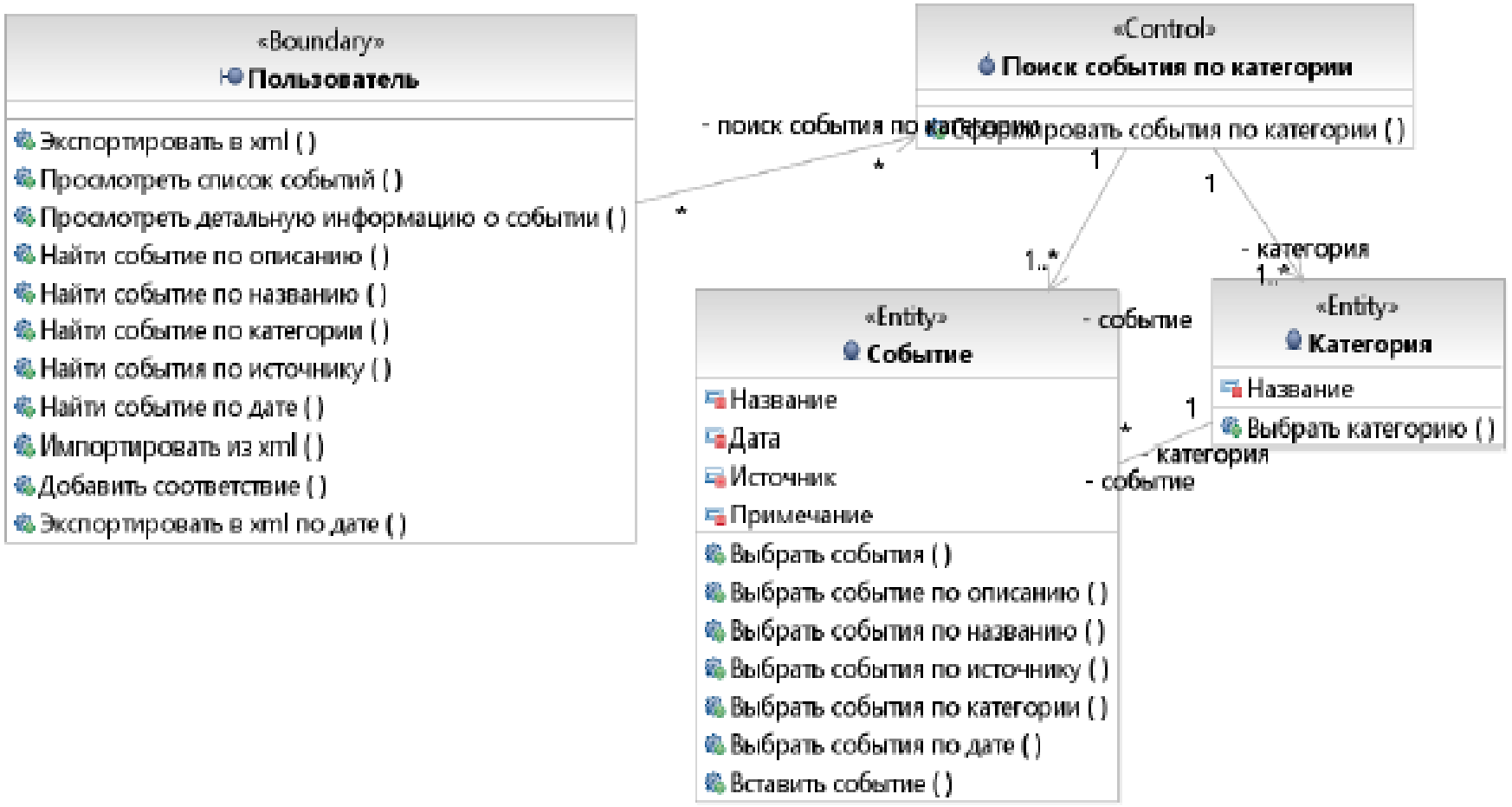
Атрибуты классов:

Классы сущностей	Граничные классы	Управляющие классы
- просты; - от предметной области; - полезные исходные данные	- Свойства коммуникационных протоколов - поля ввода	- редки; - аккумулируют / порождают значения для прецедента

Анализ классов

3. Определение ассоциаций и агрегаций
   - для указания взаимодействия объектов в кооперации
   - определяются: множество ассоциаций; роли; арность; класс ассоциаций;
   - определяют агрегацию, если: - физическая вложенность (авто и детали); - коллекция (родитель и дети).
4. Определение обобщений
   - Получить разделяемое или общее поведение классов.
5. Определение специальных требований

Анализ классов – пример ЖСС

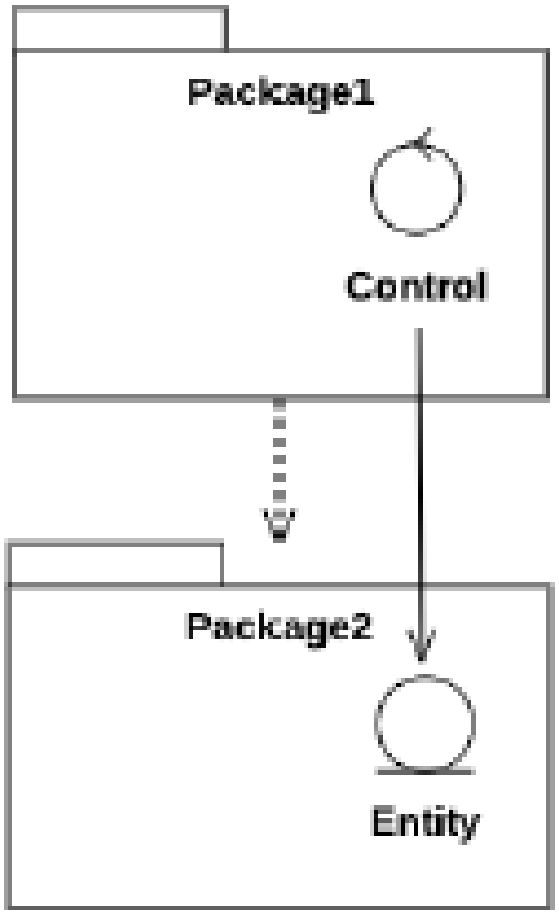


Анализ пакетов

- Цель:
  - независимость пакета от других;
  - реализация пакетом его прецедентов / классов предметной области;
  - определить зависимости пакетов (оценка внесенных изменений);
  - пакет содержит функционально ориентированные классы.

- Возможно перемещение классов

Анализ пакетов - пример



Анализ пакетов – пример ЖСС