

Учебный курс

Проектирование информационных систем

Лекция 10

кандидат технических наук, доцент

Грекул Владимир Иванович

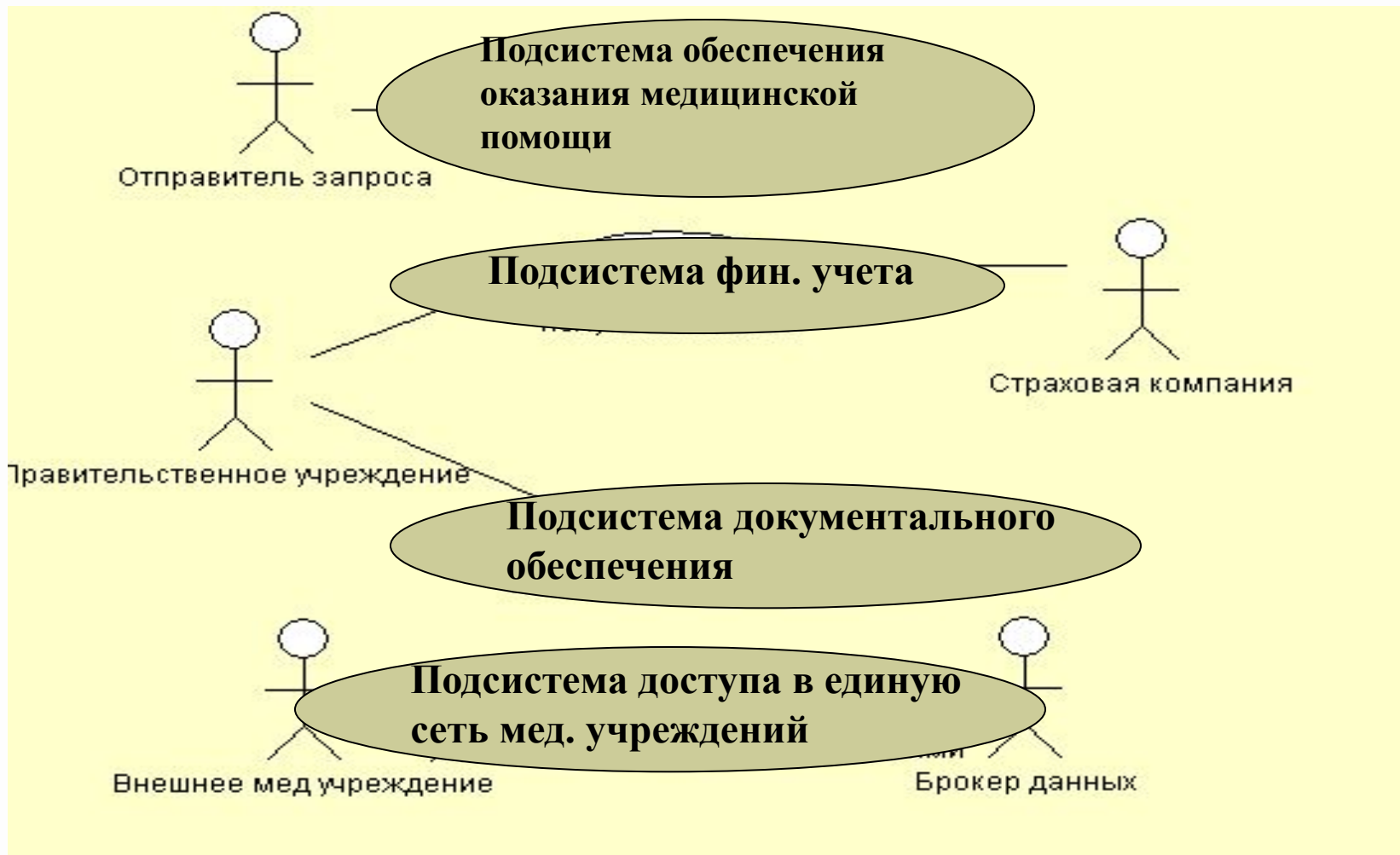
Разработка требований к системе

Преобразование бизнес-модели в модель системных прецедентов

Элементы бизнес-модели	Элементы модели системных прецедентов
Бизнес-прецеденты	Подсистемы
Внешние исполнители	Исполнители
Внутренние исполнители	Исполнители или прецеденты
Процессы, выполняемые внутренними исполнителями	Прецеденты

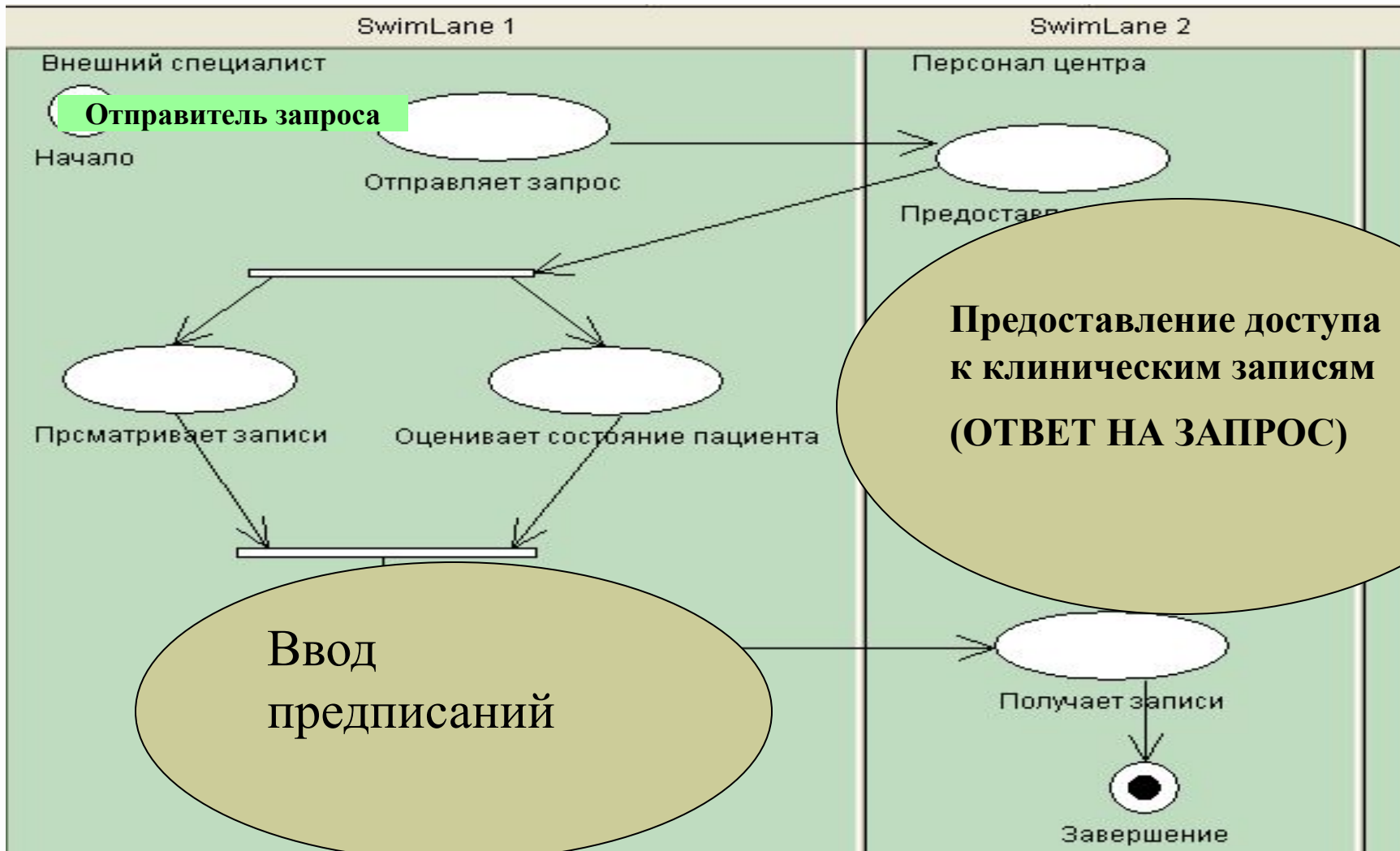
Выделение подсистем ИС

Модель бизнес-прецедентов, составляющих обслуживание пациента



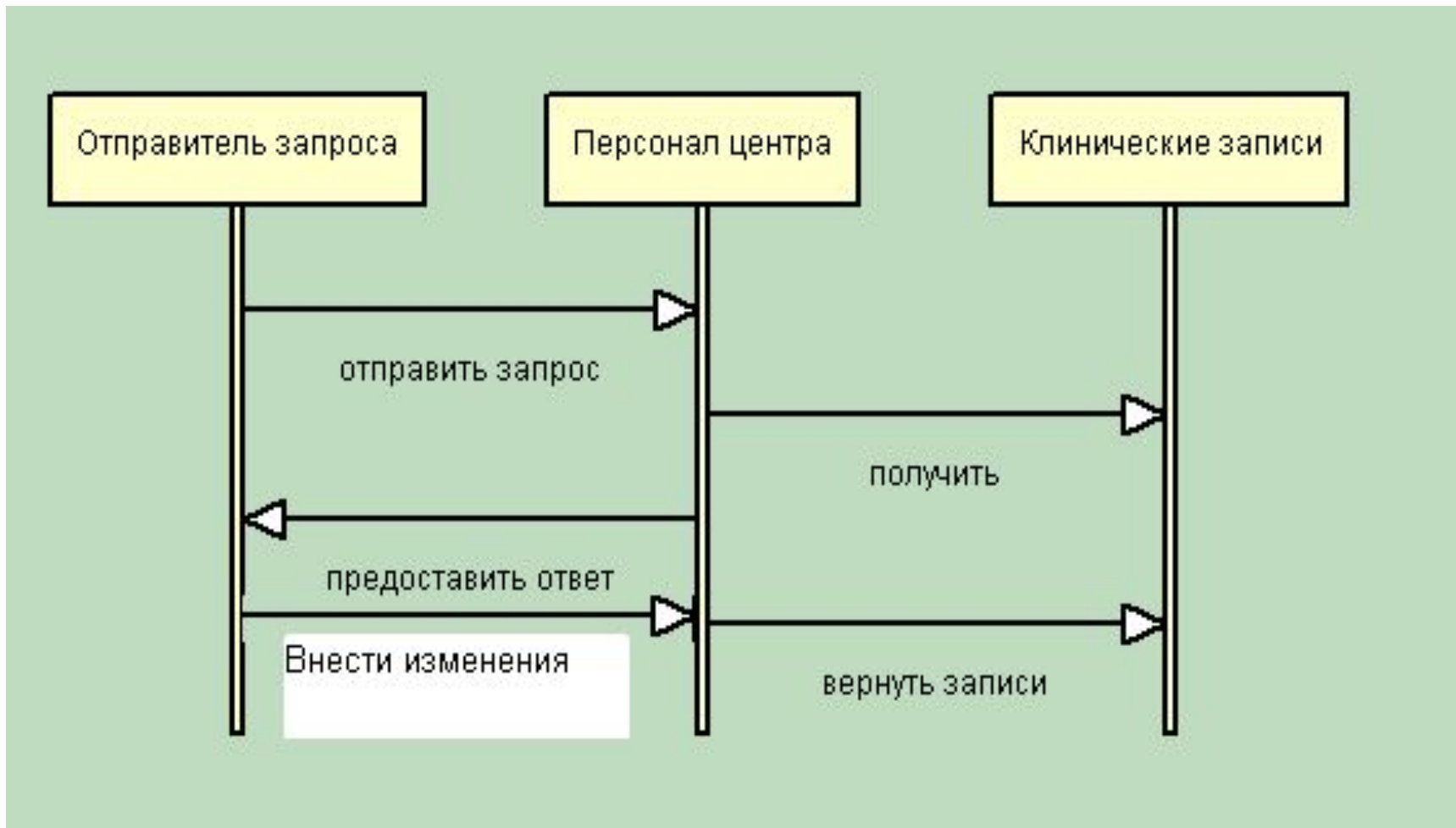
Выделение системных прецедентов

(диаграмма деятельности для прецедента «Оказание медицинской помощи»)

