



top

КОМПЬЮТЕРНАЯ  
АКАДЕМИЯ

# УПРАВЛЕНИЕ ПРОГРАММНЫМИ ПРОЕКТАМИ

# УРОК №3

## РАБОТА С ТРЕБОВАНИЯМИ, ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА, ИСПОЛЬЗУЕМЫЕ В РАМКАХ ПРОЕКТА

### СОДЕРЖАНИЕ

1. Что такое требование? .....	3
2. Классификация требований к программному продукту .....	8
Качества, которыми должны обладать хорошие требования .....	10
3. Что такое разработка требований?.....	12
4. Риски в разработке требований. ....	15
5. С-Требования (требования заказчика) и D-требования (требования разработчика) .....	19
6. Подходы для извлечения С-требований .....	21
7. Подходы к анализу С-требований. ....	25
Подходы к разработке D-требований .....	25
Пример превращения С-требования в D-требование .....	27
8. Документирование требований. ....	36
9. Управление изменениями требований. ....	37
Домашнее задание .....	41

# 1. ЧТО ТАКОЕ ТРЕБОВАНИЕ?

В индустрии разработки программного обеспечения существует несколько определений термина «требование», однако, мы будем опираться на следующее:

***Требование** – это условия или возможности, необходимые пользователю для решения проблем или достижения целей.*

© B IEEE Standard Glossary of Software Engineering Terminology (1990)

В интернете можно найти много иллюстраций, демонстрирующих важность процесса работы с требованиями. Одна из диаграмм показывает следующую статистику:

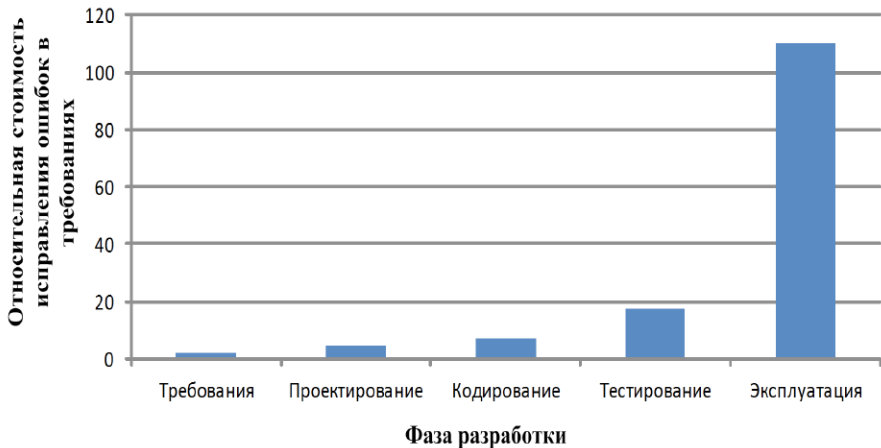


Рисунок 1

На этой диаграмме изображена зависимость стоимости внесения исправления ошибок в требованиях в зависимости от этапа проекта. Стоимость исправления ошибок на этапе эксплуатации созданного программного

продукта может превышать стоимость всего того, что уже было ранее сделано в проекте.

Можно найти и другие данные, демонстрирующие важность процесса управления требованиями к программному продукту:

- полнота требований, собранных методом обсуждения с заказчиком, составляет 25–50%, достигая в пике 70–80%;
- полнота выявления такого типа требований, как атрибуты качества, в среднем составляет 50%-60%, т. е. без специального внимания к этому типу требований аналитику удастся выявить около половины всех атрибутов качества, которые важны для заказчика проекта.

В ИТ-проектах наблюдается очень интересный эффект: практически всегда между заказчиком и командой исполнителей проекта возникает стена непонимания. Заказчик говорит на бизнес-языке, а команды – на «птичьем ИТ-языке». Устранить проблему непонимания призван бизнес-аналитик.

**Бизнес-аналитик** – это отдельная профессия в ИТ-индустрии, для овладения которой требуется достаточно много усилий и времени. У бизнес-аналитиков есть свод знаний, вбирающий в себя информацию относительно бизнес-анализа – **BAВOK** (*Business Analysis Body of Knowledge*).

BAВOK обобщает опыт ведущих бизнес-аналитиков всего мира и описывает 30 основных задач бизнес-анализа и 50 общеупотребительных техник, используемых для их решения. С использованием BAВOK проводится под-

готовка специалистов к сертификационным экзаменам Международного института бизнес-анализа (ИВА).

В задачи бизнес-аналитика при реализации проекта может входить:

- **сбор (извлечение) требований.** На этом этапе аналитику необходимо собрать максимально подробные требования к продукту проекта (например, информационной системе);
- **анализ требований.** Необходимо проверить собранные требования на полноту и реализуемость. Некоторые требования могут быть слишком сложны в реализации или вообще не реализуемы на выбранных технологиях
- **выявление проблем и противоречий.** На этом этапе аналитик должен убедиться в том, что требования не противоречат друг другу, что каждое сформулированное требование не имеет противоречий.

Пример противоречий в требованиях описан здесь: <http://triz.by/articles/refinement-of-the-notion-of-contradiction-in-triz.html>.

