

# **ТЕМА 2.**

# **Технологии**

# **проектирования**

# **информационных систем**

Лекция 7.

Структурные модели предметной области.

Методы сбора требований пользователей.

# Требования к моделям предметной области

- **Формализованность**, обеспечивающая однозначное описание структуры предметной области;
- **Понятность для заказчиков и разработчиков**, основанная на применении графических средств отображения модели;
- **Реализуемость**, подразумевающая наличие средств физической реализации модели предметной области в ИС;
- **Обеспечение оценки эффективности** реализации модели предметной области на основе определенных методов и вычисляемых показателей.

# Структурный аспект моделирования предметной области

- **Объектная структура** отражает состав взаимодействующих в процессах материальных и информационных объектов предметной области;
- **Функциональная структура** отражает взаимосвязь функций по преобразованию объектов в процессах;
- **Структура управления** отражает события и бизнес-правила, которые воздействуют на выполнение процессов;
- **Организационная структура** отражает взаимодействие организационных единиц предприятия и персонала в процессах;
- **Техническая структура** описывает топологию расположения и способы коммуникации комплекса технических средств.

# Оценочный аспект моделирования предметной области

- Оценочный аспект моделирования предметной области связан с показателями эффективности автоматизируемых процессов:
  - время решения задач;
  - стоимостные затраты на обработку данных;
  - экономические показатели эффективности, (объемы производства, производительность труда, оборачиваемость капитала, рентабельность и т.д.).
- Для отображения оценочного аспекта используются:
  - статические методы функционально-стоимостного анализа (АВС);
  - динамические методы имитационного моделирования.

# Уровни проектирования

- **Внешний** уровень проектирования – этап выяснения взаимодействия системы с внешней средой.
  - **Что и зачем** будет делать система?
  - **Почему** она должна действовать подобным образом?
- **Концептуальный** уровень проектирования – этап определения характера взаимодействия основных компонентов системы.
  - **Как** должна функционировать система?
  - **Кто, где, когда** будет выполнять необходимые операции и процедуры?
- **Внутренний** уровень проектирования – этап определения способов реализации функций системы.
  - **Какими способами** и средствами система будет выполнять свои функции?
  - С помощью каких программно-технических средств реализуются требования к системе?

# Модель объектной структуры

- **Объектная структура** отражает состав взаимодействующих в процессах материальных и информационных объектов предметной области.

Внешний уровень	Концептуальный уровень	Внутренний уровень
Определение основных классов материальных и информационных объектов	Уточнение состава объектов, определение их атрибутов и взаимосвязей	Отображение объектов в файлах БД. Динамические объекты – единицы переменной информации (документы); Статические – единицы условно-постоянной информации (справочники)

# Модель функциональной структуры

- Функциональная структура отражает взаимосвязь функций по преобразованию объектов в бизнес-процессах.

Внешний уровень	Концептуальный уровень	Внутренний уровень
Список основных бизнес-процессов	Иерархия взаимосвязанных функций	Иерархия структуры программных модулей

# Модель структуры управления

- Модель структуры управления отражает события и бизнес-правила, которые воздействуют на выполнение процессов.
- События вызывают выполнение функций, которые изменяют состояния объектов и формируют новые события.

## Событие



**Информационный аспект**  
(сообщение)

**Процедурный аспект**  
(вызов функции)

<b>Внешний уровень</b>	<b>Концептуальный уровень</b>	<b>Внутренний уровень</b>
<ul style="list-style-type: none"><li>■ Список внешних событий;</li><li>■ Список целевых установок</li></ul>	Бизнес-правила, определяющие условия вызова функций при возникновении событий	Триггеры (вызовы программных модулей)



# Модель организационной структуры

- Организационная структура отражает взаимодействие организационных единиц предприятия при выполнении бизнес-процессов.

<b>Внешний уровень</b>	<b>Концептуальный уровень</b>	<b>Внутренний уровень</b>
Иерархия организационных единиц	Организационно-штатная структура должностей и ролей для каждого подразделения.	Права доступа персонала к функциям ИС

# Модель технической структуры

- Техническая структура отражает топологию расположения и способы коммуникации технических средств.