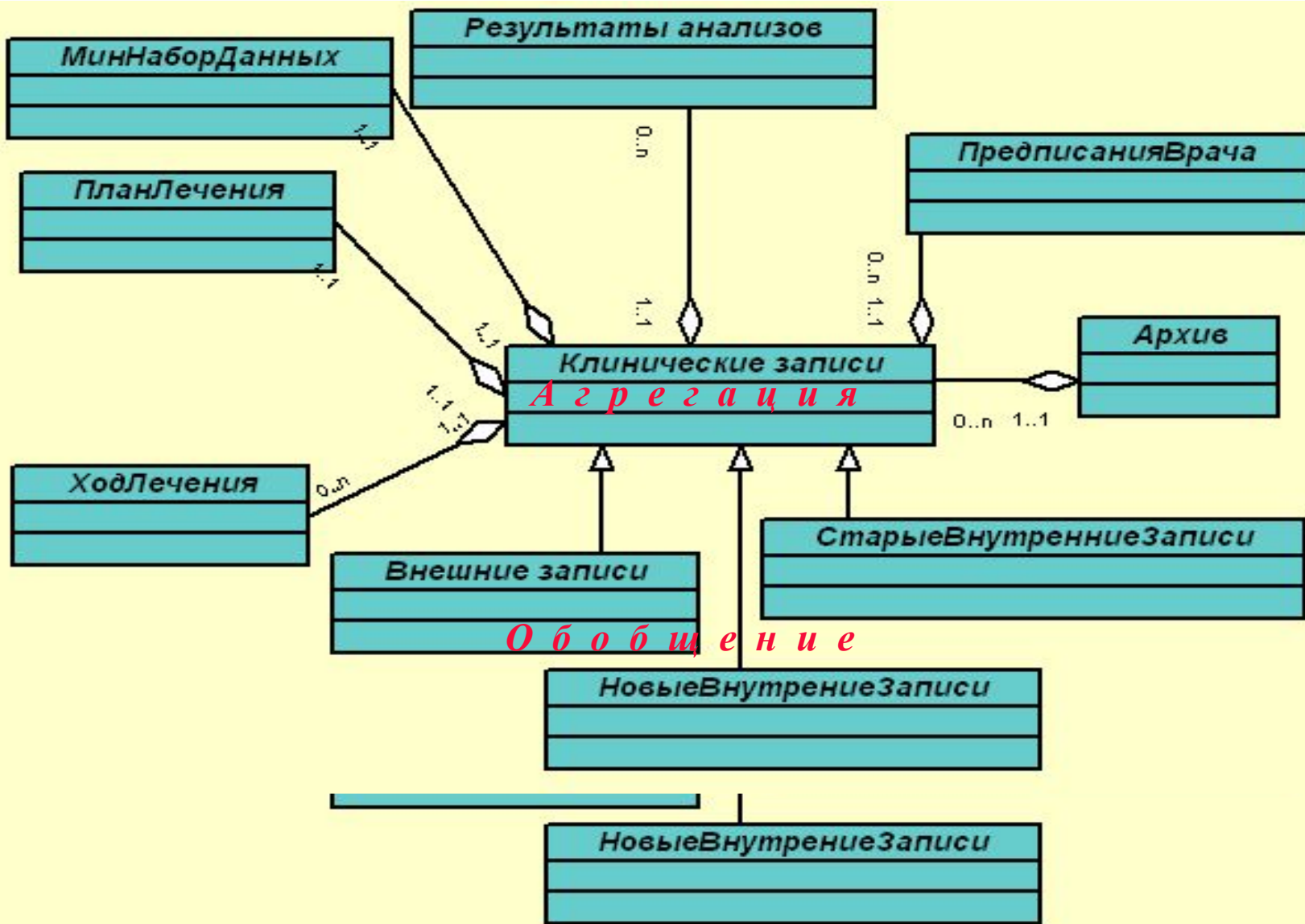


Описание функций

Диаграмма последовательности для прецедента «Ответ на запрос»