- **выработка решений для найденных проблем.** Для каждого сформулированного противоречия в требованиях необходимо найти решение, которое будет адекватно с точки зрения стоимости его реализации и сроков.
- **исследования.** В проектах бывают ситуации, когда для уточнения требований аналитику требуется провести дополнительное исследование. Например, клиент жалуется на то, что его сайт приносит все

меньше лидов. Аналитику нужно провести анализ сайта и выработать предложения по его улучшению.

- **прогнозирование.** Для некоторых ИТ-проектов важно, чтобы аналитик мог подсказать клиенту, какие существуют тренды в его сфере деятельности и прогнозы по развитию этого бизнеса. Например, аналитик может рассказать клиенту о новых бизнес-моделях его конкурентов.

Подробнее о бизнес-моделях можно прочесть в книге **«Построение бизнес-моделей. Настольная книга стратега и новатора»**.

Для решения всех этих задач требуются следующие качества, которыми должен обладать хороший бизнес-аналитик:

- аналитический склад ума;
- системное мышление;
- коммуникабельность;
- внимательность и организованность;
- креативность;
- обучаемость.

По статистике **The Standish Group** (мирового лидера в исследованиях в области разработки программного обеспечения) одна из ключевых причин провала ИТ-проектов – это неполные требования к продукту:

Причина провала ИТ-проекта	% ответов
Неполные требования	13,1
Отсутствие вовлечения пользователей	12,8

Нехватка ресурсов	10,6
Нереалистичные ожидания	9,9
Отсутствие поддержки со стороны руководства	9,3
Изменения требований и спецификаций	8,7
Недостаточное планирование	8,1
Невостребованность долгое время результатов проекта	7,5
Слабое управление проектом	6,2
Неверный выбор технологий	4,3
Другое	9,9

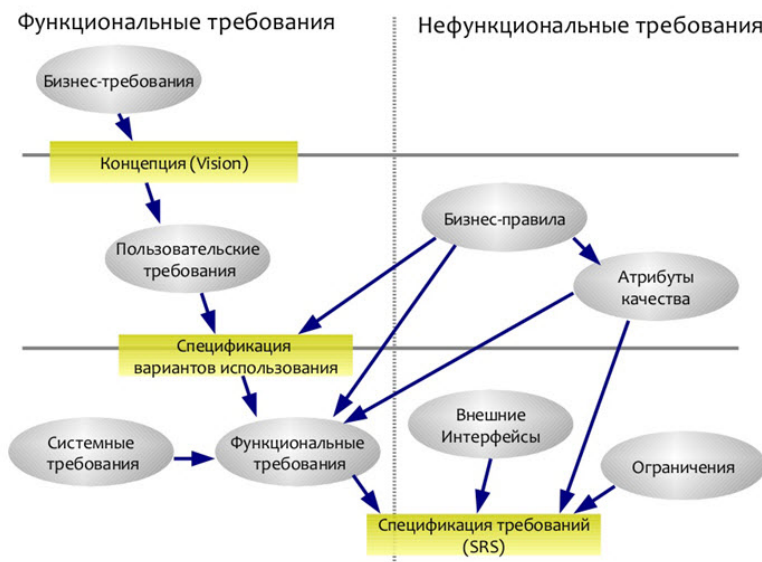
© The Standish Group. Chaos report, 2015

Так как работа с требованиями становится одним из ключевых факторов успеха ИТ-проекта, рассмотрим в следующих разделах процесс работы с требованиями и классификацию требований.

## 2. КЛАССИФИКАЦИЯ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ПРОДУКТУ

Для того чтобы не упустить важные требования к программному продукту, бизнес-аналитику следует использовать классификации требований к программному обеспечению. Один из способов классификации требований был предложен К. Вигерсом:

### Классификация требований



Условные обозначения:

- – типы информации для требований;
- – способы хранения информации (документы, диаграммы, базы данных).

Рисунок 2



**Бизнес-требования** (*business requirements*) отвечают на вопрос: зачем нужен программный продукт?

**Требования пользователей** (*user requirements*) описывают цели и задачи, которые пользователям позволит решать программный продукт.

**Функциональные требования** (*functional requirements*) описывают, что должна делать информационная система, чтобы пользователи смогли выполнить свои задачи и достичь своих целей.

**Системные требования** (*system requirements*) описывают высокоуровневые характеристики информационной системы, в т.ч.:

- требования к работе под определенными операционными системами;
- требования к аппаратному обеспечению;
- требования к линиям поддержки после внедрения программного продукта.

**Бизнес-правила** (*business rules*) описывают корпоративные политики, стандарты, процедуры и т. п.

**Атрибуты качества** (*quality attributes*) описывают информационную систему с точки зрения характеристик, важных для пользователей.

Примеры атрибутов качества:

- легкость и простота использования (*usability*);
- масштабируемость;
- интегрируемость с другими информационными системами компании.

**Ограничения** – условия, ограничивающие выбор возможных решений по реализации отдельных требований или их наборов.

**Внешние интерфейсы** – описание аспектов взаимодействия с другими системами и операционной средой. К ним относятся требования к API продукта или системы, а также требования к API других систем, с которыми осуществляется интеграция.

Зная о существующих классах требований к программному продукту, бизнес-аналитик вряд ли упустит какой-то класс требований из вида. Это значит, что он хотя бы поинтересуется у руководителя ИТ-проекта, нужно ли осуществлять сбор требований данного класса и, при положительном ответе на вопрос, планирует сбор и анализ требований для данного класса.

