

# Анализ требований к информационным системам

Лекции 1 - 8

# Введение

## Лекция № 1

# Свойства АИС

- Основаны на методологии управления
- Множественный доступ к данным
- Средства корпоративного решения задач
- Дружественный графический интерфейс
- Оперативная обработка информации
- Средства аутентификации и авторизации
- Значительные объёмы данных
- Стандартизация языков и протоколов обмена

# Определение АИС

- *АИС* – это программно-аппаратная система, предназначенная для автоматизации целенаправленной деятельности конечных пользователей, обеспечивающую, в соответствии с заложенной в неё логикой обработки, возможность получения, модификации и хранения информации.

# Определение ИС

- *ИС в широком смысле* – это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

# Классификация по масштабу

Масштаб

Однополь-  
зовательские  
АИС

Групповые  
АИС

Корпоративные  
АИС

# Требования к КИС

- централизация данных в единой базе (в основе – всегда промышленная СУБД),
- близкий к реальному времени режим работы,
- сохранение общей модели управления для предприятий разных отраслей,
- поддержка территориально-распределенных структур,
- работа на широком круге аппаратно-программных платформ и СУБД.

# Классификация по архитектуре



# Классификация по характеру использования информации

Характер  
использования  
информации

Информационно-  
поисковые  
системы

Управляющие  
системы

# Классификация по системе представления данных

Распростра-  
нённые средства  
и  
модели хранения  
данных

«Самодельные»  
форматы

Специализиро-  
ванные  
«дореляционные»  
форматы

Представление  
XML

На основе  
реляционной  
модели

На основе  
объектной,  
объектно-  
реляционной  
моделей

Документо-  
ориентированные  
хранилища

# Классификация по поддерживаемым стандартам управления и технологиям коммуникации

Распространённые  
стандарты  
управления  
и технологии  
коммуникации

MRP

CRP

MRPII

ERP

SCM

CRM

CALS

ERP II

Workflow

OLAP

Project  
Management

# Классификация по степени автоматизации

Степень автоматизации

«Ручные»  
системы

Автоматизи-  
рованные  
системы

Автоматические  
системы

# Требования – понятие и классификация

## Лекция № 2

# Требование к АИС

- **Требование – это условие или возможность, которой должна соответствовать система**
- **Требования – это исходные данные, на основании которых проектируются и создаются автоматизированные информационные системы**

# Определение IEEE

1. Условия или возможности, необходимые пользователю для решения проблем или достижения целей;
2. Условия или возможности, которыми должна обладать система или системные компоненты, чтобы выполнить контракт или удовлетворять стандартам, спецификациям или другим формальным документам;
3. Документированное представление условий или возможностей для пунктов 1 и 2

# Классификация по предмету

Виды  
требований

Требования  
к продукту

Требования  
к проекту

# Классификация по уровню

Бизнес-  
требования

Требования  
пользователей

Требования  
пользователей

Функциональные  
требования

Функциональные  
требования

Функциональные  
требования

Функциональные  
требования

# Классификация 3

Виды  
требований

Системные  
требования

Требования к  
программному  
обеспечению

# Классификация К.Вигерса

Виды  
требований

Функциональные  
требования

Нефункцио-  
нальные  
требования

Характеристики  
продукта

Внешние  
интерфейсы

Атрибуты  
качества

Ограничения

# Модель FURPS+

Виды требований

Атрибуты  
качества

Прочие  
требования

Применимость

Надёжность

Ограничения  
проекта

Требования  
выполнения

Производитель-  
ность

Пригодность к  
эксплуатации

Требования к  
интерфейсу

Физические  
требования

# Документы IEEE

- IEEE 1362 “Concept of Operations Document”.
- IEEE 1233 «Guide for Developing System Requirements Specifications».
- IEEE Standard 830-1998, «IEEE Recommended Practice for Software Requirements Specifications»
- IEEE Standard Glossary of Software Engineering Terminology/IEEE Std 610.12-1990
- IEEE Guide to the Software Engineering Body of Knowledge (1) - SWEBOK®, 2004.