<b>Внешний уровень</b>	<b>Концептуальный уровень</b>	<b>Внутренний уровень</b>
Перечень типов ТС обработки данных и их размещение по подразделениям	Способы взаимодействия между ТС	Модель архитектуры вычислительной сети

# Взаимосвязь областей проектирования и структурных моделей предметной области

Области проектирования	Компоненты ИС	Структурные модели ПрО
Объекты данных	Данные	Объектная структура
Программы обработки данных	Функциональные программные модули	Функциональная структура
	Управляющие программные модули	Структура управления
	Программные модули интерфейсов пользователей	Организационная структура
Среда (технология) обработки данных	Комплекс технических средств	Техническая структура

# Понятие требования

- *Требования* – это исходные данные, на основании которых проектируются и создаются ИС.
- *Требование* – условие или особенность, которой должна удовлетворять ИС:
  - функциональность, необходимая заказчику или пользователю для разрешения проблем или получения прибыли;
  - функциональность, которая должна быть реализована в системе в соответствии с официальным документом;
  - ограничение, наложенное заинтересованным лицом.

# Источники требований

- Федеральное и муниципальное отраслевое законодательство (конституция, законы, распоряжения)
- Нормативное обеспечение организации (регламенты, положения, уставы, приказы)
- Текущая организация деятельности объекта автоматизации
- Модели деятельности (диаграммы бизнес-процессов)
- Журналы использования существующих программно-аппаратных систем
- Конкурирующие программные продукты
- Заинтересованные лица

# Заинтересованные лица



# Использование требований при разработке ИС

<b>Заинтересованное лицо</b>	<b>Область использования требований</b>
<b>Системный аналитик</b>	Постановка задачи, определение рамок проекта
<b>Представитель заказчика</b>	Постановка задачи, определение рамок проекта, контроль работы исполнителей, приемка результатов работы
<b>Проектировщик</b>	Разработка архитектуры, проектирование подсистем
<b>Программист</b>	Разработка программного кода
<b>Тестировщик</b>	Составление планов тестирования, тестовых сценариев
<b>Менеджер проекта</b>	Планирование и контроль исполнения работ

# Методы сбора требований

- Интервью
- Анкетирование
- Наблюдение
- Самостоятельное описание требований
- Совместные семинары
- Прототипирование



# Интервью

**1. Подготовка** – планирование процесса опроса и выработка стратегии управления этим процессом.

- выбор нужного собеседника;
- договоренность о встрече;
- формирование предварительной программы встречи;
- изучение сопутствующей информации;
- согласование плана опроса с группой проектирования.

**2. Проведение опроса.**

**3. Завершение.**

# Интервью

1. *Подготовка* – планирование процесса опроса и выработка стратегии управления этим процессом.
2. *Проведение опроса.*
3. *Завершение.* Опрос нужно завершать, если:
  - получен достаточно большой объем информации;
  - поступает большой объем неподходящей информации;
  - информация перестает усваиваться;
  - эксперт начинает уставать;
  - с экспертом возник конфликт.

# Анкетирование

- **Преимущество:** наименее затратный способ извлечения информации.
- **Недостаток:** наименее эффективный способ сбора данных.
- В анкетах могут использоваться следующие *виды вопросов:*
  - *Многоальтернативные вопросы.*
  - *Рейтинговые вопросы.*
  - *Вопросы с ранжированием.*

# Наблюдение

- Применяется для непосредственного сбора сведений о параметрах, признаках и объектах в соответствующей предметной области.
- Различают *пассивное и активное* наблюдение.
- **Достоинство**: сбор информации, которую невозможно получить путем опроса или изучения документации.
- **Недостаток**: наблюдатель «вносит помехи» в результаты измерений.