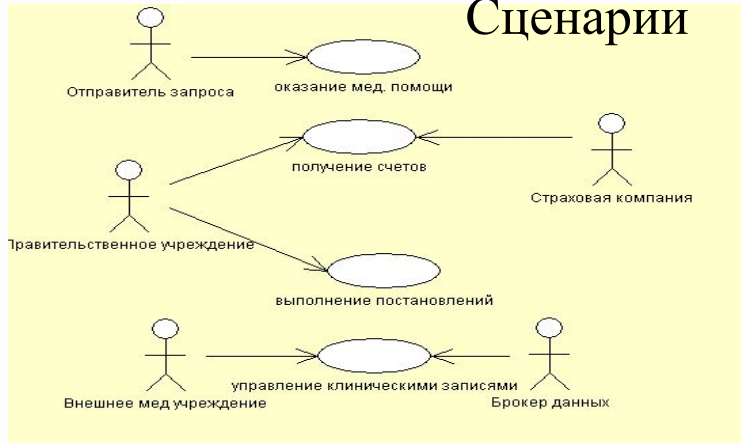


Разработка концептуальной модели данных



Модель анализа

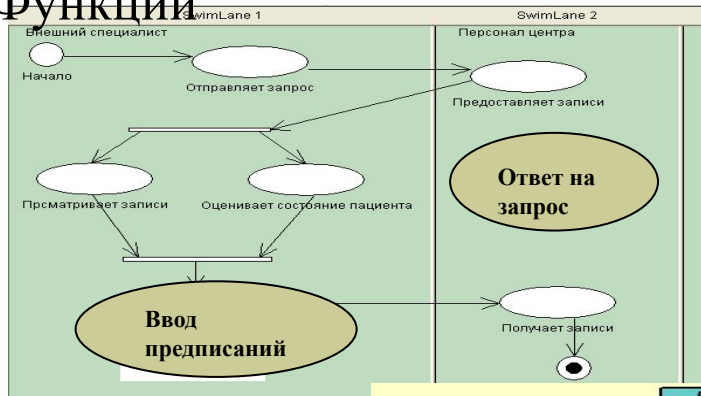
Сценарии



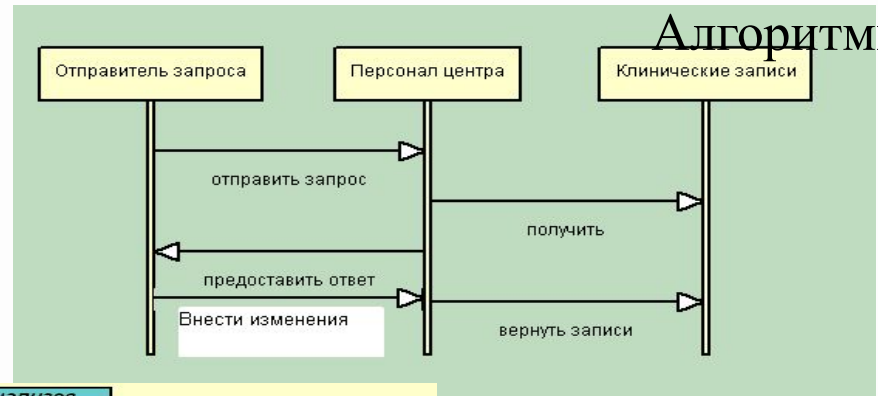
Подсистемы



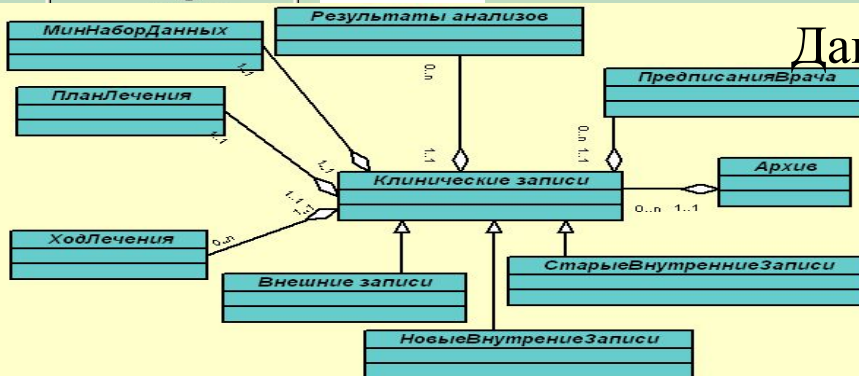
Функции



Алгоритмы

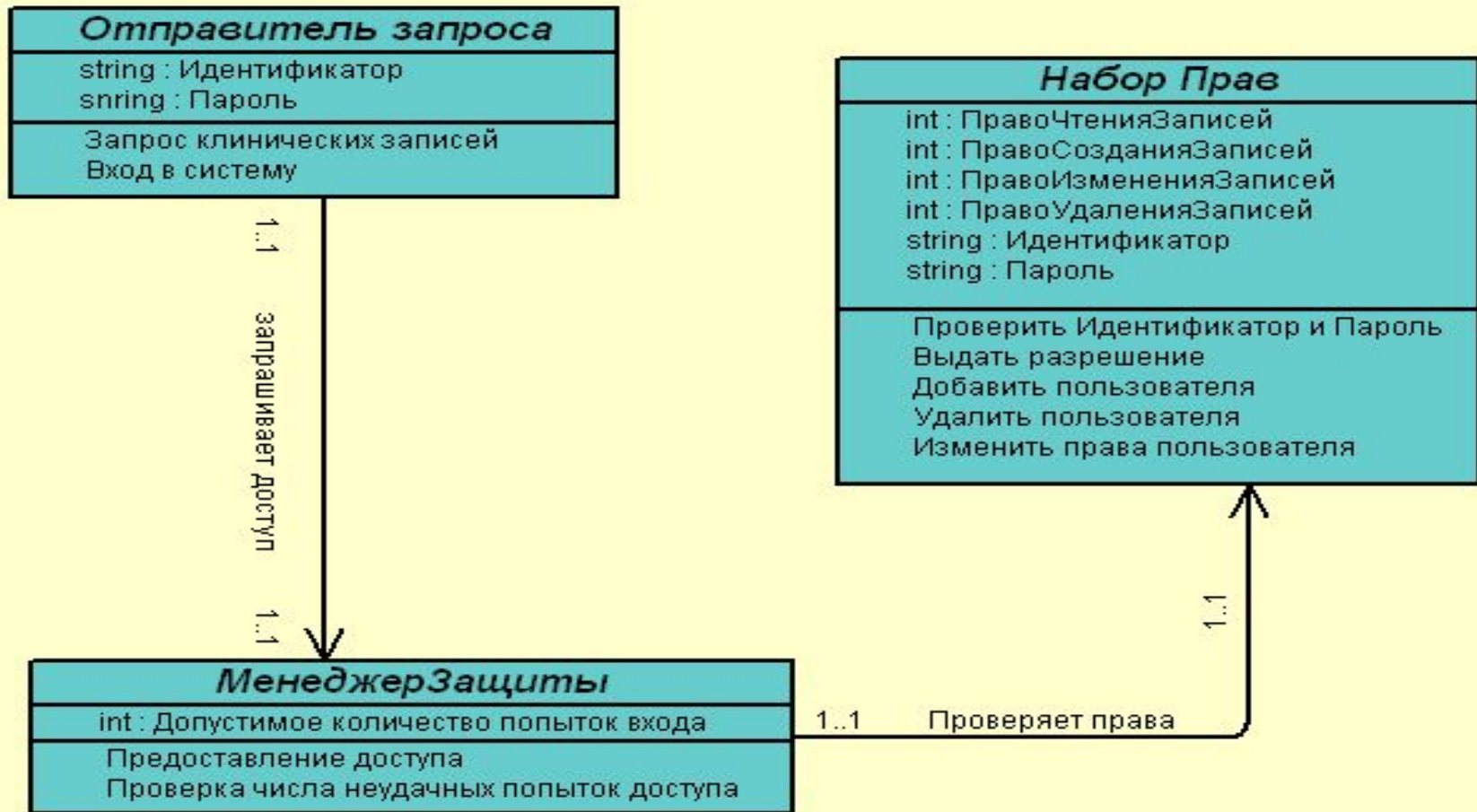


Данные



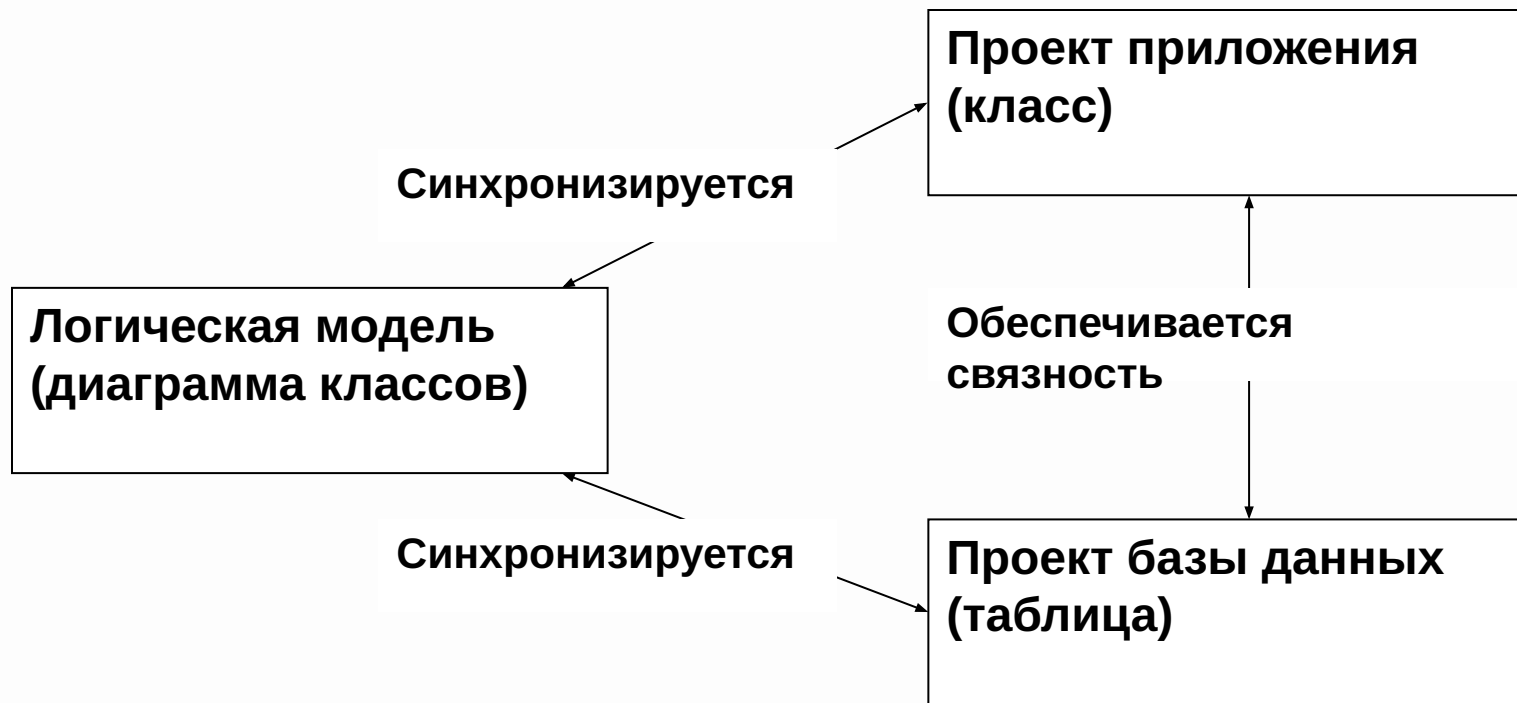
Анализ требований и проектирование системы – детальное определение классов

Диаграмма классов «Защита доступа»



Разработка моделей базы данных и приложений

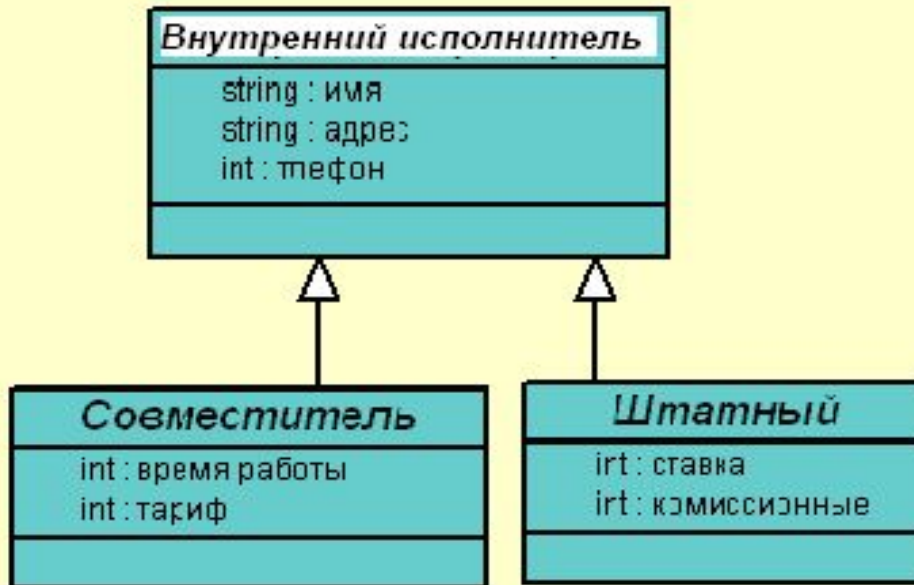
Связь между проектами базы данных и приложений



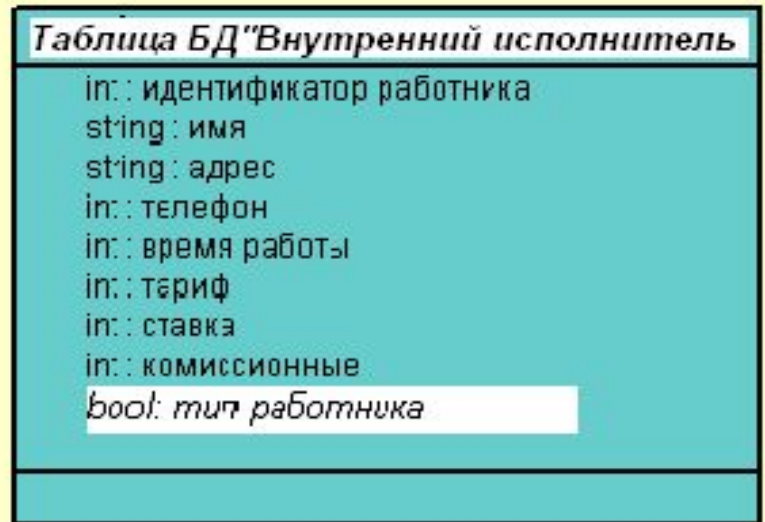
Разработка моделей базы данных и приложений