Замечено, что бизнес-аналитики чаще всего фокусируются на сборе и анализе бизнес-требований, пользовательских и функциональных требований, забывая при этом про другие классы требований.

## Качества, которыми должны обладать хорошие требования

Как можно понять, насколько хороши требования к программному продукту? Для этого нужно понимать какими качествами должны обладать требования. В литературе встречаются такие качества хороших требований:

- полные (т. е. достаточные);
- правильные (с точки зрения авторов);

- осуществимые (в данных условиях);
- необходимые (зачем-то и кому-то);
- приоритезированные (по важности);
- однозначные (не двусмысленные);
- проверяемые (попробуйте сами).

С одной стороны, качества хороших требований очевидны, однако на практике, разработать такие требования оказывается очень не просто.

## 3. ЧТО ТАКОЕ РАЗРАБОТКА ТРЕБОВАНИЙ?

Процесс работы с требованиями состоит из трех активностей:

1. Планирование работы с требованиями.
2. Разработка требований.
3. Управление изменениями требований.

В свою очередь, **разработка требований** состоит из:

- сбора (выявления) требований;
- анализа;
- документирования требований;
- проверки.



Рисунок 3

**Планирование требований к программному продукту** состоит в том, чтобы:

- продумать, какие инструменты (подходы) для извлечения требований использовать;
- согласовать использование данных инструментов с руководителем проекта;
- определить, у каких именно групп пользователей для данного продукта нужно извлекать требования и согласовать это с руководителем проекта.

Действия по разработке требований можно описать так.

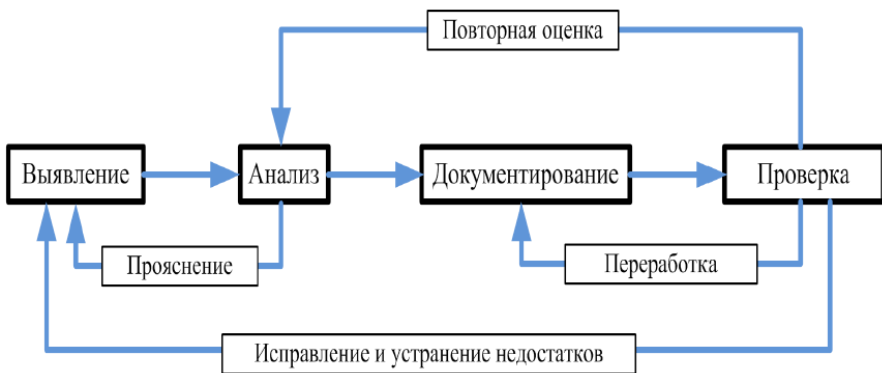


Рисунок 4

**Аналитик начинает свою работу с выявления требований.**

На этапе выявления бизнес-аналитик должен извлечь требования у заинтересованных сторон проекта (о подходах к извлечению требований будет написано ниже).

Каждое требование необходимо проанализировать на предмет его реализуемости, оценки объемов работ на реализацию, стоимости реализации и обсудить с заказчиком проекта, нужно ли реализовывать данное требование с учетом его стоимости. В ходе анализа может понадобиться прояснение некоторых моментов в требованиях.

После этого требование и подход к его реализации следует поместить в документ по требованиям.

На финальном этапе бизнес-аналитик должен убедиться в том, что требование было реализовано разработчиками правильно. Если реализация отличается от описания в документе, то возможна переработка документа по требованиям. Также по итогу реализации, заказчик проекта может внести какие-то уточнения в требования, а это приведет к необходимости вернуться на этап анализа или даже на этап выявления требований.

**Действия по управлению требованиями** могут включать:

- поддержку версии документа по требованиям в актуальном состоянии;
- анализ предлагаемых изменений в требованиях и оценку воздействия каждого изменения на объем работ проекта;
- разработку задач по реализации одобренных изменений требований;
- согласование задач из плана проекта с требованиями;
- отслеживание статусов требований.

## 4. РИСКИ В РАЗРАБОТКЕ ТРЕБОВАНИЙ

---

### 1. Недостаточное вовлечение заинтересованных сторон (в т. ч. пользователей)

Разработчики продукта не всегда понимают важность взаимодействия с пользователями продукта по поводу требований, а пользователи зачастую не представляют, чего именно они ожидают от программного продукта. Кроме формулировки: «хочу, чтобы было удобно и быстро», они самостоятельно редко могут сформулировать требования.

Недостаточное вовлечение заинтересованных сторон в проекте может привести к пропуску важных требований, что приведет к необходимости потратить незапланированное время на их обсуждение и реализацию. Все эти факторы приведут к резкому снижению вероятности для проекта уложиться в плановый срок и бюджет.

### 2. Разрастание требований пользователей

Пользователи продукта и заказчик проекта имеют склонность к тому, чтобы генерировать идеи о новых требованиях по ходу реализации проекта. Часть из этих требований имеют ценность и их вполне можно было бы реализовать, однако если проект имеет фиксированную стоимость, то добавление новых требований будет снижать прибыльность этого проекта для команды исполнителей.