# Самостоятельное описание требований

- Используется при наличии:
  - хорошо структурированной документации, описывающей устоявшиеся в организации бизнес-процессы;
  - большого опыта разработки ИС в схожих предметных областях.
- **Достоинство:** предварительное формирование требований происходит в удобном для аналитика режиме.
- **Недостаток:** возможность пропуска важной информации, связанной с выполнением бизнес-процессов в реальной жизни и не вошедшей в документы.

# Совместные семинары

- *Групповое обсуждение* по методу «мозгового штурма» проводится с целью обобщения и обсуждения важных для решения проблем вопросов.
- **Недостаток:** одна из наиболее затратных стратегий сбора данных.
- **Достоинства:** быстрота принятия решений, снижение количества ошибок, выработка нетривиальных идей.

# Прототипирование

- Прототипирование является ключевым компонентом технологии быстрой разработки приложений (*RAD – Rapid Application Development*).
- RAD базируется на следующих принципах:
  - эволюционное прототипирование;
  - использование CASE-средств, обладающих возможностями прямого и обратного проектирования и автоматической генерации кода;
  - высококвалифицированные специалисты;
  - совмещение живого общения с разработкой в режиме on-line;
  - жесткие временные рамки.

# Классификация требований





# Уровни требований



<b>Уровни модели</b>	<b>Уровни требований</b>	<b>Уровни схемы Захмана</b>	<b>Этапы жизненного цикла</b>
Внешний	Бизнес-требования	1-2 (контекст, бизнес-модель)	Анализ требований
Концептуальный	Требования пользователей	3 (системная модель)	Техническое проектирование
Внутренний	Функциональные системные требования	4-5 (технологическая модель)	Рабочее проектирование

# Бизнес- требования

## 1. Назначение:

- Формулировка цели проектирования ИС

## 2. Где описываются:

- Концепция системы (границы и содержание проекта)

## 3. Пример:

- система должна сократить срок оборачиваемости обрабатываемых на предприятии заказов в три раза.

# Требования пользователей

## 1. Назначение:

- определяют набор пользовательских задач, которые должна решать ИС, а также способы (сценарии) их решения в системе.

## 2. Где описываются:

- Диаграммы вариантов использования, сценарии взаимодействия, функциональные модели в различных нотациях

## 3. Пример:

- система должна представлять диалоговые средства для ввода исчерпывающей информации о заказе, последующей фиксации информации в базе данных и маршрутизации информации о заказе к сотруднику, отвечающему за его планирование и исполнение.

# Функциональные системные требования

## 1. Назначение:

- определяют способы реализации ИС.

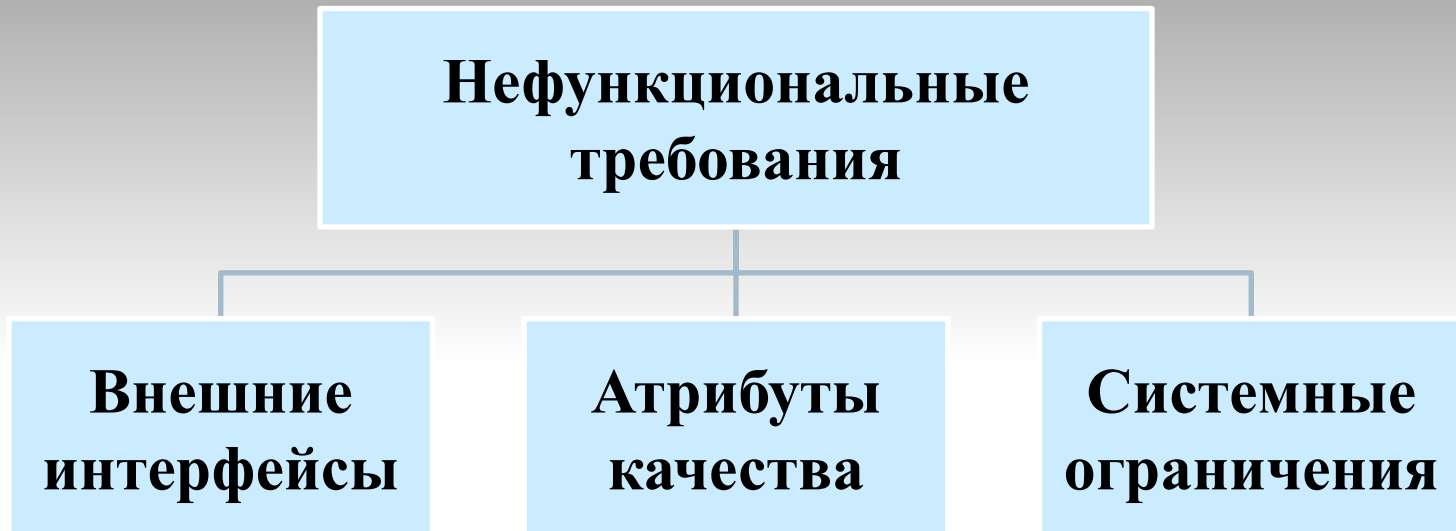
## 2. Где описываются:

- системные спецификации (system requirement specification, SRS)

## 3. Пример:

- заказ может быть создан, отредактирован, удален и перемещен с участка на участок.

- Нефункциональные требования – это требования к *характеру* поведения системы



- Интерфейс пользователя,
- Аппаратные интерфейсы,
- Программные интерфейсы,
- Коммуникационные интерфейсы

- Удобство использования*
- Надежность*
- Производительность*
- Эксплуатационная пригодность (способность к сопровождению)*

Требования, выдвигаемые ИС к среде своего функционирования (объем требуемой памяти, требования к выбору операционной системы)

# Диаграмма требований

EA - Enterprise Architect

FILE EDIT VIEW PROJECT PACKAGE DIAGRAM ELEMENT TOOLS ANALYZER EXTENSIONS WINDOW HELP

Model > Модели > Модель требований > Функциональные требования

Project Browser

- Модель анализа
- Модель анализа (по шаблону)
- Модель вариантов использования
- Модель вариантов использования (по шаблону)
- Модель требований
  - Модель требований
  - Функциональные требования
    - Функциональные требования
    - Бизнес-требования
    - Пользовательские требования
    - Системные требования
  - «RequirementsChecklist» Checklist
- Нефункциональные требования
  - Нефункциональные требования
  - Бизнес-правила
  - Ограничения
  - Требования к безопасности и надёжности
  - Требования к дизайну и юзабилити
  - Требования к документированию
  - Требования к показателям назначения