# ГОСТ РФ

- ГОСТ 34.601-90. Информационная технология. Автоматизированные системы. Стадии создания.
- ГОСТ 34.602-89. Информационная технология. Техническое задание на создание автоматизированной системы
- ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению.

# Свойства требований

## Лекция № 3

# Свойства требований

Полнота

Ясность

Коррект-  
ность

Согласо-  
ванность

Верифици-  
руемость

Необходи-  
мость

Осущест-  
вимость

Модифи-  
цируемость

Полезность

Трассируе-  
мость

Упорядо-  
ченность

Наличие  
ко-  
личествен-  
ной

метрики

# Полнота

- Полнота отдельного требования – свойство, означающее, что текст требования не требует дополнительной детализации, то есть в нём предусмотрены все необходимые нюансы, особенности и детали данного требования.
- Полнота системы требований – свойство, означающее, что совокупность артефактов, описывающих требования, исчерпывающим образом описывает всё то, что требуется от разрабатываемой системы.

# Ясность

- **Синонимы:**
  - недвусмысленность,
  - определённость,
  - однозначность спецификаций.
- **Требование обладает свойством ясности, если оно сходным образом воспринимается всеми совладельцами системы**

# Корректность

- Свойство корректности задаёт дихотомию: требование либо корректно, либо нет.
- Корректное требование –
  - непротиворечивое,
  - обеспечивающее требуемую точность,
  - обеспечивающее связь с источниками

# Согласованность

- Вертикальная согласованность:
  - непротиворечивость требованиям родительского уровня иерархии.
- Горизонтальная согласованность:
  - непротиворечивость требованиям своего уровня иерархии.

# Верифицируемость

- Верифицируемость означает пригодность к проверке.
- Основано на:
  - ясности,
  - полноте,
  - трассируемости.

# Необходимость. Полезность

- *Необходимыми* следует считать свойства, без выполнения которых невозможно, либо затруднено выполнение автоматизированных бизнес-функций пользователей;
- *Полезными при эксплуатации* следует считать любые свойства, повышающие применимость продукта.

# Осуществимость (выполнимость)

- Выполнимость требования определяется комбинацией следующих факторов:
  - технической возможностью осуществления;
  - разумным балансом между ценностью (степенью необходимости и полезности) и потребными ресурсами.

# Треугольник компромиссов



# Трассируемость

- Трассируемость требования определяется возможностью отследить связь между ним и другими артефактами информационной системы (документами, моделями, текстами программ и пр.)
- Различают трассируемость в *прямом* и *обратном* направлениях.

# Упорядоченность по важности и стабильности

- *Приоритет требования* представляет количественную оценку степени значимости (важности) требования. Приоритеты требований обычно назначает представитель Заказчика. Разработчик, отталкиваясь от приоритетности требований, управляет процессом реализации информационной системы.
- *Стабильность* требования характеризует прогнозную оценку неизменности требований во времени.

# Наличие количественной метрики

- Количественные метрики играют важную роль в верификации и аттестации информационных систем.
- В первую очередь это относится к нефункциональным требованиям, которые, как правило, должны иметь под собой количественную основу
- Функциональные требования также могут расширяться количественными мерами, например при помощи аспектов применимости

# Каких требований не должно быть

- Спецификация требований не должна содержать деталей проектирования или реализации (кроме известных ограничений).
- Требования должны отвечать на вопрос: «*что* должна делать система», абстрагируясь от того, *как* она это будет делать.

