

# РАЗРАБОТКА И АНАЛИЗ ТРЕБОВАНИЙ

Лектор: Сухорукова Ирина Геннадьевна  
*ст. преподаватель кафедры программной инженерии*  
локация ауд.408 к.1

- Лекции – 8
- Лабораторные занятия ~ 7
- Зачет

Ближайшая цель: применить полученные знания в курсовом проекте по ООП

В 1994г. группой Стендиша (Standish Group) было исследовано 175 000 проектов, разрабатываемых в США. Исследования группы свидетельствуют о следующем:

- ✓ 31% проектов прекращаются до завершения
- ✓ затраты на 52,7% проектов составят 189% от первоначальной оценки...

При анализе выявлено, что три наиболее часто встречающихся ключевых фактора, создающих “проблемы” в проектах, это:

- Недостаток исходной информации от клиента: 13% всех проектов
- Неполные требования и спецификации: 12% проектов
- Изменение требований и спецификаций: 12% всех проектов

ОДНАКО около 9% проектов крупных компаний и 16% проектов мелких компаний были завершены вовремя и в пределах бюджета. Каковы главные “факторы успеха” в этих проектах?

Согласно проведенному исследованию тремя наиболее важными факторами были следующие:

- Подключение к разработке пользователя: 16% всех успешных у проектов
- Поддержка со стороны исполнительного руководства: 14% проектов
- Ясная постановка требований: 12% всех успешных проектов

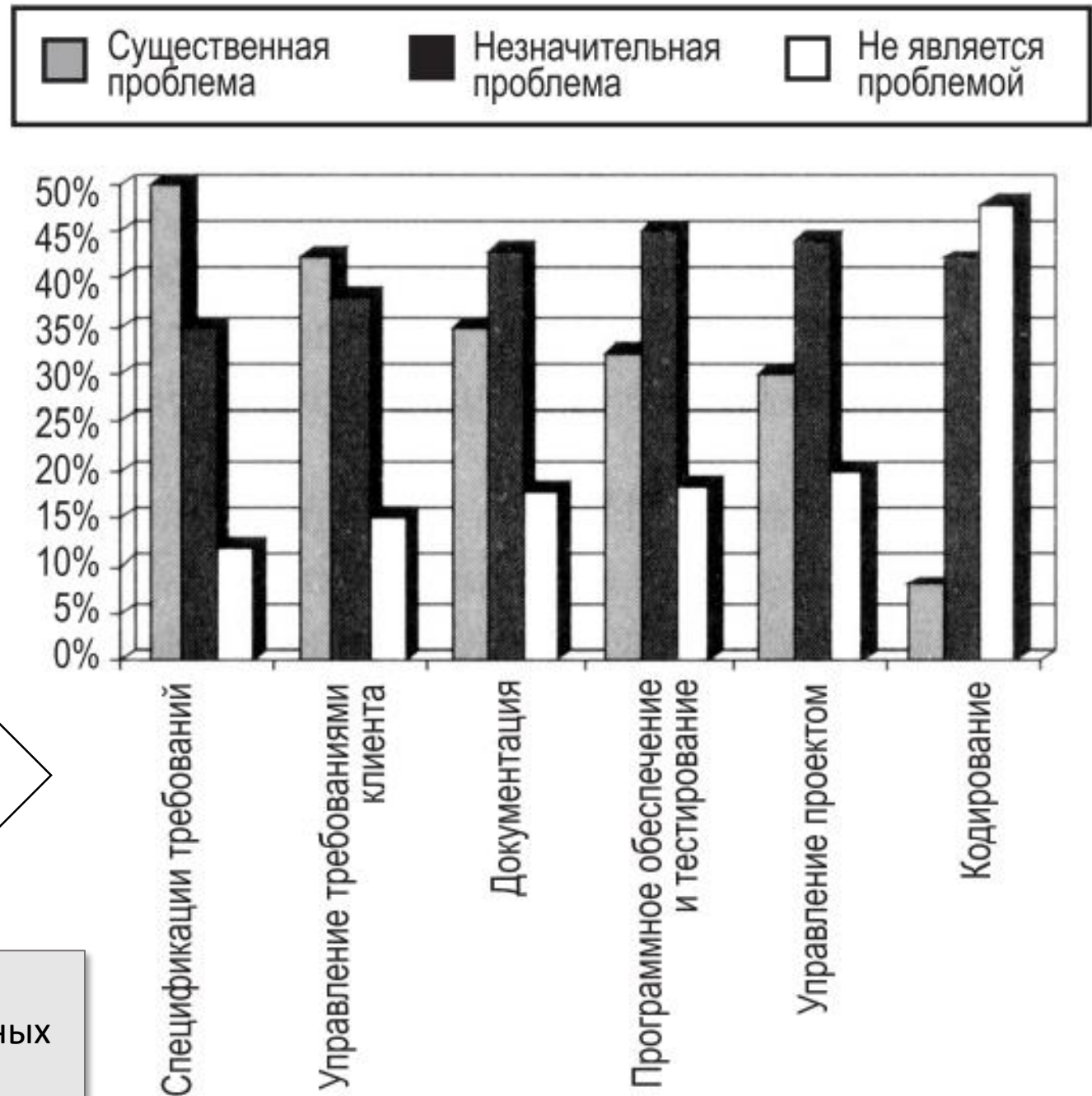
В 1995 году европейской организацией ESPIT1 был проведен **опрос с целью определить относительную важность различных проблем** существующих в отрасли. В нем приняли участие 3800 респондентов.