Преобразование иерархии в таблицу

Логический проект

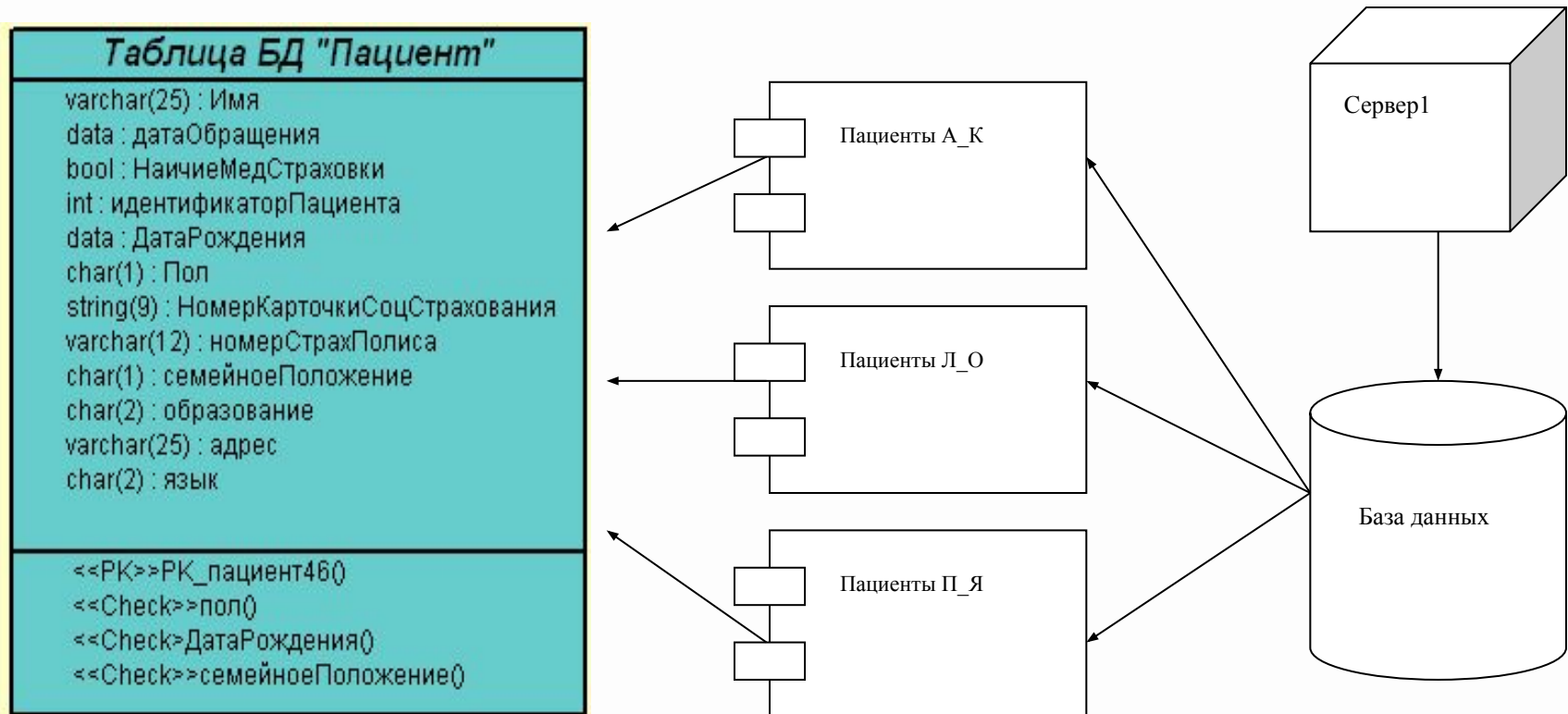


Проект базы данных



Проектирование физической реализации системы

Фрагмент диаграммы развертывания ИС



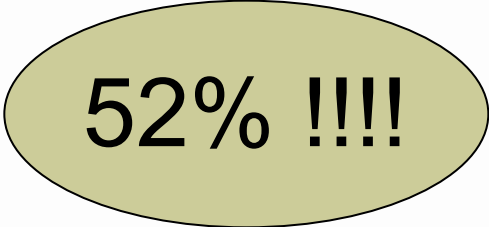
Управление требованиями

1. Определения и классификация требований
2. Процессы формирования и изменения требований
3. Связи между требованиями

Причины провала проектов

- Неполные требования 13.1%
- Недостаточное участие пользователей 12,4%
- Недостаток ресурсов 10,6%
- Нереалистические ожидания 9,9%
- Недостаток поддержки от руководства 9,3%
- Изменение требований/спецификаций 8,7%
- Недостаточное планирование 8,1%
- Потеря актуальности 7,5%

Standish Group



52% !!!!

Определение и классификация требований

Требование – условие или возможность, которой должна соответствовать система.

Функциональные требования – определяют действия, которые должна быть способна выполнить система (без рассмотрения физических связей).
Определяют внешнее поведение системы.

Функциональные требования используются для выражения поведения системы путем задания предпосылок и возможностей, ожидаемых в качестве результата.

Нефункциональные требования описывают только атрибуты системы или среды.

Нефункциональные требования служат для создания системы с приемлемым качеством.

Модель FURPS+

- **Functionality** (функциональность)
- **Usability** (применимость)
- **Reliability** (надежность)
- **Performance** (производительность)
- **Supportability** (пригодность к эксплуатации)

+

- **Проектные ограничения**
- **Требования к исполнению**
- **Требования к интерфейсу**
- **Физические требования**

Нефункциональные требования

- **Применимость (*Практичность*)**

Требования практичности связаны с человеческим фактором— эстетикой, легкостью изучения и использования, с согласованностью пользовательского интерфейса, пользовательской документации и обучающих материалов.

- ***Надежность***

Требования надежности связаны с частотой появления и серьезностью ошибок, возможностью восстановления, предсказуемостью и точностью.

- ***Производительность***

Требования производительности накладывают ограничения на функциональные требования - требования, задающие частоту, скорость, точность, время отклика, объем памяти.

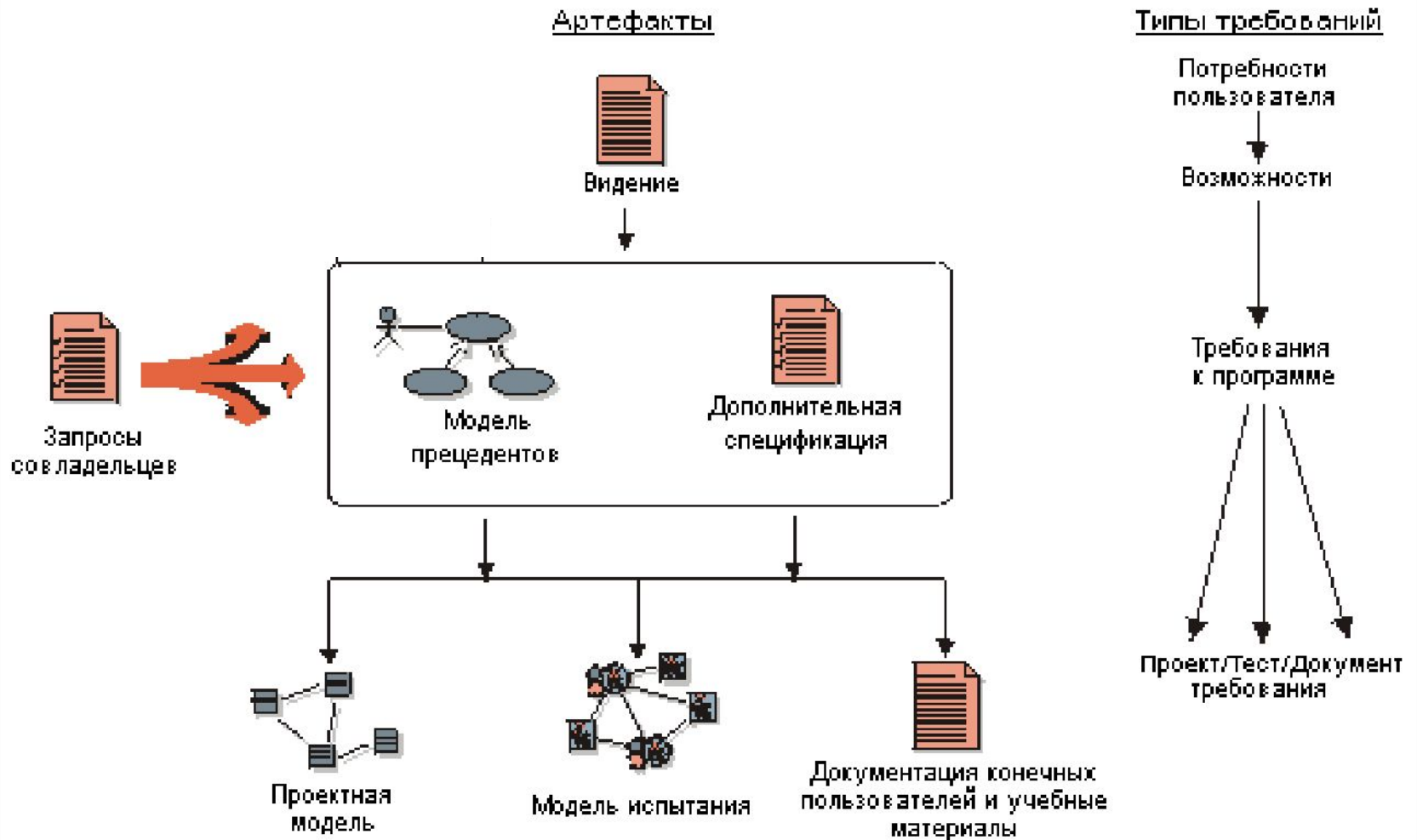
- ***Возможность поддержки***

Требования этого типа связаны с возможностью контроля состояния, эксплуатации и другими параметрами качества, необходимыми для организации эксплуатации и обновления системы после ее выпуска.

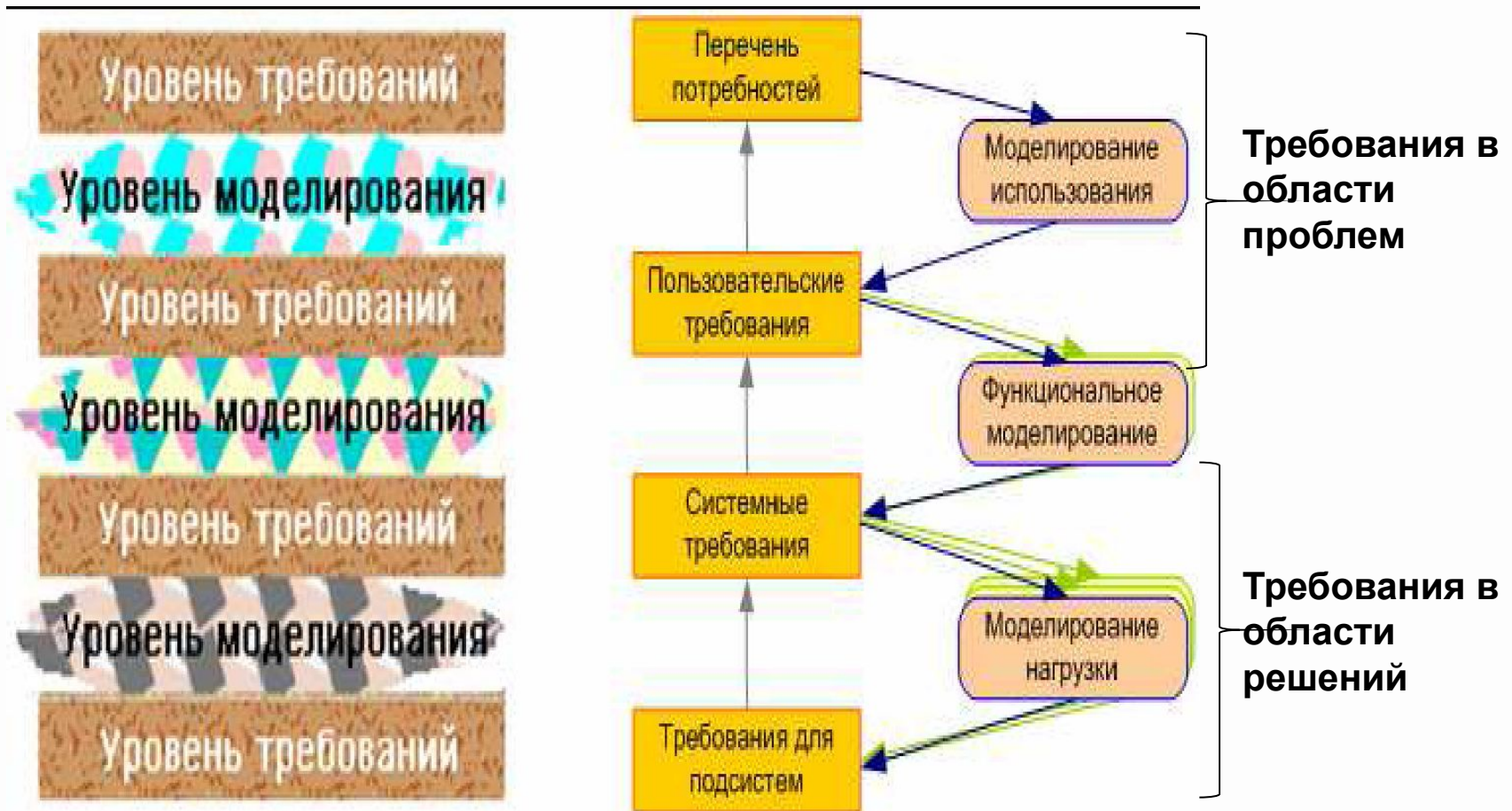
Цели разработки требований

- Разработчики системы вместе с заказчиками и другими заинтересованными сторонами должны выработать единое мнение о том, что должна делать система.
- Разработчики должны полнее понять системные требования.
- Должны быть однозначно определены границы системы.
- Должна быть создана основа для планирования технического содержания итераций, а также оценки стоимости и времени разработки системы
- Должен быть определен пользовательский интерфейс системы