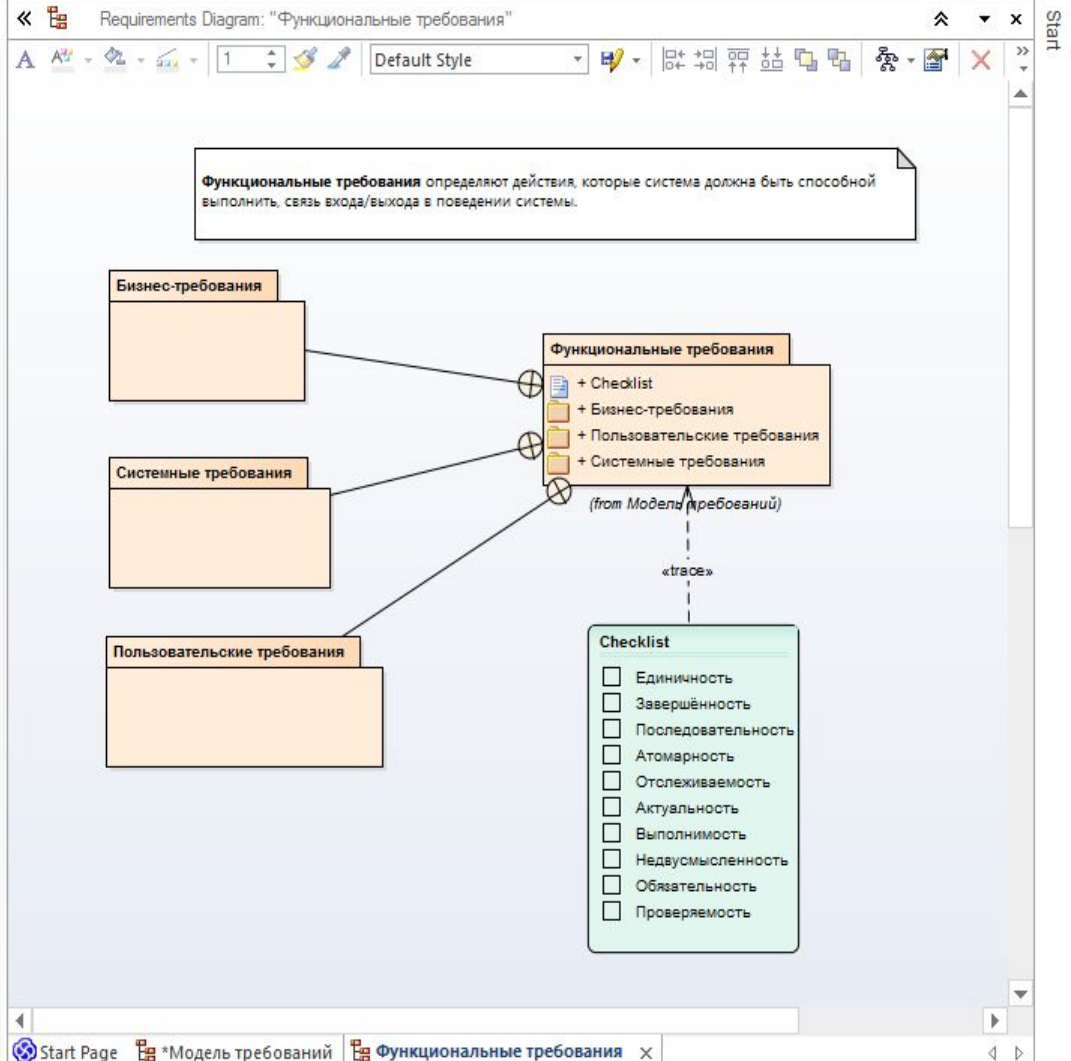
Project Browser Resources

Notes

B I U A | | | x² x₂ |

Toolbox

- Requirements
  - Package
  - Requirement
  - Feature
  - Risk
  - Object
- Extended Requirements
  - Functional
  - Business
  - User
  - Non-Functional
  - Architectural
  - Implementation
  - Regulatory
  - Security
  - Requirements Checklist
- Requirements Relationships
- Patterns
  - Functional Requirements
  - Non-Functional Requirement
- Common
- Artifacts



Properties Notes

Ready

Start Page \*Модель требований Функциональные требования

CAP NUM SCRL CLOUD

# Особенности нефункциональных требований

- Заказчики часто забывают про эти требования и не предоставляют их, пока не будут заданы соответствующие вопросы.
- Заказчики обычно не в курсе стоимости улучшения определенных возможностей.
- У нетехнических пользователей часто возникают проблемы с пониманием смысла некоторых технических требований.
- Некоторые требования являются сложными в измерении, например: «Система должна быть простой для обучения».



# Категории нефункциональных требований

1. Основные:
  1. Удобство использования
  2. Надежность
  3. Производительность
  4. Эксплуатационная пригодность
2. Дополнительные:
  1. Ограничение на дизайн
  2. Требования реализации
  3. Требования интерфейса
  4. Требования аппаратного обеспечения
  5. Требования документации
  6. Требования лицензий и юридических норм

# Требование «Удобство использования»

<b>Подкатегория</b>	<b>Пример</b>
<b>Доступность</b>	Функциональность бронирования билета на самолет должна быть доступна с домашней страницы.
<b>Эстетичность</b>	Поля ввода на одной странице должны быть выровнены вертикально.
<b>Соответствие интерфейсу пользователя</b>	Пользовательский интерфейс должен соответствовать стандарту IBM
<b>Эргономичность</b>	При открытии диалогового окна курсор должен быть на первом поле ввода
<b>Легкость в использовании</b>	Среднее время процедуры бронирования должно быть не более двух минут.