Двумя самыми главными проблемами оказались:

- ✓ Спецификации требований
- ✓ Управление требованиями клиента

Результаты опроса: Основные типы проблем, возникающих при разработке программного обеспечения

На ошибки, внесенные на этапе сбора требований, приходится от 40 до 50% всех дефектов, обнаруженных в программном продукте (Davis, 2005).



# Когда нет продуманного и четкого технического задания



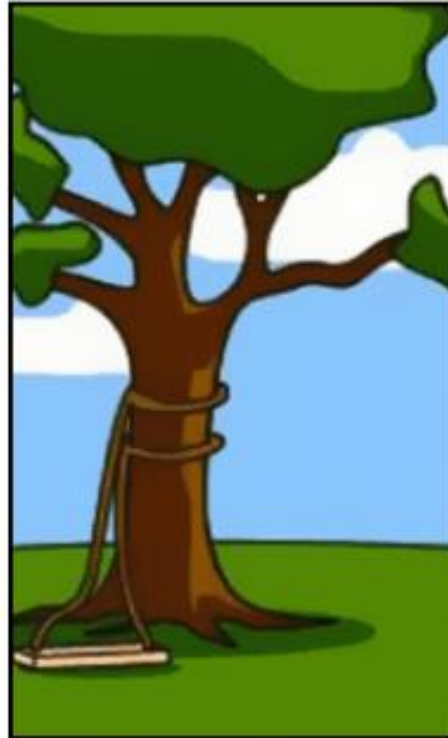
Как это объяснил  
заказчик



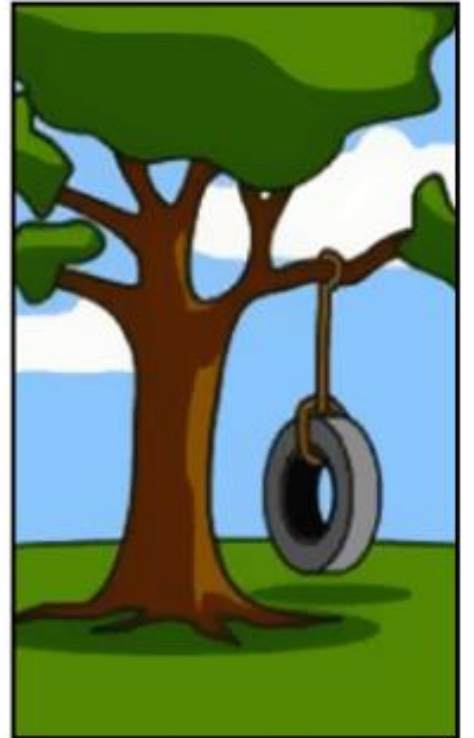
Как это понял  
руководитель проекта



Как спроектировал  
дизайнер



Как это реализовал  
программист



Что реально хотел  
заказчик

## Цели разработки требований

- обеспечение наиболее полного и точного отражения условий или возможностей, необходимых заказчику для решения его проблем и достижения бизнес-целей;
- снижение затрат на разработку, обслуживание и поддержку программного обеспечения;
- минимизация рисков переделки продукта, создания негодного ПО или срыва сроков сдачи проекта.

## Требования

- Что они из себя представляют?
- Какие виды требований выделяются?
- Как они согласуются?
- Какие источники требований можно выделить?

**SWEBOK** (Software Engineering Body of Knowledge) — международный стандарт в котором описана общепринятая сумма знаний по программной инженерии.

Документ был создан в 2004 году при сотрудничестве нескольких профессиональных организаций. В 2005 году он был принят как стандарт ISO/IEC TR 19759:2005.

В конце 2013 года была одобрена и опубликована новая версия SWEBOK V3, которая стала стандартом **ISO/IEC TR 19759:2015**.

SWEBOK описывает свод знаний, которыми должен обладать человек после 4 лет практики в сфере программной инженерии.

**Документ SE2004** (Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering) рекомендации по составлению учебных планов для университетов по специальности программная инженерия.

Разработан в 2004. Позднее был пересмотрен и из-за растущего объема знаний по теме программной инженерии был разделен на несколько документов: *Computer Engineering*, *Computer Science*, *Cybersecurity*, *Information Systems*, *Information Technology* и собственно *Software Engineering*.