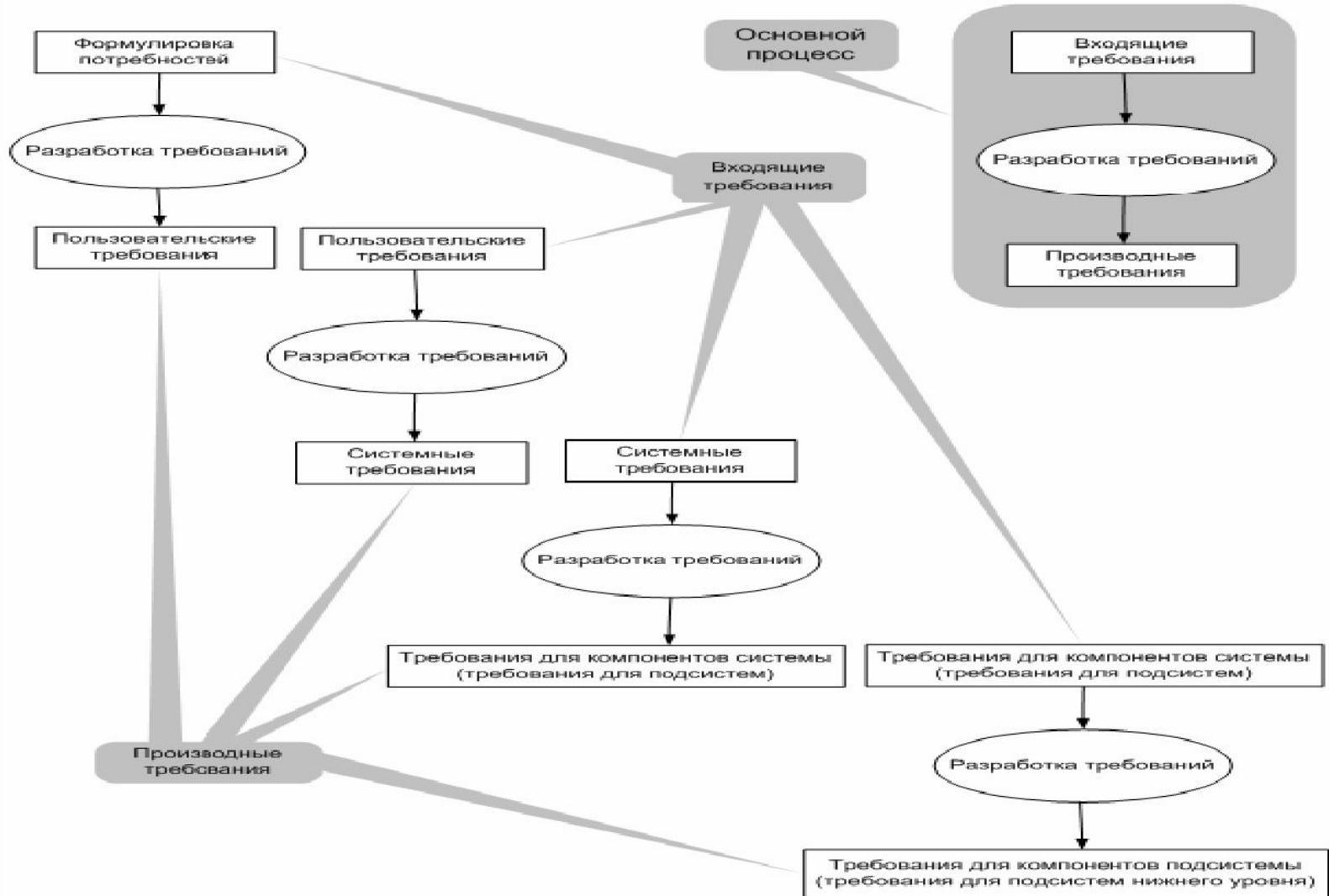
Типы требований и артефакты RUP



Пользовательские и системные требования



Входящие и производные требования



Атрибуты требований

Позволяют не перегружать требование излишними деталями

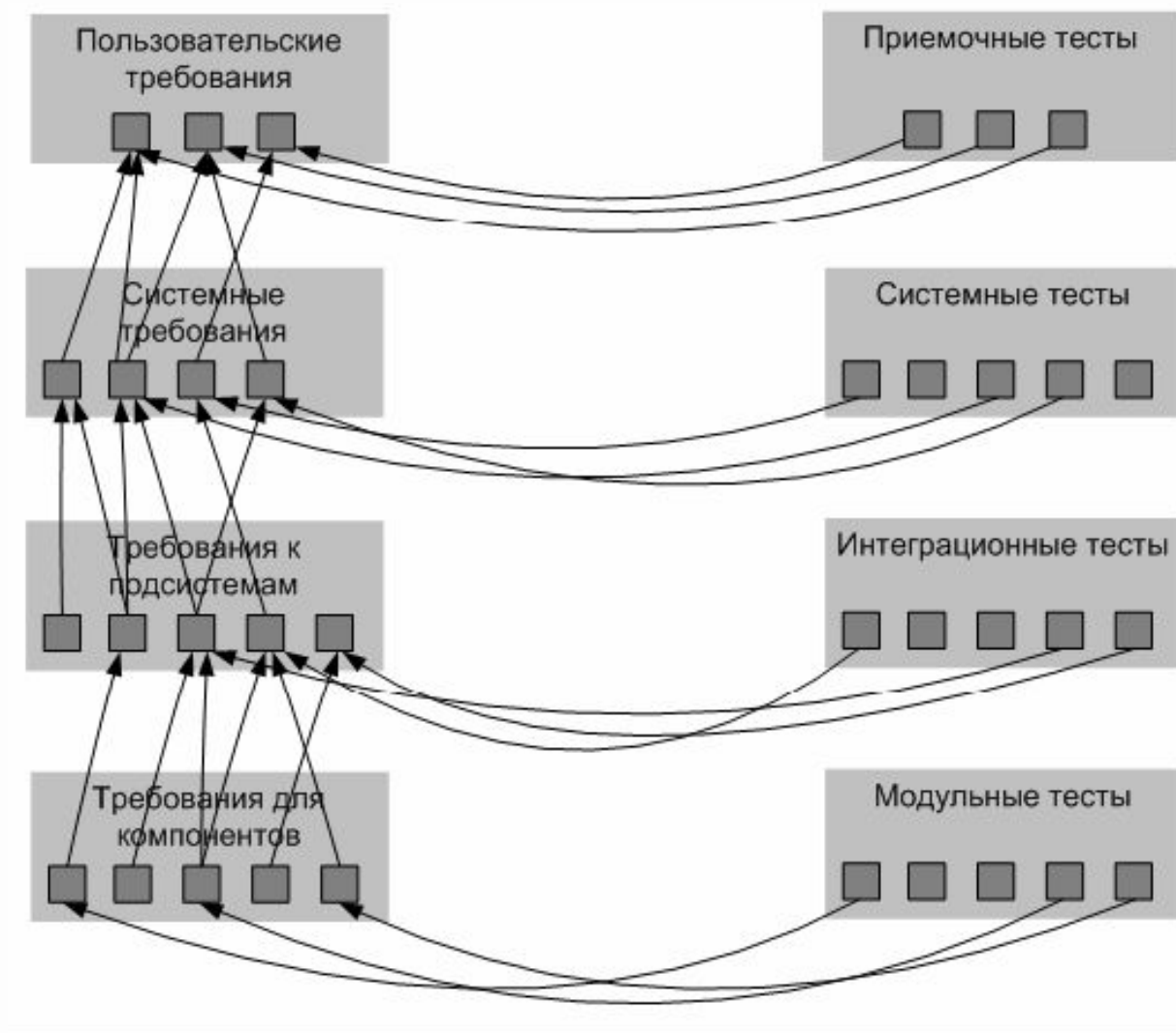
[SH234] Система управления скорой помощью должна быть способна принимать до 100 ста вызовов одновременно

Автор:	R. Thomas
Приоритет:	Обязательное
Релиз:	1
Статус рецензирования:	Одобрено
Возможность проверки:	Да
Способ проверки:	Симуляция, затем системные тесты

Категории атрибутов

Категория	Примеры значений
Идентификация	
<ul style="list-style-type: none">ИдентификаторНазвание	Уникальный номер требования (ID) Уникальное краткое название, характеризующее требование
Внутренние характеристики	
<ul style="list-style-type: none">Основной типКачественный подтипТип продукта/процессаКоличественный/качественный типФаза жизненного цикла	Функциональность, производительность, качество, окружение, интерфейс, ограничение, не требование Доступность, гибкость, целостность, ремонтпригодность, портативность, легкость поддержки, легкость использования, квалификация Продукт, процесс, данные, сервис Количественный, качественный Предварительная концепция, окончательная концепция, разработка, производство, интеграция/тестирование, внедрение/поставка/установка, функционирование, поддержка, удаление/демонтаж
Приоритет и важность	
<ul style="list-style-type: none">ПриоритетВажность	Ключевое, необходимое, дополнительное, желательное (Key, mandatory, optional, desirable) или Обязательное, рекомендуется, возможное, желательно (Must, Should, Could, Wish) Шкала от 1 до 10
Источник и владелец	
<ul style="list-style-type: none">Способ полученияИсточникВладелецСогласовано	Назначение, декомпозиция Название документа или имя заинтересованного лица Имя заинтересованного лица Имя человека
Контекст	