Заказчик проекта и команда исполнителей должны на старте проекта договориться о том, как они будут управлять изменениями требований (*об этом см. ниже*). Иначе команда исполнителей будет препятствовать проникновению новых требований в проект.

### 3. Двусмысленность требований

Один из симптомов двусмысленности требований состоит в том, что пользователь может интерпретировать одно и то же требование двумя или более способами.

Другой симптом – у людей, изучающих требование, возникает разное представление о том, как это будет работать в продукте. Часто после того как требования реализованы, они говорят: это работает не так, как записано или не так, как мы ожидали.

Для снижения этого риска руководителю проекта и бизнес-аналитику стоит предложить нескольким пользователям выразить свое мнение относительно понимания требования. Второй способ снижения риска состоит в том, чтобы разработать вариант тестирования для требования и разработать прототип.

### 4. «Золочение» продукта

Смысл этого риска в том, что разработчики добавляют в продукт функции, которых нет в спецификации требований, при этом они верят в то, что новые функции понравятся пользователям. Однако, гораздо чаще сами пользователи требуют реализовать функции или элементы интерфейса, которые с их точки зрения будут востребованы, но по факту окажется, что они не представляют особой ценности для продукта.



Чтобы снизить эффект «золочения», стоит проследить для каждого требования пользователей в какое бизнес-требование оно входит.

### 5. Минимальная спецификация

Заказчики проекта, не понимая важности разработки детальных требований, часто не готовы платить за разработку документа по требованиям. Это подмывает команду исполнителей создать упрощенный вариант спецификации требований или ограничиться списком бизнес-требований. Такой вариант спецификации повышает рост вероятности того, что объем работ, сроки и бюджет проекта будут рассчитаны неправильно.

### 6. Пропуск групп пользователей

Если бизнес-аналитик упустил из вида группу пользователей для вашего продукта, то некоторые требования не попадут в план работ по проекту, что приведет к ошибкам в оценке сроков и бюджета проекта. Для снижения этого риска нужно провести идентификацию всех классов пользователей и убедиться, что представители этих классов будут услышаны бизнес-аналитиком.

### 7. Небрежное планирование

Все начинается с вопроса: «Я хочу реализовать вот такую штуку для нашего продукта. Когда вы сможете это сделать?». Необдуманный ответ на этот вопрос может привести к возникновению обязательства, которое потопит ваш проект. Ответ на такой вопрос лучше дать так: «На 80% я уверен, что это можно реализовать. Однако, мне нужно посоветоваться с командой и оценить стоимость реализации этой штуки и плановые сроки».

## **8. Медленная скорость обсуждений требований с пользователями**

Часто они заняты оперативной работой, не находят времени на участие в интервью, не хотят читать документы по требованиям и т. д. Надо минимизировать этот риск закладыванием резервов времени в расписание проекта на задачи по согласованию требований и в контракте с заказчиком прописывать санкции за превышение сроков по согласованию требований.

## **9. Медленная обратная связь от пользователей по реализации требований**

После того как команда реализует часть требований, она хочет получить обратную связь от пользователей, но они не хотят участвовать в митингах по обзору реализованных требований или переносят эти митинги.

## 5. С-ТРЕБОВАНИЯ (ТРЕБОВАНИЯ ЗАКАЗЧИКА) И D-ТРЕБОВАНИЯ (ТРЕБОВАНИЯ РАЗРАБОТЧИКА)

Существует еще один подход к классификации требований (описан в книге «Технология разработки программного обеспечения» Эрик Дж. Брауде, 2004), который подразумевает деление требований на 2 группы:

1. **Требования заказчика (С-требования)** – пишутся на языке, понятном заказчику проекта и отражают его ожидания и потребности.
2. **Требования разработчика (D-требования)** – пишутся на языке, понятном команде разработки и документируются в структурированной для команды форме.

Если использовать выше приведенную классификацию требований **К. Вигерса**, то можно условно разделить С-требования и D-требования так:

Классификация на С и D-требования	Классификация требований по Вигерсу
<b>С-требования</b>	Бизнес-требования Пользовательские требования Бизнес-правила Атрибуты качества
<b>D-требования</b>	Функциональные требования Системные требования Внешние интерфейс Ограничения

Оба класса требований попадают в документ SRS, структуру которого мы приводили ранее.

В случае использования шаблона SRS по IEEE в разделы «Введение» и «Общее описание» следует включить С-требования, а в раздел «Конкретные требования» попадают D-требования.

## 6. ПОДХОДЫ ДЛЯ ИЗВЛЕЧЕНИЯ С-ТРЕБОВАНИЙ

Самые распространенные подходы для извлечения требований представлены на рисунке 5.



Рисунок 5

Подходы к извлечению требований расположены на шкале сложности: слева расположены простые подходы, при продвижении по шкале слева направо сложность подходов растет. Отметим, что такое ранжирование – это субъективное мнение нескольких бизнес-аналитиков.

- **Наблюдение** – предполагает получение информации о том, как пользователи выполняют свои задачи.

Способы наблюдения могут быть разными: наблюдение через получение статистики о частоте использования тех или иных функций в программном продукте или наблюдение с присутствием бизнес-аналитика.