**Software requirements (требования к ПО)** — одна из 15 областей знаний в сфере программной инженерии определенной В SWEBOOK .

- **software requirements** — требования к ПО;
- **software design** — проектирование ПО;
- **software construction** — конструирование ПО;
- **software testing** — тестирование ПО;
- **software maintenance** — сопровождение ПО;
- **software configuration management** — управление конфигурацией;
- **software engineering management** — управление IT проектом;
- **software engineering process** — процесс программной инженерии;
- **software engineering models and methods** — модели и методы разработки;
- **software quality** — качество ПО;
- **software engineering professional practice** — описание критериев профессионализма и компетентности;
- **software engineering economics** — экономические аспекты разработки ПО;
- **computing foundations** — основы вычислительных технологий, применимых в разработке ПО;
- **mathematical foundations** — базовые математические концепции и понятия, применимые в разработке ПО;
- **engineering foundations** — основы инженерной деятельности.

# Основные источники информации



Разработка требований к  
программному обеспечению

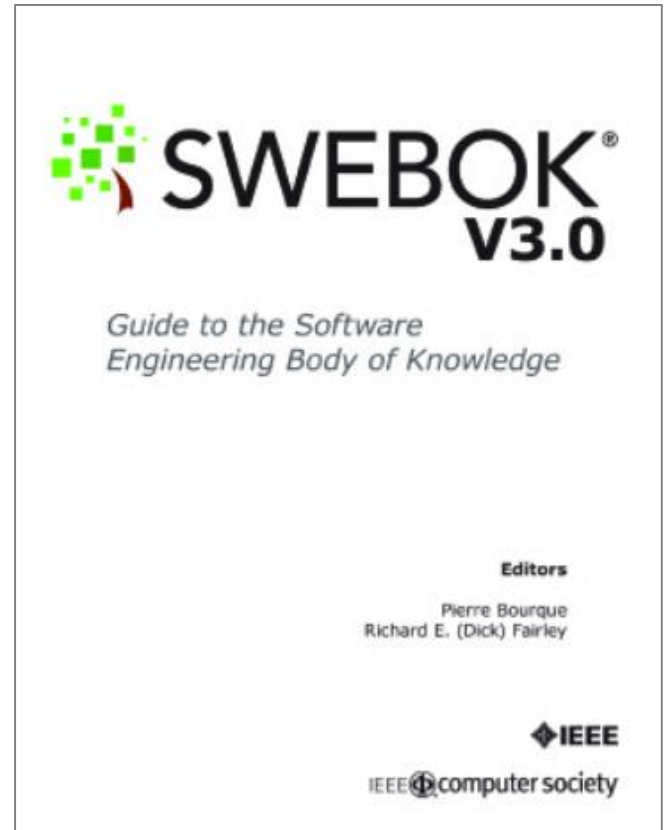
*Джой Битти, Карл Вигерс*

Свод знаний по  
программной  
инженерии SWEBOK

*можно найти русский перевод*

Лекции на  YouTube

Статьи от практиков





Вопрос, что считать требованием к ПО, является дискуссионным.

Поэтому **обратимся к уже устоявшимся определениям:**

**Требования** — это спецификация того, что должно быть реализовано. В них описано поведение системы, свойства системы или ее атрибуты. Они могут служить ограничениями в процессе разработки системы (*Ian Sommerville u Pete Sawyer, 1997*).

**Требования** — совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Создаются в процессе разработки требований к программному обеспечению (ПО), в результате анализа требований (*Википедия*).