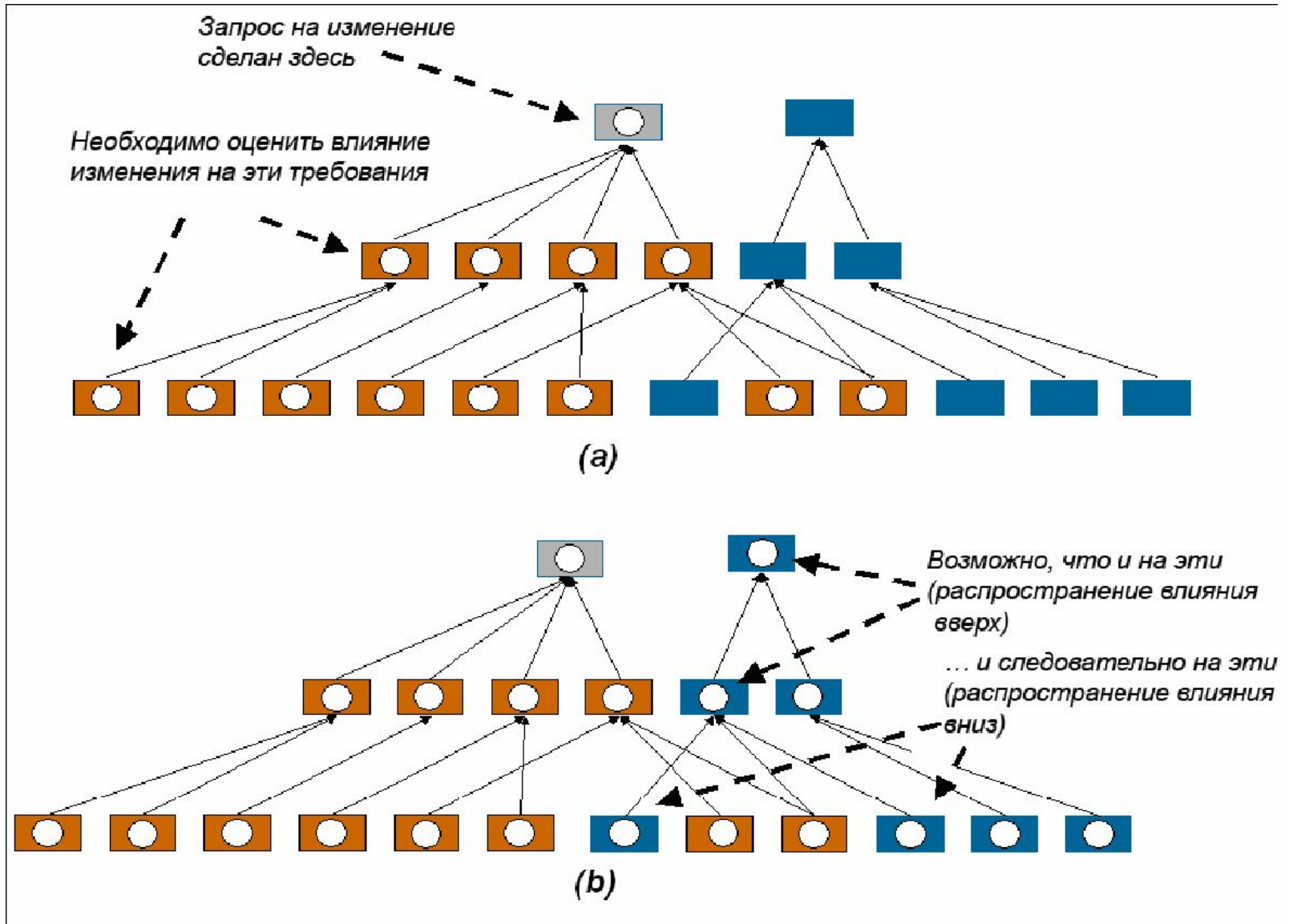
Связи между требованиями



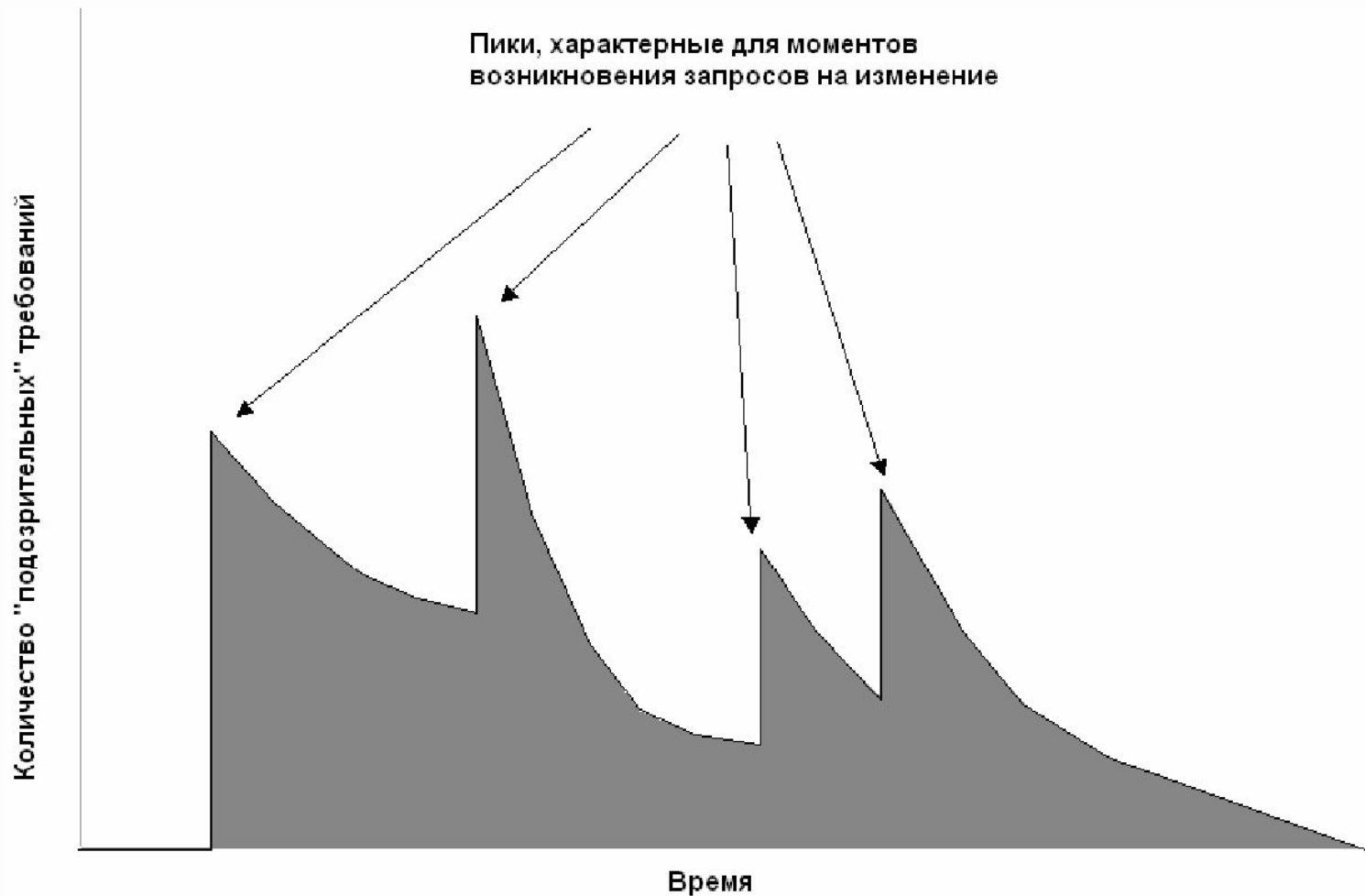
Аспекты анализа связей

Аспект	Что рассматривается	Когда применяется
Анализ влияния	Анализ входящих связей: что будет, если изменить требование?	Для оценки изменений
Анализ последствий	Анализ исходящих связей: действительно ли требование необходимо?	Для анализа экономической целесообразности
Анализ покрытия	Анализ любых связей: все ли учтено в требованиях (тестах, спецификациях...)?	Для поддержки проектирования и подготовки отчетности

Анализ связей в процессе управления изменениями



Динамика появления «подозрительных» требований



«Требования к требованиям»

- Требования должны быть четко сформулированы
- Требования должны быть исполнимыми в рамках проекта
- Требования должны быть проверяемыми
- Документ с требованиями должен быть структурирован таким образом, чтобы пользователь мог легко понять смысл каждого требования в контексте всего документа
- Формулировка каждого требования должна четко и точно отражать его суть и обеспечивать возможность устанавливать связи с другими требованиями

Рекомендации

При разработке требований, следует:

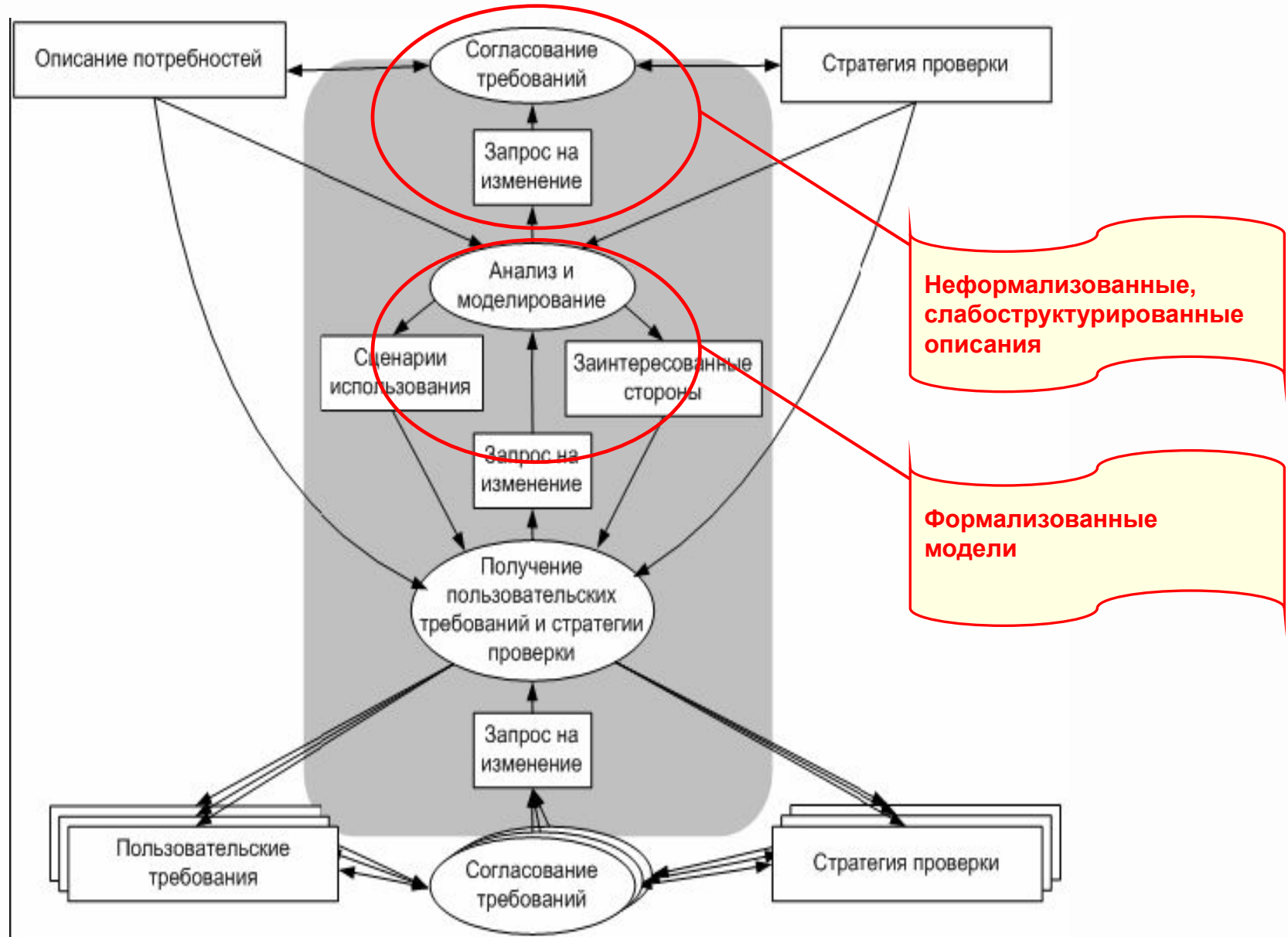
- *избегать хаоса*: формулируя требование, необходимо сконцентрироваться на самом важном; требование не должно быть похоже на роман;
- *избегать «лазеек»*: например, таких выражений, как «если это необходимо», поскольку такие «лазейки» делают требование неоднозначным и зачастую бесполезным;
- *избегать размещения более одного требования в один параграф*: зачастую, наличие в одном параграфе более одного требования легко определить по наличию союза «и»;
- *избегать рассуждений*;
- *избегать «размытых» понятий и слов*: обычно, в основном, часто, нормально, типично;
- *избегать использования неопределенных терминов*: например, удобный в использовании, универсальный, гибкий;
- *избегать принятия желаемого за требуемое*: напр., 100% надежный, приятный для всех пользователей, безопасный, подходящий для всех платформ, не должен никогда ломаться, обрабатывать все неожиданные сбои, быть готов к модернизациям для любых ситуаций, которые могут возникнуть в будущем и т.д.