# Требование «Надежность»

<b>Подкатегория</b>	<b>Пример</b>
<b>Работоспособность</b>	Система должна быть доступна 99,93% времени.
<b>Прочность</b>	На каждый неверный ввод данных пользователем система должна реагировать соответствующим сообщением об ошибке
<b>Точность</b>	Денежные расчеты должны выполняться и храниться с точностью до двух десятых
<b>Восстанавливаемость</b>	После восстановления системы из состояния неработоспособности обработка данных должна производиться в том же режиме, что и до сбоя.
<b>Корректность</b>	После выпуска релиза система может иметь не более чем 20 незначительных ошибок.

# Требование «Производительность»

<b>Подкатегория</b>	<b>Пример</b>
<b>Скорость обработки данных</b>	Система должна обрабатывать 1000 процедур бронирования билетов в минуту.
<b>Время ответа</b>	Среднее время отображения списка полетов должно быть не более 10 секунд.
<b>Время восстановления</b>	Среднее время восстановления должно быть менее часа.
<b>Время загрузки /выхода</b>	Система должна быть работоспособной в течение одной минуты от момента загрузки.
<b>Емкость</b>	Система должна обслуживать 5000 пользователей одновременно.
<b>Использование ресурсов</b>	Система должна хранить в БД не более 1 млн. транзакций. При превышении лимита старые транзакции архивируются.

# Требование «Эксплуатационная пригодность»

- Тестируемость
- Приспособляемость
- Совместимость
- Способность к обновлению
- Расширяемость
- Переносимость
- Возможность многократного применения
- Взаимодействие с другими ИС
- Способность к аудиту
- Способность к локализации

## Заполнить таблицу

Утверждение	Тип требований
1. Должны создаваться связи между разными группами пользователей	
2. Пользователь видит в адресной книге только тех пользователей, с которыми контактировал в течение последних 20 дней, остальных – только если выбрать опцию "Показать всех".	
3. Версия программы для MS Windows не может иметь функциональность меньше, чем существующая версия программы для Linux	
4. Новая версия программы должна позволить компании удержать существующих клиентов и укрепиться на рынке	
5. Должна обеспечиваться устойчивая работа системы при наличии до 1000 одновременно работающих пользователей	
6. Для пользователей должны быть доступны следующие действия: <ul style="list-style-type: none"><li>• создание задачи</li><li>• удаление задачи</li></ul>	

- *Строжайшее и единственное правило построения систем программного обеспечения - решить точно, что же строить. Никакая другая часть концептуальной работы не является такой трудной, как выяснение деталей технических требований, в том числе и взаимодействие с людьми, с механизмами и с иными системами ПО. Никакая другая часть работы так не портит результат, если она выполнена плохо. Ошибки никакого другого этапа работы не исправляются так трудно.*

**Ф. Брукс**

# Свойства требований

1. Полнота
2. Ясность (простота, точность, недвусмысленность)
3. Верифицируемость (тестируемость)
4. Необходимость и полезность при эксплуатации
5. Осуществимость (выполнимость, реализуемость)
6. Элементарность и трассируемость (прослеживаемость)
7. Независимость от других требований (атомарность)
8. Независимость от реализации (абстрактность)
9. Корректность (согласованность, непротиворечивость)
10. Постоянство (стабильность).



- ***Полнота требования*** означает, что текст требования не требует дополнительной детализации, то есть, в нем предусмотрены все необходимые нюансы, особенности и детали данного требования.
- Различают полноту отдельного требования и полноту системы требований.

- ***Ясность*** – недвусмысленность, определенность, однозначность спецификаций.
- Требование обладает свойством ***ясности***, если оно сходным образом воспринимается всеми заинтересованными лицами.

## **Требование 1 (неясное):**

Система не должна принимать слишком короткие пароли.

## **Требование 1 (ясное):**

Система не должна принимать пароли менее 8 символов. Если пользователь вводит менее 8 символов при выборе пароля, сообщение об ошибке должно информировать пользователя о необходимом исправлении пароля.