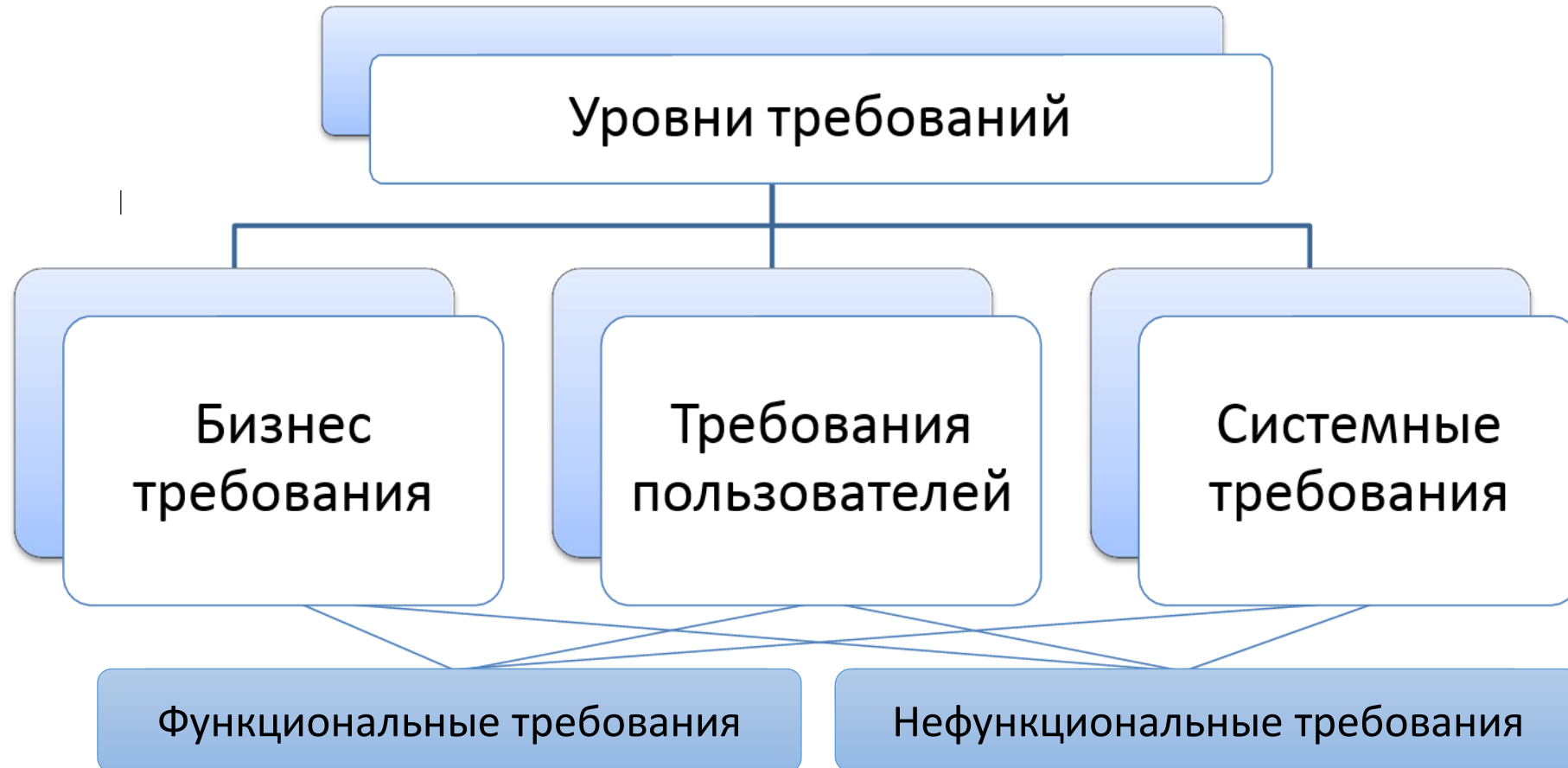
**Требования** – это:

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.

*Определение требований в соответствии с  
Standard glossary of Software Engineering Terminology IEEE (1990)*