Требования в области проблем

Возможные вопросы к потенциальному пользователю:

- **Что Вы хотите, чтобы эта система делала?**
- **Зачем Вам нужна система? Какие задачи она должна решать?**
- **Что Вы хотите, чтобы Вы могли делать?**

Процесс разработки пользовательских требований

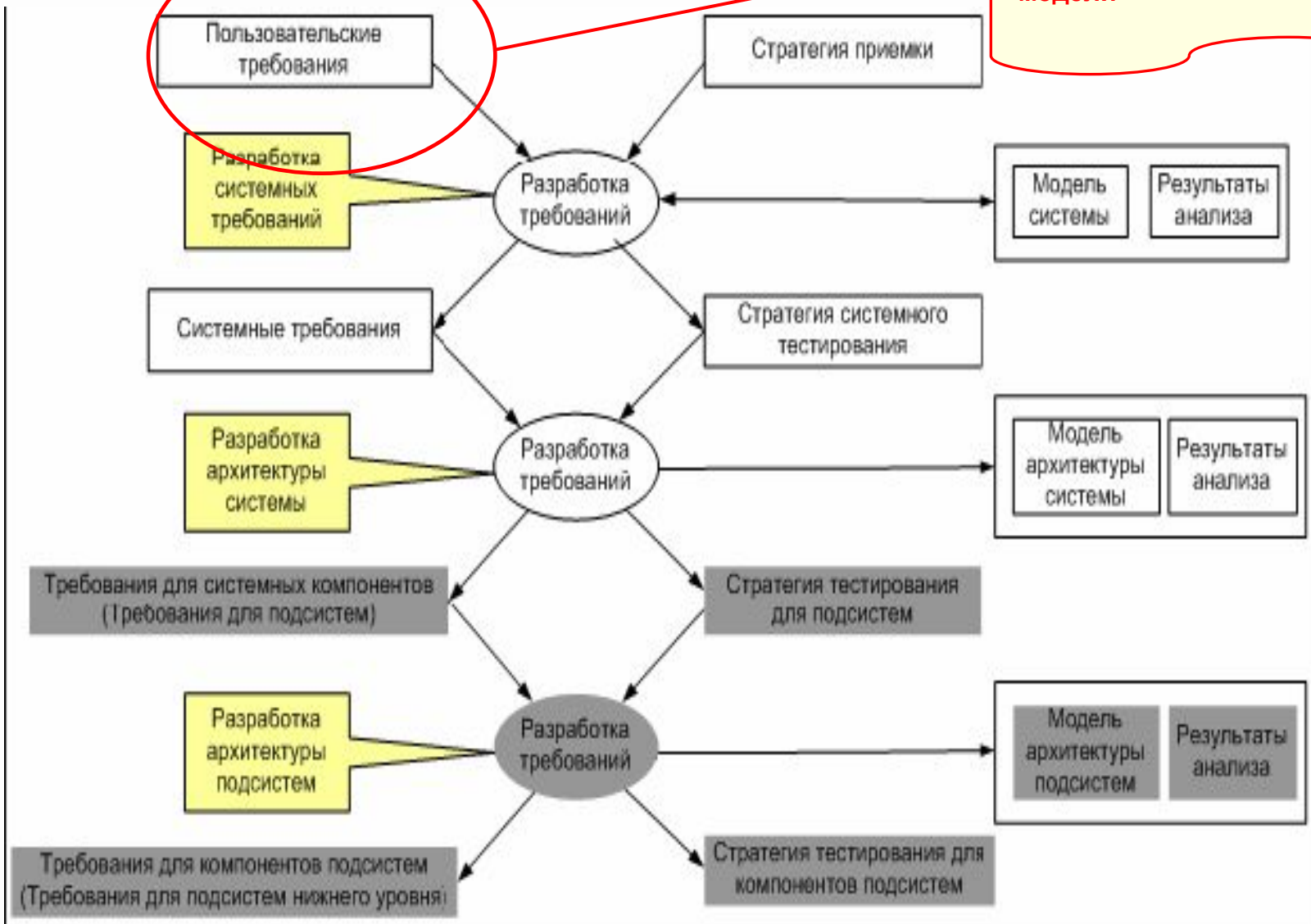


Категории заинтересованных сторон

- Руководство (проекта, использования)
- Инвесторы
- Пользователи
- Обслуживающий персонал
- Утилизаторы
- Обучающий персонал
- Покупатели
- Продавцы (маркетологи)
- Эксперты по эргономике
- Правительство
- Органы стандартизации
- Общественное мнение
- Регулирующие органы

Этапы разработки системных требований

Формализованные модели



Содержание системных моделей

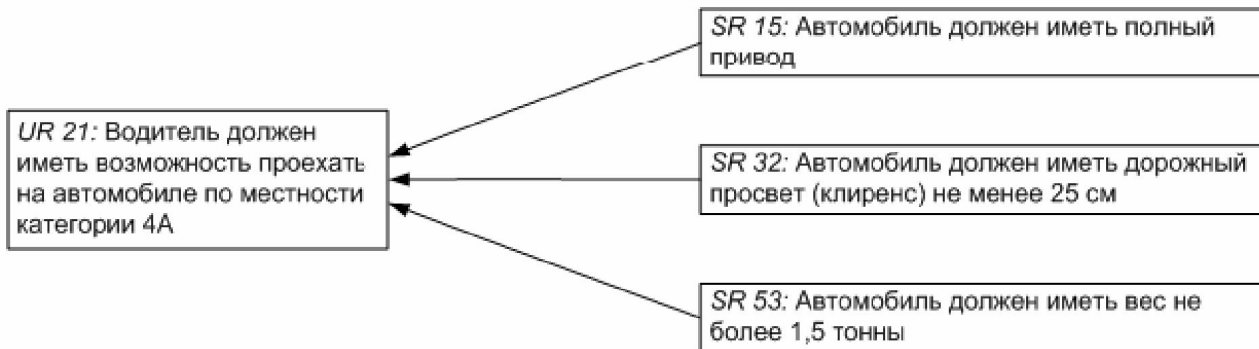
Модель системы

- Внутренняя функциональность (что система должна делать?)
- Функциональность взаимодействия с окружением
- Функциональность взаимодействия с людьми
- Защитная функциональность
- Системные транзакции
- Режимы функционирования

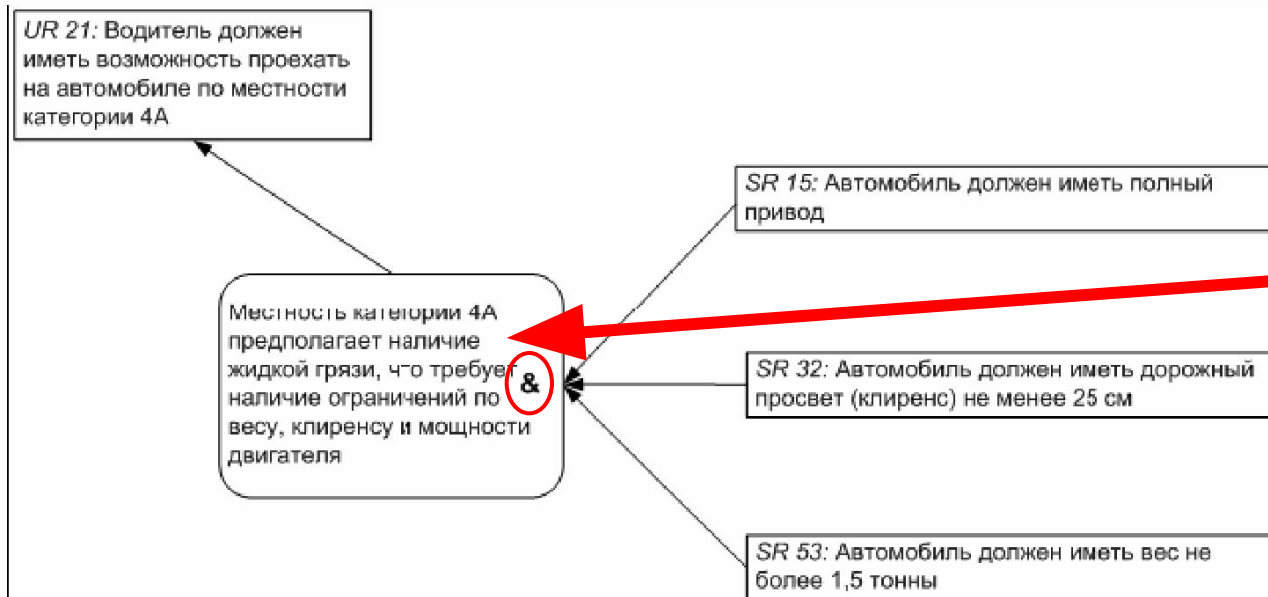
Модель архитектуры– набор компонентов системы, которые, взаимодействуя, порождают системные свойства, определенные в системных требованиях