# Процесс анализа требований

## Лекция № 4

# SWEBOK: Requirement Process

- Requirements Elicitation (Извлечение требований)
- Requirements Analysis (Анализ требований в узком смысле)
- Requirements Specification (Специфицирование требований)
- Requirements Validation (Проверка требований)

# RUP: Requirements

- Analyze the Problem (Анализ проблемы)
- Understand Stakeholder Needs (Понимание потребностей совладельцев)
- Define the System (Определение системы)
- Manage the Scope of the System (Управление контекстом системы)
- Refine the System Definition (Уточнение определения системы)

# Работа с требованиями (построение лекционного курса)

- Формирование видения
- Выявление требований
- Классификация и спецификация требований
- Расширенный анализ требований  
(моделирование и прототипирование)
- Документирование требований
- Проверка требований
- *Управление требованиями*
- *Совершенствование процесса работы с требованиями*

# Хорошо проработанные требования позволяют:

- Выработать общее понимание между Заказчиком и Разработчиком
- Определить рамки проекта
- Определить финансовые и временные характеристики проекта
- Обезопасить Заказчика от риска получить продукт, в котором он не сможет работать
- Обезопасить Разработчика от риска попасть в ситуацию неконтролируемого размытия границ

# RUP: Цели потока Requirements

- Добиться одинакового понимания с заказчиком и пользователями о том, что должна делать система
- Дать разработчикам наилучшее понимание требований к системе
- Определить границы системы
- Определить интерфейс пользователя и системы

# Как и кем используются требования?

- *Специалист по АТ* – постановка задачи, определение рамок проекта
- *Представитель заказчика* – постановка задачи, определение рамок проекта, контроль работы исполнителя, приёмка результатов работы
- *Архитектор системы* – разработка архитектуры, проектирование подсистем
- *Программист* – разработка программного кода
- *Тестировщик* – составление тест-плана, тестовых сценариев
- *Менеджер проекта* – планирование и контроль исполнения работ

# Фазы процесса MSF

- Envisioning (выработка концепции)
- Planning (планирование)
- Developing (разработка)
- Stabilizing (стабилизация)
- Deploying (внедрение).

# Envisioning

Ролевой кластер	Фокус
Управление продуктом	Цели проекта; потребности заказчика; документ видение / рамки
Управление программой	Цели дизайна; концепция решения; структура проекта
Разработка	Прототипирование; анализ технологических возможностей; анализ осуществимости
Удовлетворение потребителя	Необходимые эксплуатационные характеристики решения и их влияние на его разработку.
Тестирование	Стратегии тестирования; критерии приемлемости, их влияние на разработку решения.
Управление выпуском	Требования внедрения и их влияние на разработку решения; требования сопровождения.

# Planning

Ролевой кластер	Фокус
Управление продуктом	Анализ бизнес-требований
Управление программой	Функциональная спецификация
Удовлетворение потребителя	Сценарии/примеры использования, пользовательские требования, требования локализации и общедоступности
Тестирование	Требования тестирования
Управление выпуском	Эксплуатационные требования

# Developing

Ролевой кластер	Фокус
Управление продуктом	Ожидания заказчика
Управление программой	Управление функциональной спецификацией