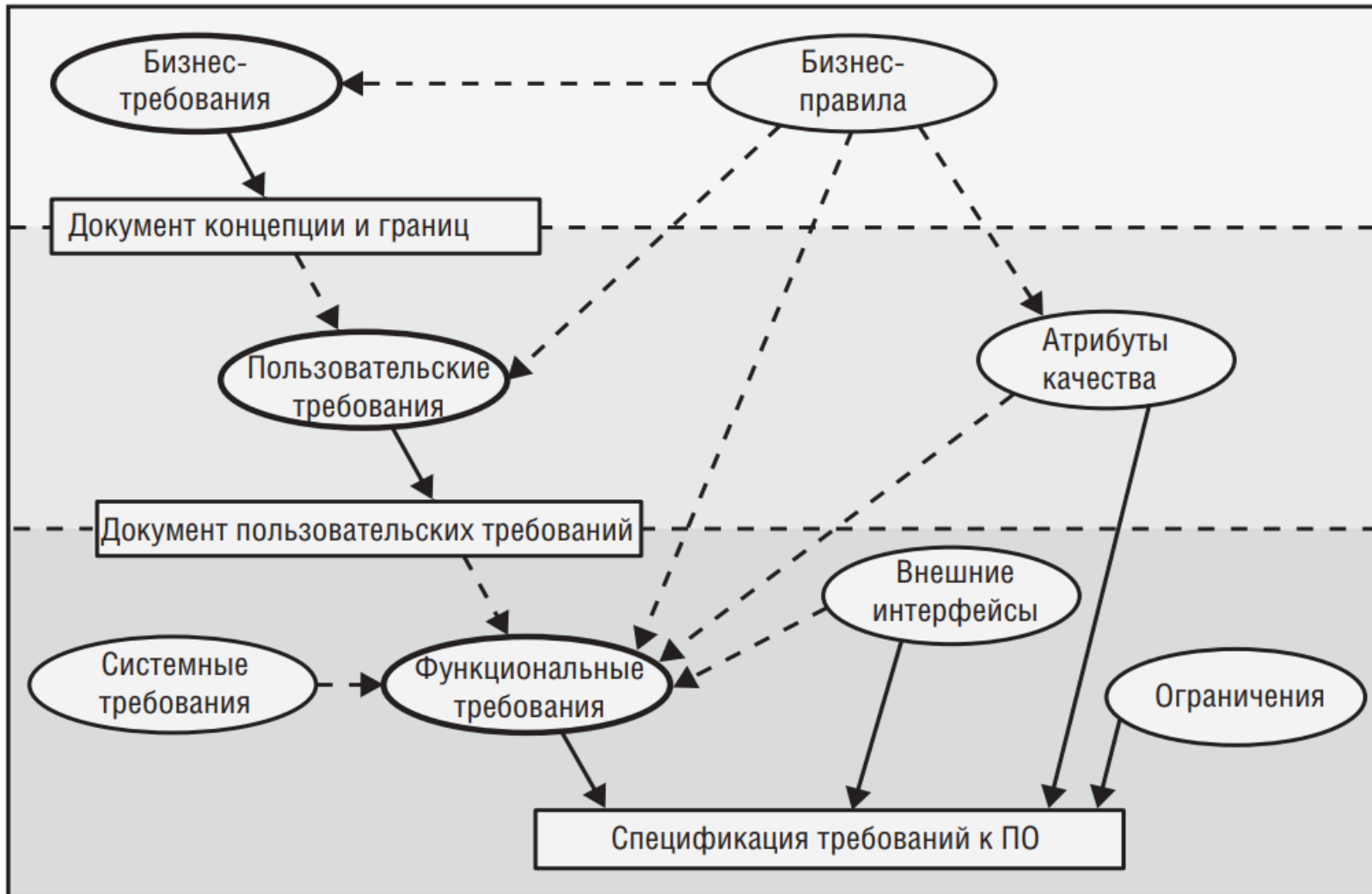
# Классификация требований

Термин «требование» охватывает довольно широкую предметную область. Поэтому возникает вопрос типизации и классификации требований.



Понятие	Определение
Бизнес-требование	Высокоуровневая бизнес-цель организации или заказчиков системы
Бизнес-правило	Политика, предписание, стандарт или правило, определяющее или ограничивающее некоторые стороны бизнес-процессов. По своей сути это не требование к ПО, но оно служит источником нескольких типов требований к ПО
Ограничение	Ограничение на выбор вариантов, доступных разработчику при проектировании и разработке продукта
Внешнее требование к интерфейсу	Описание взаимодействия между ПО и пользователем, другой программной системой или устройством
Характеристика	Одна или несколько логически связанных возможностей системы, которые представляют ценность для пользователя и описаны рядом функциональных требований
Функциональное требование	Описание требуемого поведения системы в определенных условиях
Нефункциональное требование	Описание свойства или особенности, которым должна обладать система, или ограничение, которое должна соблюдать система
Атрибут качества	Вид нефункционального требования, описывающего характеристику сервиса или производительности продукта
Системное требование	Требование верхнего уровня к продукту, состоящему из многих подсистем, которые могут представлять собой ПО или совокупность ПО и оборудования
Пользовательское требование	Задача, которую определенные классы пользователей должны иметь возможность выполнять в системе, или требуемый атрибут продукта

Требования к ПО состоят из трех уровней — бизнес-требования, пользовательские и функциональные требования. Вдобавок в каждой системе есть свои нефункциональные требования. На рис. схематично представлена взаимосвязь различных типов требований. Сплошные линии означают «содержатся в», а пунктирные — «являются отправной точкой» или «влиют на»



Овалы обозначают типы информации требований, а прямоугольники — документы, в которых хранится эта информация. Сплошные линии указывают, что в указанном документе хранится информация определенного типа. (Бизнес-правила и системные требования хранятся отдельно от требований к ПО, обычно соответственно в каталоге бизнес-правил или в спецификации системных требований.) Пунктирная линия указывает, что информация одного типа является источником или влияет на информацию другого типа или на требование.



# Бизнес-требования BRQ — «Зачем?»

**Бизнес-требования** (business requirements) описывают, почему организации нужна такая система, то есть цели, которые организация намерена достичь с ее помощью. Основное их содержание — бизнес-цели организации или клиента, заказывающих систему.

Бизнес-требования — это верхний уровень абстракции требований к системе. Они не относятся напрямую к реализации проекта, а в первую очередь отражают цели бизнеса, абстрагированные от реализации системы.

Как правило, бизнес-требования высказывают те, кто финансируют проект, покупатели системы, управляющий реальными пользователями, отдел маркетинга или ответственный за концепцию продукта.

Бизнес-требования обычно записывают в форме *документа о концепции и границах (vision and scope document)*. К другим руководящим документам, которые еще иногда используют в этом качестве, относят *устав проекта (project charter)*, *вариант использования (business case)* или *документ рыночных требований (market requirements document)*.