## **Требование 2 (неясное):**

Иногда пользователь будет вводить Код Аэропорта, который система будет распознавать. Но иногда код можно заменить близлежащим городом, и тогда пользователю не нужно знать код аэропорта, т.к. система будет понимать название города.

## **Требование 2 (ясное):**

Система должна идентифицировать аэропорт на основании Кода Аэропорта или Названия Города.

- **Верифицируемость** – пригодность к проверке. Тестировщики должны иметь возможность проверить, было ли требование реализовано корректно.
  - *Треб.1: Функция поиска должна позволять пользователю искать заказ на основе Фамилии, Имени, Даты, и т.д.*
  - *Треб. 2 Система должна препятствовать одновременному доступу **большого** числа пользователей.*
  - *Треб.3: Код аэропорта должен быть введен.*

**Задание:** Записать требования так, чтобы они стали верифицируемыми.

- **Необходимым** считается требование, невыполнение которого угрожает работоспособности или эффективности ИС.
- В требовании нет необходимости, если:
  - Ни одному заинтересованному лицу требование не нужно.
    - *Пример.* Пользователь должен иметь возможность просмотра карты аэропорта.
  - Удаление требования не повлияет на систему, т.к. оно не предоставляет никакой новой информации:
    - *Пример.* Все требования, указанные в документе Концепция, должны быть реализованы и протестированы.
- **Полезность при эксплуатации** – требование, выполнение которого повышает эргономические качества продукта.

## ■ *Осуществимость (выполнимость)*

- Требование должно быть выполнимо в рамках существующих ограничений, таких как время, деньги и доступные ресурсы.



**Выполнимость  
требования  
определяется разумным  
балансом между  
степенью  
необходимости и  
требуемыми  
ресурсами.**

- Выполнимо ли требование заказчика:  
«Реализовать новую функциональность ИС в процессе проведения опытной эксплуатации» при следующих условиях:
- Стоимость контракта на разработку ИС – 10000 у.е.
- Затраты на выполнение нового требования – 4000 у.е.
- Срок выполнения контракта – 1 год
- Новое требование возникло за 3 месяца до окончания работ.



- Требование считается *элементарным*, если оно содержит только один *трассируемый элемент*, который дает возможность отследить связь между ним и другими элементами информационной системы.
  - *Пример*: Система должна предоставлять возможность бронировать рейс, покупать билет, бронировать номер в гостинице, бронировать машину, а также предоставлять информацию о развлечениях.

**Задание:** Записать требование так, чтобы оно стало элементарным.

- Требование является *независимым*, если для его понимания не нужно знать другие требования.

- *Пример* *Треб.1: Список доступных рейсов должен включать номер рейса, время отправления и время прибытия для каждого отрезка пути.*

*Треб.2: Он должен быть отсортирован по ценам.*

- Требования должны быть *независимыми* от реализации.

- *Пример: Информация должна храниться в текстовом файле.*

- **Корректность** – согласованность, непротиворечивость. Требования не должны противоречить требованиям своего уровня иерархии и требованиям "родительского" уровня.
  - Если требование содержит факты, эти факты должны быть достоверны:
    - *Треб.1: Цена заказа должна включать все соответствующие платежи (включая стоимость пересылки – 200 руб.)*
- Требование считается **корректным**, если не существует конфликтов между ним и другими требованиями.

**Задание:** Записать требование так, чтобы оно стало корректным.

- **Прямые** конфликты возникают, когда ожидается различное поведение системы в одной и той же ситуации:

- *Треб.1(конфл.): Дата должна отображаться в формате ММ/ДД/ГГ.*
- *Треб.1 (конфл.): Дата должна отображаться в формате ДД/ММ/ГГ.*

*Требование корректное:*

*Даты должны отображаться в формате, принятом в веб-браузере пользователя.*

# Каких требований не должно быть

- Спецификация требований не должна содержать деталей проектирования или реализации.
- Требования должны отвечать на вопрос: "что должна делать система", абстрагируясь от вопроса "как она это должна делать".