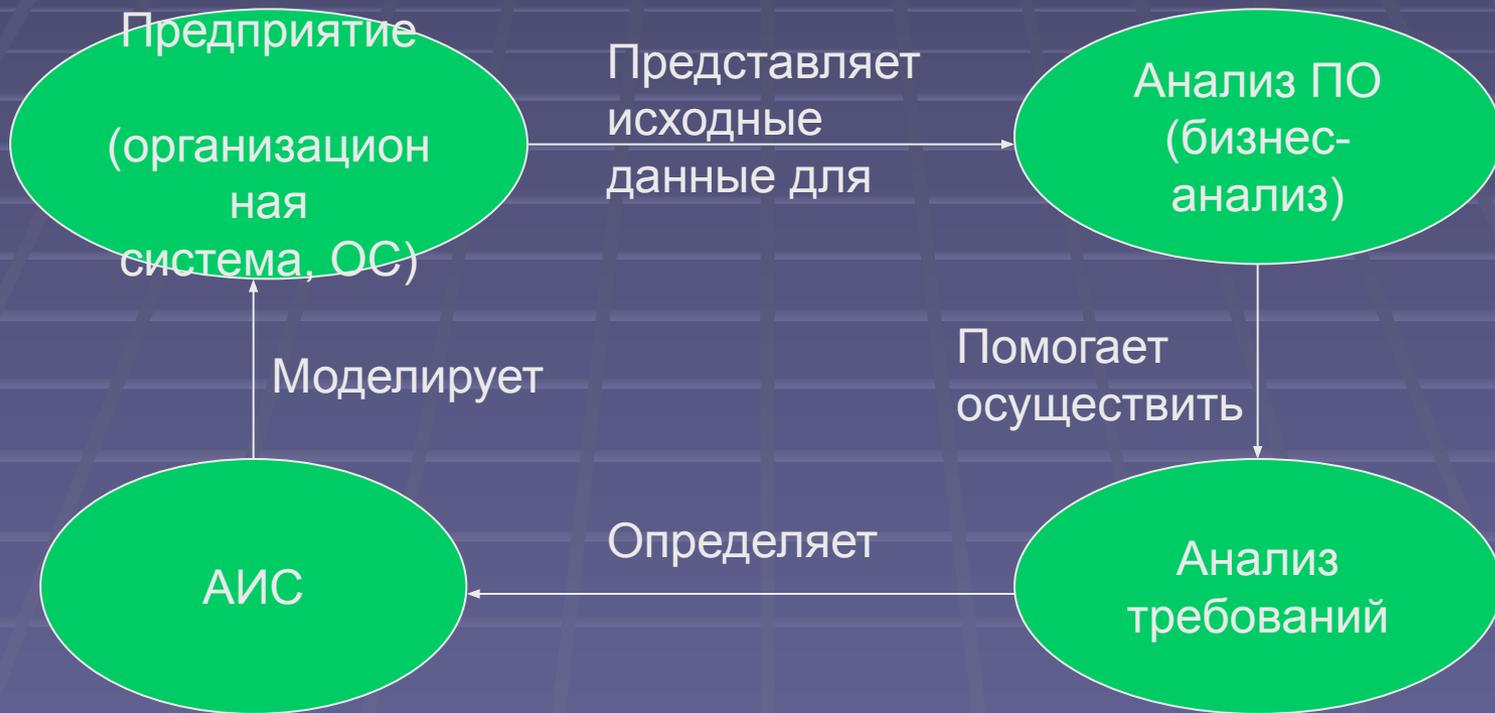
# Deploying

<b>Ролевой кластер</b>	<b>Фокус</b>
Управление продуктом	Получение отзывов и оценок заказчика; акт о приеме выполненной работы
Управление программой	Сопоставление рамок проекта с поставленным решением; управление стабилизацией