# Примеры бизнес требований

<b>Цель</b> (это ориентиры, которые организация или заказчик намерены достичь)	<b>Концепция</b> (система взглядов на будущий продукт, описывающая видение группы заинтересованных лиц)
Сократить время обработки заказа на 50%	Система должна предоставить интерфейс, упрощающий создание заказа
Увеличить количество клиентов до 35%	В системе должны быть механизмы побуждения клиента к заказу
Авиакомпания хочет на 25% снизить затраты на сотрудников у стойки в аэропорту	? ? ?



# Пользовательские требования URQ

**Пользовательские требования** (user requirements) описывают цели или задачи, которые пользователи должны иметь возможность выполнять с помощью продукта, который в свою очередь должен приносить пользу кому-то. Область пользовательских требований также включает описания атрибутов или характеристик продукта, которые важны для удовлетворения пользователей.

Пользовательские требования могут быть представлены в виде:

- текстового описания,
- диаграммы вариантов использования (Use Case),
- пользовательских историй (User Story).

Пользовательские требования описывают, что пользователь должен иметь возможность делать с системой.

**Пользовательские требования** определяют набор пользовательских задач, которые должна решать программа, а также способы (сценарии) их решения в системе

## Пользовательская история

Как пассажир я хочу зарегистрироваться на рейс, чтобы можно было сесть на самолет

## Сценарий

Регистрация на рейс с использованием веб-сайта или терминала в аэропорту

 FRQ

# Функциональные требования FRQ — «Что делает?»

**Функциональные требования (functional requirements)** определяют, каким должно быть поведение продукта в тех или иных условиях. Они определяют, что разработчики должны создать, чтобы пользователи смогли выполнить свои задачи (пользовательские требования) в рамках бизнес-требований.

Функциональные требования самые низкоуровневые. Являются результатом декомпозиции верхнеуровневых требований и описывают атомарные функции, которые должны быть реализованы в системе.

Функциональные требования описываются в форме традиционных утверждений со словами «должен» или «должна»:

Пользователь должен иметь возможность  
добавить объект в избранное

FRQ

Если в профиле пассажира не указаны предпочтения  
по выбору места, система резервирования должна  
сама назначить ему место

FRQ

**Бизнес-аналитик** документирует функциональные требования в *спецификации требований к программному обеспечению* (*software requirements specification*, **SRS**)

«Бизнес-аналитик» — это роль в проекте, которая прежде всего отвечает за действия по работе с требованиями в проекте.

Функциональные требования самые низкоуровневые. Являются результатом декомпозиции верхнеуровневых требований

## Пример функциональных требований:

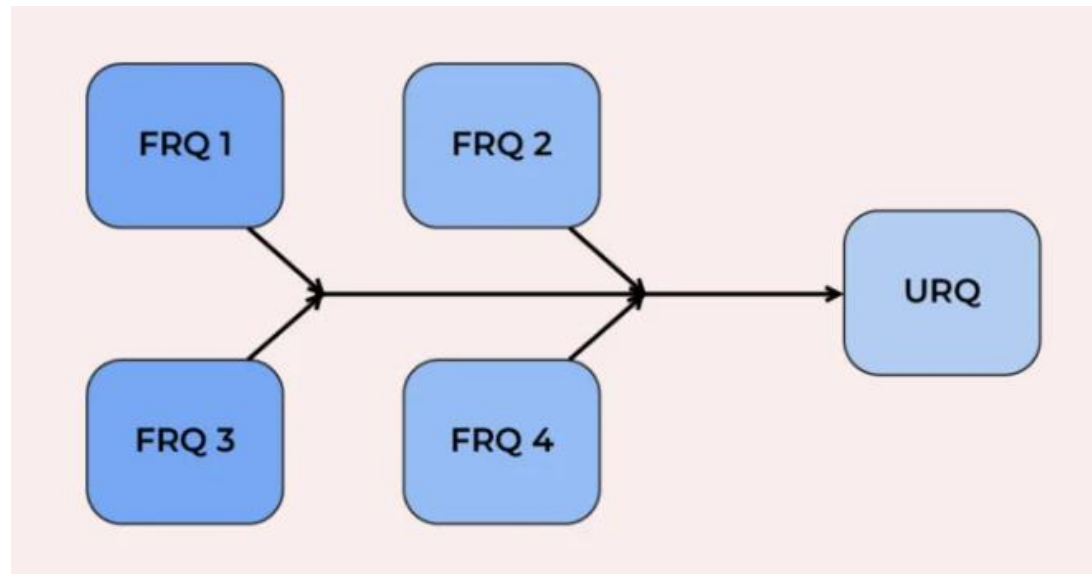
Пользователь должен иметь возможность добавить объект в избранное (URQ):

FRQ 1 — Добавить в избранное.

FRQ 2 — Удалить из избранного.