Расширенные связи

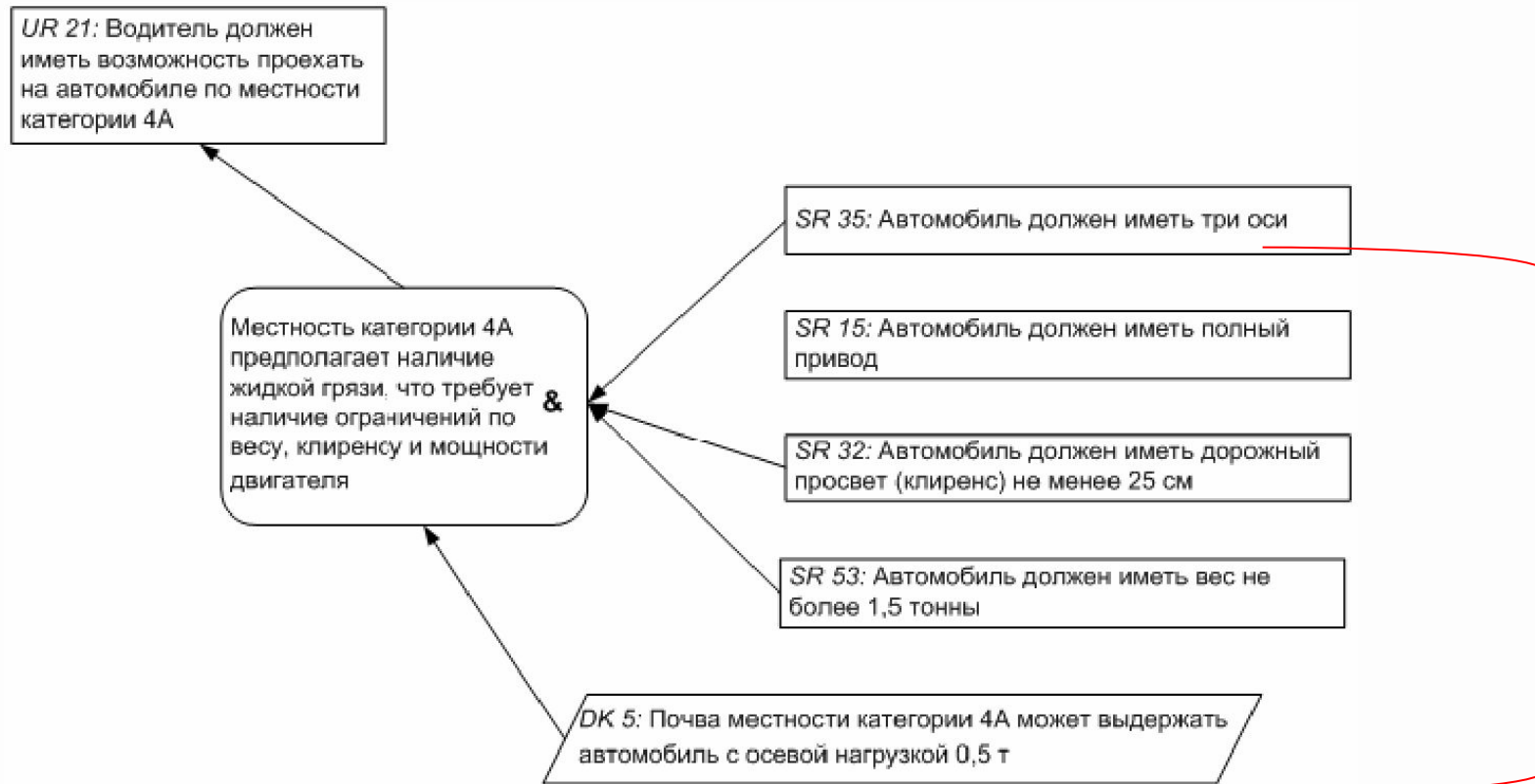
Элементарные
связи



Расширенные
связи с
«аргументом
удовлетворения»



Связь с дополнительными знаниями о предметной области



DK - *Domain Knowledge* – конкретный факт или предположение о предметной области, которое, по своей природе, не является непосредственным ограничением для системы

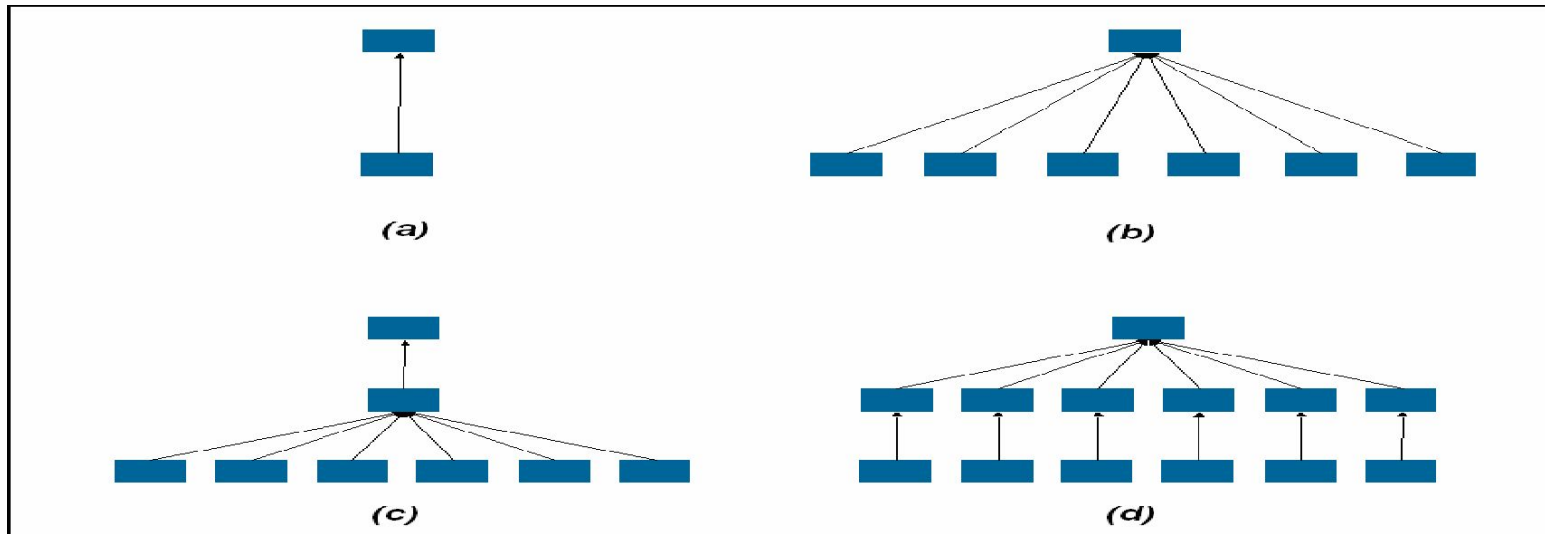
Расширенные связи на многих уровнях



Параметры и метрики связей

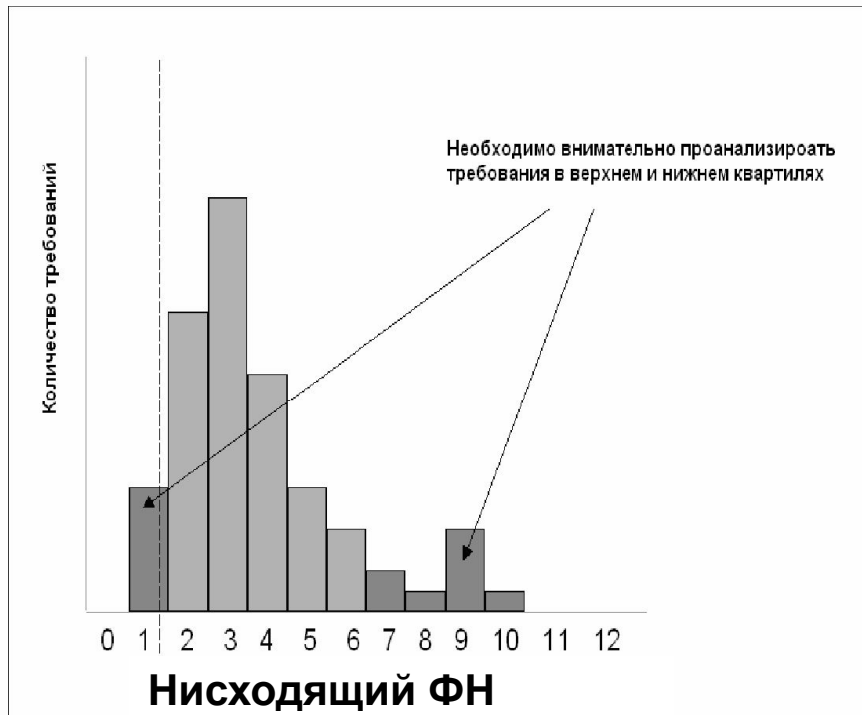
- **Широта** – насколько полно требования данного уровня «охватывают» требования верхнего (нижнего, соседнего) уровня? – *Количественная оценка хода работ*
- **Глубина** – насколько далеко вниз (или вверх) через уровни продолжается данная связь? – *Выявление оснований (источников) требований*
- **Наращение** – насколько широко разрастается связь через уровни? – *Оценка потенциального влияния изменений*

Пример оценок связей

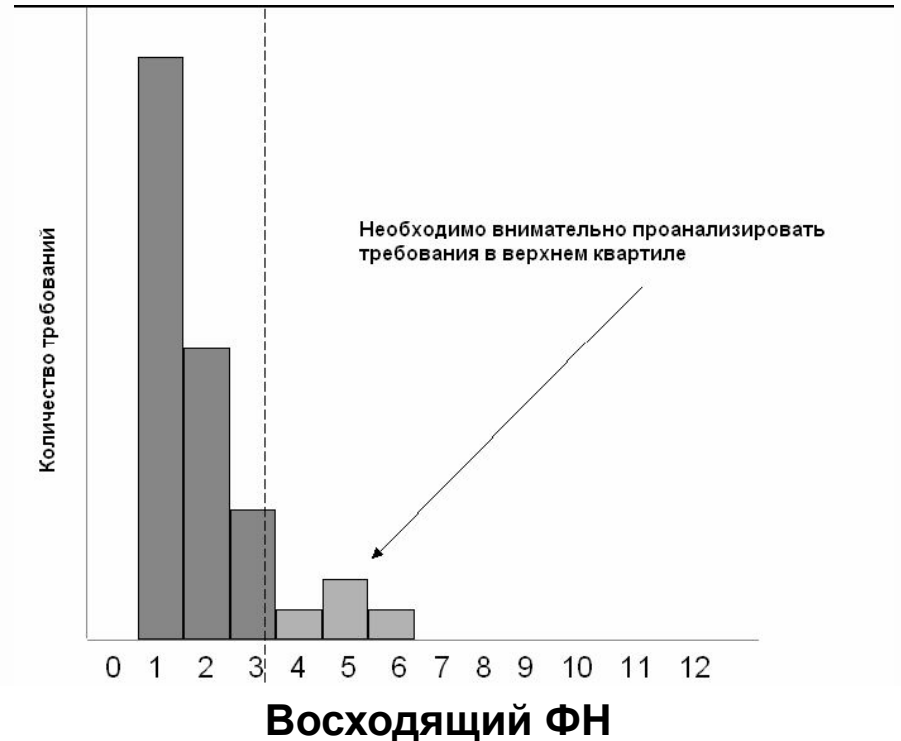


- (b) - требование верхнего уровня значительно сложнее, чем в случае (a);
 - изменения в требовании верхнего уровня будут иметь более существенные последствия, чем в случае (a);
 - требование верхнего уровня плохо сформулировано и требует декомпозиции;
- (c) - требование верхнего уровня слишком абстрактное;
- (d) - требования среднего уровня излишне детализированы.

Анализ частотного распределения значений фактора нарастания



Выявляет плохо сформулированные требования.



Выявляет наиболее критичные требования, от которых многое зависит в системе.

Роли в управлении требованиями

Роль	Функции
Автор	Создает требования и принимает изменения
Издатель	Выпускает и актуализирует документ требований
Цензор	Рецензирует требования и предлагает изменения
Аналитик, разработчик	Анализирует требования и обсуждает изменения