# Контекст задачи анализа требований

## Лекция № 5

# Анализ требований и бизнес-анализ





# Методологии бизнес-анализа

Категории  
моделей

модели для  
анализа и  
улучшения  
организационной  
системы

модели общего  
назначения

модели,  
предназначенные  
для  
использования  
при  
автоматизации

# Модели для улучшения ОС

SWOT

VCM

BPR

BSC

CPI/TQM/  
ISO9000

# Модели общего назначения

IDEF1

IDEF3

IDEF5

SADT

DFD

# Модели для автоматизации

ISA

BSP

ARIS

RUP

# Модели организации (ARIS)

Организа-  
ционная

Функцио-  
нальная

Входов/  
выходов

Информаци-  
онная

Процессов  
управления

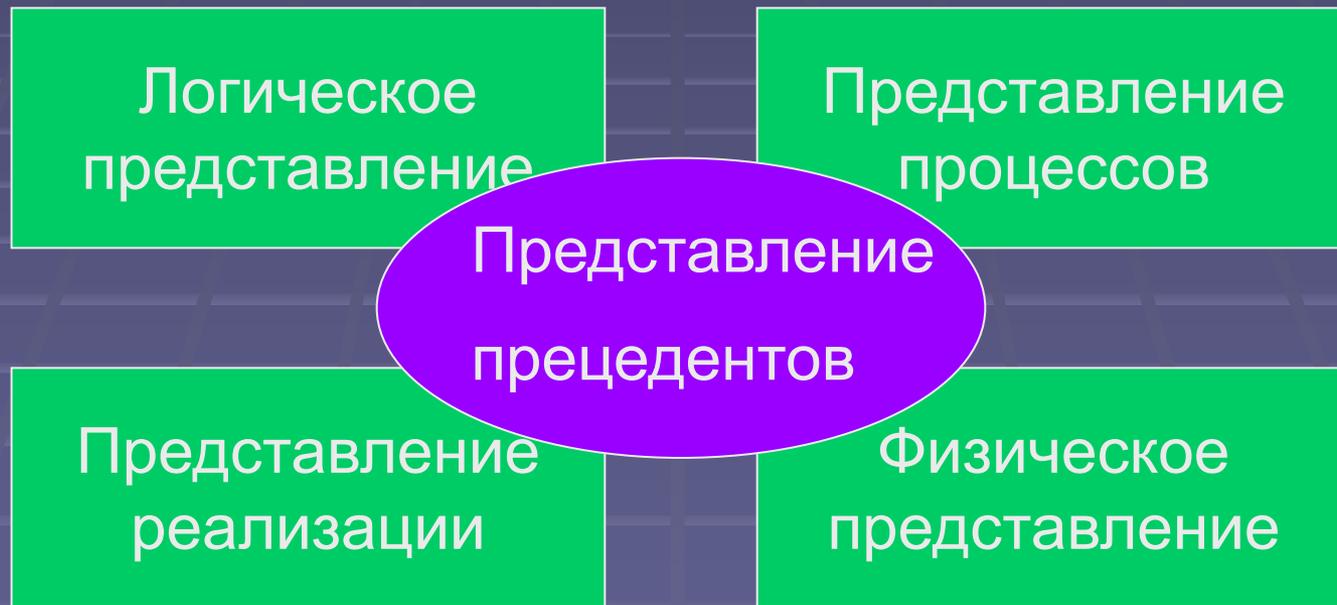
Целей  
организации

Средств  
производства

Человеческих  
ресурсов

Территори-  
альная

# Архитектура программной системы (RUP)



# Зависимости между АТ и др. потоками работ (RUP)



# Выявление требований

## Лекция № 6

# Источники требований



# Стратегии выявления требований

## Стратегии выявления требований

Интервью

Анкетирование

Наблюдение

Самостоятельное описание требований

Совместные семинары

Прототипирование

# Интервью

```
graph TD; A[Интервью] --- B[Организация интервью]; B --- C[Подготовка]; B --- D[Проведение]; B --- E[Завершение];
```

Организация  
интервью

Подготовка

Проведение

Завершение

# Анкетирование

- Недостатки анкетирования: респонденты часто бывают неспособны, либо слабо мотивированы в том, чтобы хорошо и информативно заполнить анкету. Высок риск получить неполную или вовсе ложную информацию.
- Преимущество – в том, что подготовка и анализ анкет требуют небольшой ресурс.

# Совместные семинары

Совместные  
семинары

Мозговой  
штурм

JAD-метод

Разъясняющие  
встречи

# Мозговой штурм

- Правила *мозгового штурма* предполагают полную раскрепощённость и свободу мнений, даже самых вычурных и на первый взгляд «бредовых». Первое правило мозгового штурма – «полный запрет на любую критику». Всякое высказанное мнение представляет ценность, а полное отсутствие запретов позволяет полноценным образом подключить творческую фантазию.
- Затем, на втором этапе, все высказанные мнения тщательным образом обсуждаются, заведомо неприемлемые варианты отсеиваются, формируются коллективные предложения.

# Участники JAD-совещания

- **Ведущий** – специалист в области межличностных коммуникаций. Должен ориентироваться в предметной области, но не обязательно хорошо ориентироваться в проблемах IT.
- **Секретарь** – стенографист встречи. Фиксирует её результаты на компьютере. Возможно применение CASE-средств.
- **Заказчики** – пользователи или руководители, основные участники, формирующие, обсуждающие требования и принимающие решения.
- **Разработчики** – аналитики и другие участники проектной команды. Работают в большей части в пассивном режиме с целью наилучшего понимания проблемной области.