FRQ 3 — Редактирование дополнительных атрибутов.

FRQ 4 — Обращение к объекту из меню избранного.

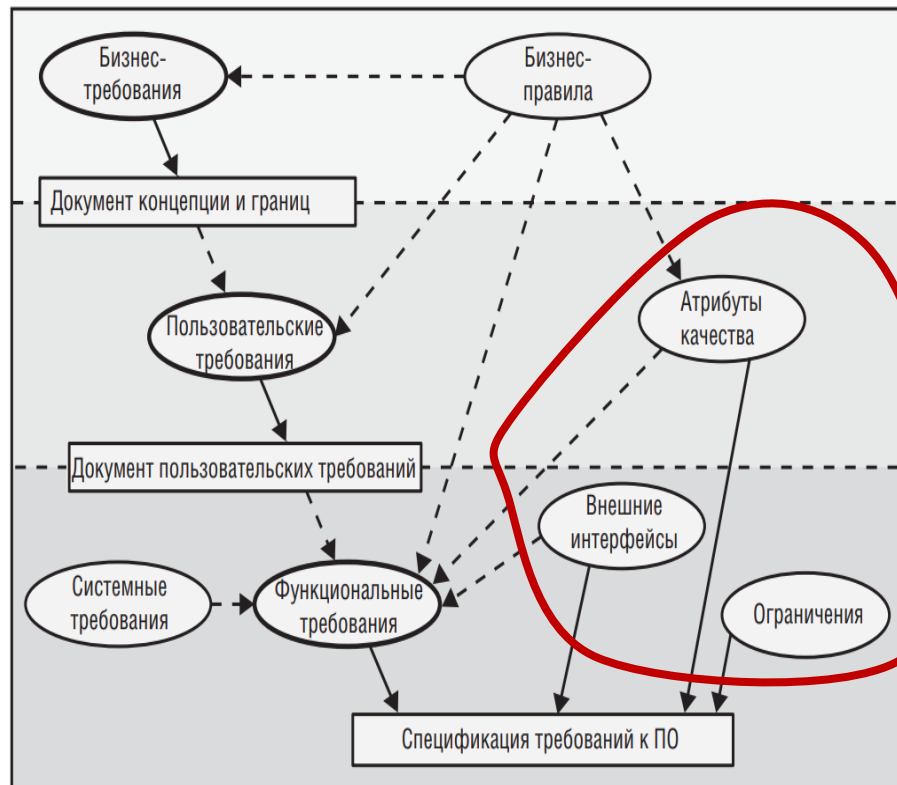




# Нефункциональные требования NFRQ — «Как делает?»

Нефункциональные требования определяют характеристики и ограничения системы и **не связаны непосредственно с функциональными требованиями**. Они формируются на основе имеющихся атрибутов качества, требований к внешнему интерфейсу и ограничений:

Нефункциональные требования напрямую или косвенно влияют на формирование каждого уровня требований.



— требования к характеристикам качества

требования к надежности  
требования к совместимости  
требования к эффективности  
требования к гибкости  
требования к эргономике

— ограничения

соответствия стандартам и правилам  
предопределенные архитектурные решения  
бюджет  
сроки  
и т.д.

Пример:

Ключом успеха проекта Тик Ток являются его нефункциональные требования. Какие?