Например, бизнес-аналитик может находясь в офисе заказчика проекта наблюдать за тем, как выполняется работа сотрудников. По итогам наблюдений он формулирует требования о том, как эту работу можно автоматизировать в программном продукте.

- **Анкеты** – использование заранее разработанных вопросов и получение письменного ответа от заинтересованных сторон проекта на эти вопросы.

Этот подход используется в том случае, когда участнику нужно всего лишь выбрать из нескольких вариантов один или несколько ответов или когда стоимость интервью оказывается очень дорогой.

- **Анализ документов** – предполагает изучение стандартов предприятия, процедур и регламентов, касающихся правил и последовательности выполнения бизнес-процессов, которые команда планирует автоматизировать.
- **Интервью** – самый распространенный подход для сбора С-требований, предполагающий личную встречу бизнес-аналитика с заинтересованным лицом проекта.

Аналитик заранее готовит вопросы на интервью, назначает место и время встречи, а в ходе интервью задает интересующие вопросы, следит за невербальными сигналами заинтересованного лица, задает уточняющие вопросы и записывает ответы. По окончании интервью аналитик дает возможность интервьюируемому изучить письменный результат интервью и внести свои коррективы.

- **Фокус-группы** – этот инструмент пришел из маркетинга. Он позволяет собрать несколько представителей пользователей и других заинтересованных лиц проекта и провести митинг, в результате которого можно получить список требований к продукту.

- **Мозговой штурм** – этот подход давно известен и используется для того, чтобы собрать широкий спектр идей о требованиях.
- **Инновационные игры (бизнес-игры)** – подход разработан **Л. Хоманном** и популярен в США. Идея заключается в том, чтобы вовлечь заинтересованные стороны проекта в процедуру сбора требований и превратить это действие в веселую, но при этом полезную игру.
- **Прототипирование** – самый распространенный в **Agile** способ сбора требований. Команда разработки создает прототип нескольких функций программного продукта и демонстрирует его будущим пользователям. Прототип позволяет получить обратную связь и уточнить требования.
- **Моделирование процессов** – предполагает использование графических моделей деятельности пользователей.

Для сбора информации перед началом моделирования часто используются интервью, анкеты и фокус-группы. Существует огромное количество нотаций моделирования процессов, но самую большую популярность приобрели такие как: **IDEF0**, **eEPC**, **BPMN 2.0**, **UML** (особенно варианты использования – **use case**), **DFD** (диаграммы потоков данных).

- **SQFD (Software quality function deployment)** – подход, зародившийся в Японии как **QFD**, который начали использовать в разработке программного обеспечения.

В QFD используется сложный анализ матриц причинно-следственных связей между пожеланиями потребителя и техническими характеристиками продукта.

SQFD можно использовать и для анализа требований.



## 7. ПОДХОДЫ К АНАЛИЗУ С-ТРЕБОВАНИЙ

После того как С-требования собраны, бизнес-аналитику следует их проанализировать на предмет полноты, отсутствия противоречий и возможности реализации. При анализе требований бизнес-аналитик может использовать такие инструменты:

- реверсивный анализ требований;
- анализ систем-аналогов;
- ТРИЗ;
- Root Conflict Analysis Plus (RCA+);
- Value-Conflict Mapping +;
- быстрое прототипирование.

### Подходы к разработке D-требований

Для реализации С-требований, команда разработки должна сформулировать детальные D-требования. Мэпинг разделов SRS по IEEE 830 на D-требования представлен далее, на рисунке 6.

Для разработки D-требований к продукту, разработчики могут использовать следующие модели:

- диаграммы вариантов использования (**use-case diagram**) UML;
- диаграммы потоков данных (**DFD**);
- диаграммы состояний (**statechart diagram**) UML.

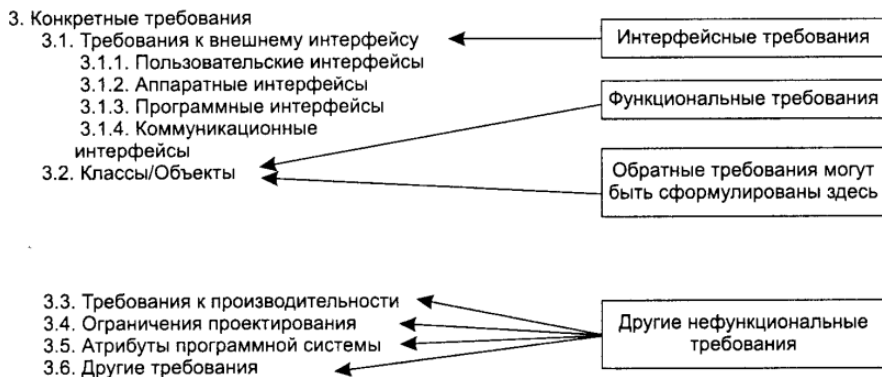


Рисунок 6

**Правила использования диаграмм** включают следующие шаги:

1. Если требование простое и влияет на другие требования, то диаграммы не используют, а требование формулируют словесно в подходящем разделе SRS.
2. Если требование представляет собой взаимодействие пользователя и приложения, то его выражают с помощью диаграммы варианта использования (use-case diagram).
3. Если требование затрагивает элементы обработки, каждый из которых получает и выдает данные, используют диаграмму потоков данных (DFD).
4. Если требование затрагивает состояния, в которых может находиться программа, строят диаграмму состояний (statechart diagram).