# Разъясняющие встречи

*“Разъясняющие встречи”* или *“запланированный мозговой штурм”* – термин, пришедший из общей практики менеджмента и базирующийся на идеях сотрудничества заинтересованных лиц для совместного анализа путей решения проблем, определения и предупреждения рисков и т.п.

# Выявление через прототипы. Принципы RAD-метода

- Эволюционное прототипирование;
- CASE-средства, как основной инструмент, включая возможности прямого и обратного проектирования и автоматической генерации кода;
- Высококвалифицированные специалисты, хорошо владеющие развитыми инструментальными средствами;
- Интерактивный JAD-метод, в котором общение совмещается с разработкой в режиме online;
- Жёсткие временные рамки, как противовождение от «расползания границ» проекта: если команда не укладывается в срок – функционал сужается.

# Формирование видения

## Лекция № 7

# Видение / Рамки

Концепция

Видение

Образ

Границы

Рамки

Контекст

# ГОСТ 34.601-90

Работы этапа разработки концепции

```
graph TD; A[Работы этапа разработки концепции] --- B[Изучение объекта]; A --- C[Проведение НИР]; A --- D[Разработка вариантов концепции]; A --- E[Оформление отчёта о проделанной работе];
```

Изучение объекта

Проведение НИР

Разработка вариантов концепции

Оформление отчёта о  
проделанной работе

# RUP

## Формирование документа «Vision»

Формулировка проблем

Идентификация совладельцев

Определение границ системы

Идентификация ограничений

Формулировка постановки задач

Определение возможностей системы

Определение результатов

# RUP – шаблон для формулировки проблемы

Проблема	(описание проблемы)
Затрагивает	(совладельцы, затрагиваемые проблемой).
Ее следствием является	(каково влияние проблемы).
Успешное решение	(список некоторых ключевых преимуществ от успешного решения).

# Идентификация совладельцев. Определение границ системы.

- Идентификация *совладельцев* предполагает поиск и фиксацию интересантов проекта – представителей Заказчика и Исполнителя, инвесторов, внешних экспертов и пр.
- Определение *границ системы* представляет собой нетривиальный процесс. Для этого используют контекстные диаграммы.
- RUP в поиске границ предлагает отталкиваться от акторов и вариантов использования.

# Классификация ограничений



# Шаблон документа «Vision» RUP

1. Введение
2. Позиционирование
3. Описания совладельцев и пользователей
4. Краткий обзор изделия
5. Возможности продукта
6. Ограничения
7. Показатели качества
8. Старшинство и приоритеты
9. Другие требования к изделию
10. Требования к документации
11. Приложение.

# Vision / Scope (MSF)

Согласно белой книге MSF, на фазе выработки концепции (envisioning phase) закладывается одна из фундаментальных основ успеха проекта – **создание и сплочение проектной группы** на основе выработки единого *видения*.

Проектная группа должна **четко представить себе**, что она хочет сделать для заказчика и сформулировать свою цель таким образом, чтобы максимально мотивировать как заказчика, так и саму проектную команду.

Выработка высокоуровневого взгляда на цели и условия проекта может рассматриваться как ранняя форма планирования.

# MSF – envisioning phase

- **Основными задачами** фазы выработки концепции являются создание ядра проектной группы и подготовка vision/scope document.
- *Видение (vision)* – это ничем не ограничиваемое представление о том, каким должно быть решение
- *Рамки (scope)* же дают четкие границы того, что из предложенного этим видением будет реализовано в условиях существующих проектных ограничений.

# Шаблон документа «Vision/Scope» MSF

1. Бизнес-преимущества
  - 1.1. Описание преимуществ
  - 1.2. Формулировка видения
  - 1.3. Анализ выгод
2. Концепция решения
  - 2.1. Цели, задачи, предположения и ограничения
  - 2.2. Анализ применимости
  - 2.3. Требования
3. Рамки
  - 3.1. Список характеристик/функций
  - 3.2. Вне рамок
  - 3.3. Стратегия подготовки релизов
  - 3.4. Критерии применимости
  - 3.5. Эксплуатационные критерии
4. Стратегии проектирования решения
  - 4.1. Стратегия проектирования архитектуры
  - 4.2. Стратегия технического проектирования

# Классификация и специфицирование требований

## Лекция № 8

# Требования совладельцев

- Результатом выявления требований, является реестр требований.
- Требования совладельцев обычно оформляются в простой письменной форме, без какой-либо особой регламентации.
- Пример оформления требования к программе электронной почты – «Система должна позволять набирать текст сообщения с возможностью форматирования текста и вставки смайликов».