## Пример превращения С-требования в D-требование

**С-требование** было сформулировано бизнес-аналитиком так.

*Для регистрации потребности заказчика в перевозке груза экспедитор должен создать в системе документ «Заявка».*

Список реквизитов документа «Заявка» представлен ниже.

**Таблица 1. Реквизиты документа «Заявка»**

№	Наименование реквизита	Краткое описание
1	Номер документа	
2	Дата создания документа	
3	Заказчик (клиент)	
4	Экспедитор	
5	Грузоотправитель	
6	Грузополучатель	
7	Уведомление	Текст: «Уведомить сторону о прибытии груза»
8	Поручение	Текст: «Заказчик поручает, а Экспедитор принимает на себя обязательство по доставке груза»
9	Наименование груза	
10	Страна происхождения груза	
11	Товарный код	

№	Наименование реквизита	Краткое описание
12	Маркировка	
13	Масса (вес) груза (брутто, нетто)	
	Объем	
	Стоимость	
14	Количество грузовых мест	
15	Вид (размер) упаковки	
16	Пункт отправления	Наименование пункта отправления или передачи груза экспедитору
17	Пункт назначения	Наименование пункта назначения груза
18	Предельный срок доставки	Дата доставки груза получателю
19	Способ перевозки	Методы и способы доставки груза: вид(ы) транспорта, повагонные отправки, судовые партии, сборные отправки, перевозка в контейнерах и т. д.
20	Инструкция по перевозке груза	Свойства груза, особые условия его перевозки, иная информация о грузе
21	Требуемые документы	
22	Страхование груза	Сведения о страховании груза: застрахован, поручается застраховать и так далее
23.1	Обязанности сторон	Текст: «Обязанности сторон, кроме перечисленных в договоре»
23.2	Обязанности экспедитора	
24	Обязанности Заказчика	

№	Наименование реквизита	Краткое описание
24.1	Ответственность сторон	Текст: «Ответственность сторон, кроме перечисленных в договоре»
24.2	Ответственность экспедитора	
25	Ответственность Заказчика	
25.1	Порядок расчетов	
25.2	Стоимость услуг Экспедитора	
26	Порядок и сроки оплаты услуг	
27	Срок действия поручения	
28	Прочие условия (особые отметки)	
29	Наименование Экспедитора	
30	Подпись и печать Экспедитора	
31	Наименование Заказчика	
32	Подпись и печать Заказчика	

На следующей диаграмме представлена модель состояний заявки в системе:

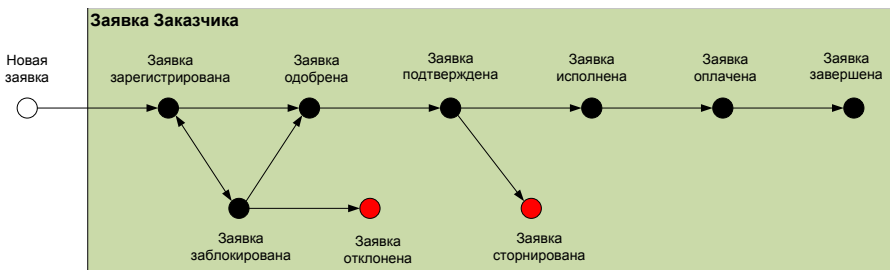


Рисунок 7

Заявка попадает в систему в процессе регистрации.

В следующей таблице представлено описание состояний заявки Заказчика в системе.

**Таблица 2. Состояния заявки Заказчика в системе**

№	Состояние	Описание состояния
1	Заявка зарегистрирована	Данные заявки Заказчика введены в систему. Заявке присвоен номер.
2	Заявка одобрена	Заявка прошла проверку и может быть выполнена. Одобрение заявки не означает, что заявка будет выполнена. См. также п. 5.
3	Заявка заблокирована	Заявка не прошла проверку по условиям выполнения заявки.
4	Заявка отклонена	Заявка может быть отклонена по одной из следующих причин: <ol style="list-style-type: none"> <li>1. Лимит дебиторской задолженности Заказчика превышен, и условия выполнения заявки не могут быть изменены соответствующим образом.</li> <li>2. Лимит дебиторской задолженности с учетом выполнения заявки будет превышен, и условия выполнения заявки не могут быть изменены соответствующим образом.</li> </ol>
5	Заявка подтверждена	Заказчику выслано подтверждение заявки в виде документа «Поручение экспедитору». Изменения, которые вносятся в заявку с этого момента, фиксируются, сохраняются и дополнительно согласовываются с Заказчиком.
6	Заявка сторнирована	Сторнирование означает отмену выполнения заявки по инициативе компании или Заказчика и, в случае возникновения финансовых обязательств (например, штрафов), их удовлетворению.

№	Состояние	Описание состояния
7	Заявка исполнена	Все заказы, связанные с заявкой, исполнены.
8	Заявка оплачена	Все взаиморасчеты между Заказчиком и компанией по заявке завершены.
9	Заявка завершена	Заявка закрыта для редактирования пользователями.

Для того чтобы превратить **С-требование** в **Д-требование**, бизнес-аналитики приняли решение создать **Use-case** работы с заявками в системе.

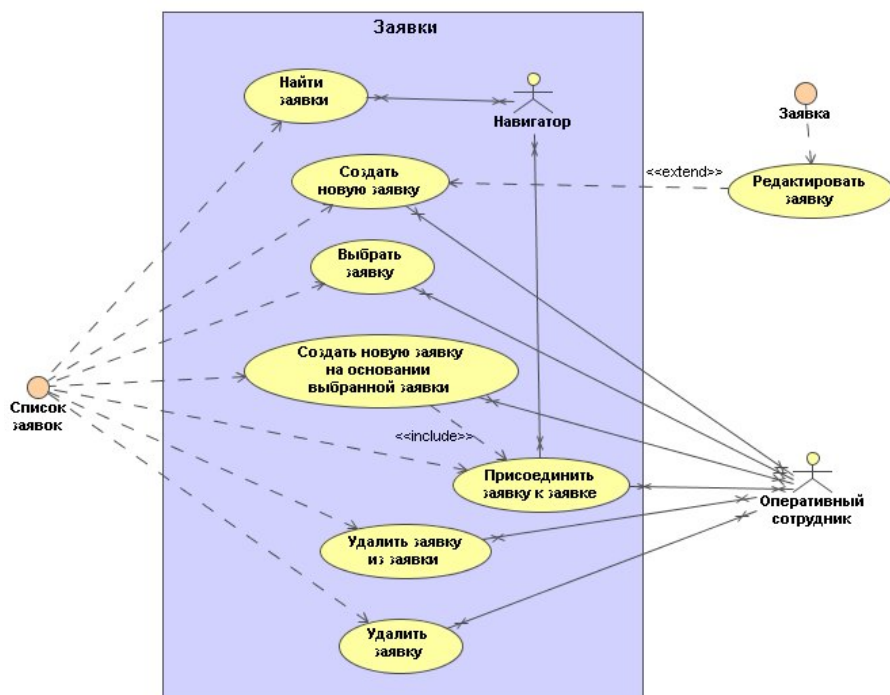


Рисунок 8

Для создания объекта **Заявка** в системе и описания его взаимодействия с другими объектами системы, бизнес-аналитики создали описание структуры данных заявки:

**Таблица 3. Структура данных объекта Заявка**

### Общие параметры

№	Реквизит документа	Описание
1	Номер документа	Номер документа в системе
2	Дата	Дата и время регистрации заявки в системе
3	Реф №	№ в заявке заказчика
4	Договор	
5	Контактное лицо	
6	Валюта	

### Груз

Представление описания груза. Вызывается подчиненный справочник **Грузы**.

### Транспорт

№	Реквизит	Краткое описание
1	Вид транспорта	
2	Вид ТС	Вызывается справочник <b>Вид ТС</b>
3	Оборудование	
4	Комментарий	



## Сервисы

№	Реквизит	Краткое описание
1	Сервис	
2	Входит в ставку	
3	Сумма	
4	Сумма, EUR	

## Контрольные точки

Вызывается справочник **Контрольные точки**.

## Параметры заявки

№	Реквизит	Краткое описание
1	Организация	
2	Подразделение	
3	Офис	
4	Куратор основной	
5	Куратор направления	
6	Группа заявок	
7	Сделка	
8	Направление	
9	Программа сделок	
10	Вид перевозки	
11	Дата погрузки (план)	

## Условия заявки

Заявка выполняется на основании условий, определенных в соответствующем договоре. Экспедитор может редактировать условия договора в заданных границах, а также добавлять другие условия.

Условия заявки представляют собой список, в котором отдельный элемент списка представляет собой отдельное условие. Структура данных, описывающая условие заявки, представлена в следующей таблице.

### Структура данных условия заявки

№	Реквизит	Краткое описание
1	Условие оплаты	
2	Условие штрафа	
3	Дополнительные условия	
4	Комментарий	

## Условие оплаты

Для заявки может быть определено только одно условие оплаты. Структура данных, описывающая условие оплаты заявки, представлена в следующей таблице.

### Структура данных параметров условия оплаты в заявке

№	Реквизит	Краткое описание
1	Текст условия	В заявке не редактируется. Изменяется автоматически при редактировании отдельных параметров условия оплаты в заявке.

№	Реквизит	Краткое описание
2	Вид оплаты	Устанавливается в договоре. В заявке не редактируется. Возможные значения: по документу, за период.
<b>Раздел «Предоплата»</b>		
3	Доля предоплаты, %	Устанавливается в договоре. В заявке редактируется в рамках разрешенного интервала.
4	Вид счета на оплату	Устанавливается в договоре. В заявке не редактируется. Возможные значения: оригинал, копия (факс), копия (электронная почта).
5	КТ периода оплаты	Устанавливается в договоре. В заявке не редактируется.
6	Направление периода оплаты (до, после)	Устанавливается в договоре. В заявке не редактируется. Возможные значения: до, после
7	Срок периода оплаты	Устанавливается в договоре. Редактируется в рамках разрешенного интервала
<b>Раздел «Оплата»</b>		
8	Вид счета на оплату	Устанавливается в договоре. В заявке не редактируется. Возможные значения: оригинал, копия (факс), копия (электронная почта)
9	КТ периода оплаты	Устанавливается в договоре. В заявке не редактируется.
10	Направление периода оплаты (до, после)	Устанавливается в договоре. В заявке не редактируется. Возможные значения: до, после.
11	Срок периода оплаты	Устанавливается в договоре. В заявке редактируется в рамках разрешенного интервала.