# Акторы и варианты использования

- Самым популярным и весьма эффективным способом повышения информативности требований является оформление их в виде вариантов использования, предложенный И. Якобсоном.
- Прежде, чем приступить собственно к специфицированию требований в форме вариантов использования, RUP рекомендует выявить реестр акторов (actors) и вариантов использования (use cases).

# Актор

- Актор – это некто или нечто, обладающее активностью по отношению к программной системе.
- Помимо пользователя в качестве актора может рассматриваться другая программная система, аппаратное устройство, в ряде случаев – активная компонента самой системы.

# Вариант использования

- Вариант использования (прецедент) в первом приближении можно рассматривать, просто, как функцию, реализуемую системой.
- вариант использования должен быть полезен конкретному *актору*.
- вариант использования должен позволять получать ему конкретные законченные результаты

# Глоссарий

- Служит основой для единообразного понимания описаний требований Заказчиком и Разработчиком.
- Является отправной точкой для построения более развёрнутых моделей проблемной области, которые, на стадии реализации информационной системы, ложатся в основу объектной модели (для объектно-ориентированных приложений) и модели данных (для генерации схемы базы данных).
- Оформляется, как текст, состоящий из абзацев, каждый из которых определяет значение одного из терминов проблемной области. Термин обычно выделяют полужирным кеглем.
- Иногда проблемную область целесообразно сегментировать на ряд «подобластей» (subject areas). Тогда каждой из них в глоссарии выделяется отдельный параграф.

# Спецификации прецедента

Свободный  
формат

Полный  
формат

Таблица  
в три  
колонки

Язык  
описания  
алгоритма

Стиль  
RUP

Таблица  
в две  
колонки

Псевдокод

Диаграмма  
активности  
UML

Другие  
графические  
модели

# Полный формат

1. Название
2. Контекст использования
3. Область действия
4. Уровень
5. Основное действующее лицо
6. Участники и интересы
7. Предусловие
8. Минимальные гарантии
9. Гарантии успеха
10. Триггер
11. Основной сценарий
12. Расширения
13. Список изменений в технологии и данных
14. Вспомогательная информация

# Таблица в 2 колонки

<i>Актор</i>	<i>Действие</i>
Пользователь	Формирует запрос на поиск заказов
Система	Отображает список заказов
Пользователь	Выбирает требуемый заказ
Система	Показывает подробную информацию по заказу

# Таблица в 3 колонки

№ шага	Пользователь	Система
1	Делает запрос на поиск заказов	Отображает список заказов
2	Выбирает требуемый заказ	Показывает подробную информацию по заказу

# Формат RUP

1. Наименования и краткое описание
2. Поток событий
  - 2.1. Основной поток
  - 2.2. Альтернативные потоки
3. Специальные требования
4. Предусловия
5. Постусловия
6. Точки расширения

# Спецификация нефункциональных требований

- Описание нефункциональных требований обычно осуществляется в форме, близкой к свободному формату описания варианта использования.
- RUP рекомендует концентрировать нефункциональные требования в документе, описывающем вариант использования во всех случаях, когда это возможно.
- В случае, если нефункциональные требования носят общий характер, они выносятся в документ «Дополнительная спецификация».

# Атрибуты требований

- Атрибуты требований описываются матрицей атрибутов требований, где для каждого типа требований перечисляются требования по одной оси и атрибуты требований этого типа по другой.
- Для каждого требования указываются значения его соответствующих атрибутов.
- Примеры атрибутов: статус во времени, приоритет, важность, риск, № итерации (этапа) в плане.