Данное **D-требование** бизнес-аналитики включили в раздел SRS «Конкретные требования».

## 8. ДОКУМЕНТИРОВАНИЕ ТРЕБОВАНИЙ

---

Для документирования требований обычно используются шаблоны документа SRS, структура которых была приведена в Уроке 2.

Для автоматизации документирования требований и совместной работы с документами по требованиям можно использовать специальные программные продукты, такие как **Confluence**, **Zoho Wiki** или бесплатный **Google Docs**.

## 9. УПРАВЛЕНИЕ ИЗМЕНЕНИЯМИ ТРЕБОВАНИЙ

---

Рассмотрим, как осуществляется управление изменениями требованиями при использовании **водопадной модели ЖЦ проекта** (*waterfall*) и при работе по итерационной модели, которая используется в **Agile**.

В случае, когда **Заказчик проекта** понимает, что из первоначальной концепции продукта не обязательно все требования к продукту должны попасть в его первую версию, а сроки и бюджет проекта при этом фиксируются, команда проекта должна иметь список приоритезированных требований к продукту проекта.

Приоритеты для требований нужны чтобы, в случае, если утвержденные в договоре сроки подходят к завершению и бюджета не хватает на реализацию всех требований, **Заказчику проекта** можно было быстро принимать решения от каких требований к продукту можно отказаться. При этом, заказчик проекта должен быть уверен в том, что при отказе от части требований получится продукт, который способен решать выявленные проблемы будущих потребителей.

При таком подходе управление изменениями в требованиях по ходу реализации проекта сводится к тому, чтобы постоянно ранжировать оставшиеся требования. Этот подход используется в фреймворке **Scrum**.

Рассмотрим вариант проекта, в котором требования являются фиксированными, а срок и бюджет проекта могут изменяться по ходу проекта.

В этом случае проработка требований к продуктам проекта происходит, как правило, до старта проекта реализации продукта и предполагается, что требования к продуктам изменяться не будут. Однако, по мере реализации требований и появления прототипа продукта первоначальные требования могут показаться заказчику проекта неактуальными, либо у него могут появиться новые требования.

В этом случае руководитель проекта должен быть готов к переговорам с заказчиком о необходимости внесения изменений в SRS, об изменении стоимости и сроков проекта. В случае согласования изменений в требованиях или добавления требований, ему следует инициировать подписание дополнительного соглашения к договору.

Для работы с изменениями вводится специальный документ «**Запрос на изменение**», и алгоритм принятия решения выглядит примерно так, как на рисунке 9.

Для того чтобы принять решение о внесении изменения, руководитель проекта, бизнес-аналитик и заказчик проекта должны выполнить много действий (на шагах 2 и 3 – привлекается бизнес-аналитик и заказчик проекта, на шаге 4 – нужно участие заказчика проекта). В результате на обработку каждого запроса на изменение тратится много времени и энергии. Но если этого не делать, то объем работ проекта увеличится без корректировки сроков и бюджета, что приведет к срыву первоначально утвержденных сроков и бюджета проекта.



Рисунок 9

### Подведем итоги урока.

- Хорошая работа с требованиями к продукту проекта резко повышает вероятность успеха проекта.
- Пропуск требований, обнаруженный на поздних этапах ИТ-проекта, скорее всего приведет к серьезным затратам на его реализацию.
- Основное лицо, отвечающие за работу с требованиями в ИТ-проекте – бизнес-аналитик.
- Все требования к программному продукту можно разбить на две группы: С-требования и D-требования; дополнительно к этой классификации стоит использовать классификацию К. Вигерса.

- Бизнес-аналитики должны владеть большим количеством методов и инструментов для извлечения требований.
- Требования должны быть записаны в документ, например, SRS, который обязательно должен утвердить заказчик ИТ-проекта.
- По мере реализации требований и появления первого прототипа будущего продукта у заказчика может появиться желание внести изменения в список требований. Есть два варианта управления изменением требований, – до старта проекта нужно согласовать с заказчиком один из них.



# ДОМАШНЕЕ ЗАДАНИЕ

1. Найдите в вашем окружении человека, который не доволен веб-сайтом своей компании. Проведите интервью этого человека и зафиксируйте проблему текущей версии веб-сайта.
2. Проанализируйте эти проблемы и опишите бизнес-требования к улучшению веб-сайта.
3. Найдите нескольких пользователей этого сайта и проведите интервью с ними, по итогам которого оформите 10-15 требований пользователей.
4. Найдите в вашем окружении программиста с опытом разработки веб-сайтов и отправьте ему ваш список пользовательских требований, на оценку трудоемкости.
5. По итогу оценки трудоемкости, у программиста, возможно, появятся вопросы по сути требований или по способу их реализации. Ответьте на его вопросы.
6. Зафиксируйте список задач по реализации требований и их тестированию, внесите оценки трудоемкости по каждой задаче.

## Список задач по реализации требований:

Требование	Задачи по реализации требования	Трудозатраты, чел-часов	Исполнитель