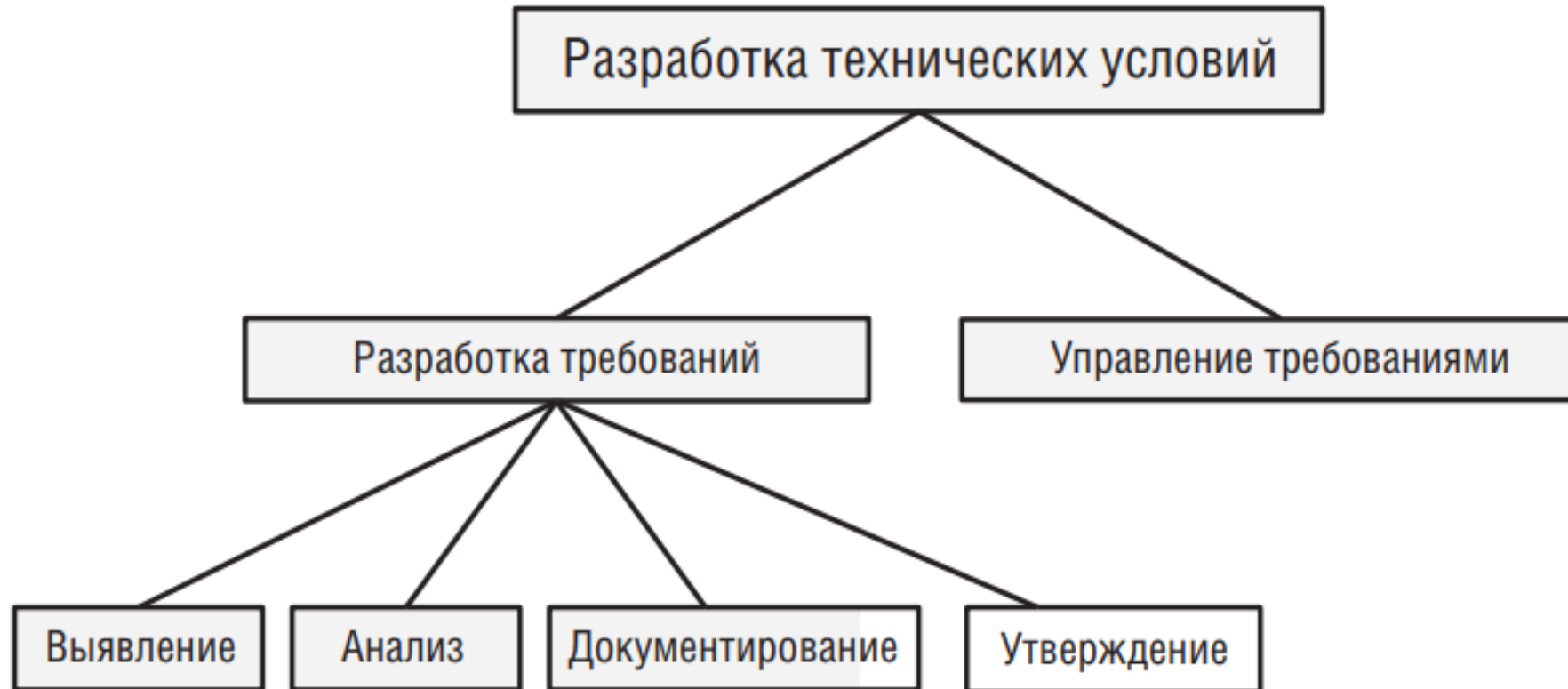
Пример: нефункциональные требования написаны черным цветом

1. Работает в режимах: «Обычный», «Инженерный» и «Программист»
2. Выполняет арифметические операции
3. Совместим с Windows
4. Выполняет логические операции
5. Вычисляет сложные функции, ...
6. Время вычисления тригонометрических функций меньше 1 минуты
7. Наличие графического пользовательского интерфейса
8. Наличие справки
9. Справка выводится в формате Windows
10. Память, отводимая на одно число равна ...
11. Реализация памяти
12. Поддержка скобок

# Разработка и управление требованиями

Область разработки технических условий полезно разделить на:

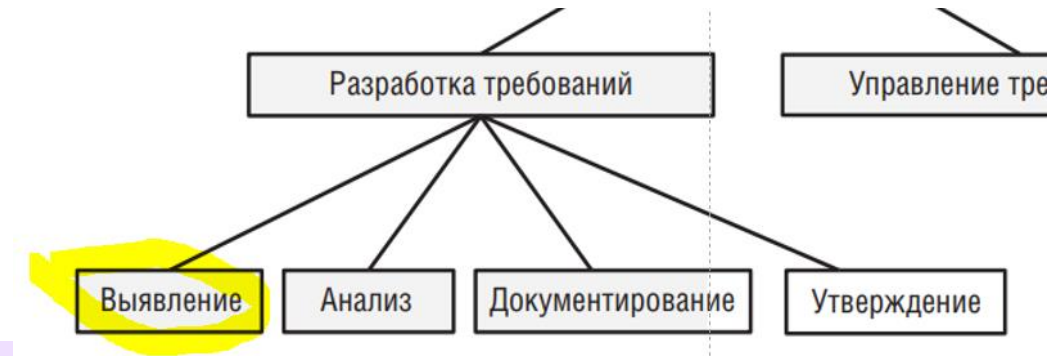
- разработку требований (requirements development) и
- управление требованиями (requirements management).



# Разработка требований

Разработка требований подразделяется на:

- выявление (elicitation),
- анализ (analysis),
- документирование (specification)
- утверждение (validation).



## Выявление и сбор требований (elecitation)

Этот этап включает в себя все действия, связанные с выявлением требований, таких как *интервью, совещания, анализ документов, создание прототипов* и другие. К ключевым действиям относятся:

- ✓ Определение классов ожидаемых пользователей продукта и других заинтересованных лиц;
- ✓ Понимание задач и целей, а также бизнес-целей, которым соответствуют эти задачи;
- ✓ Изучение среды, в которой будет использоваться новый продукт;
- ✓ Работа с отдельными людьми для понимания их потребностей и ожидания в отношении качества.

## Анализ требований (analyzing requirements)

Этот этап подразумевает получение более обширного и точного понимания всех требований и представление наборов требований в различном виде.



### Основными действиями на этапе анализа требований будут:

- анализ информации и отделение функциональных требований от нефункциональных, бизнес-правил, предполагаемых решений и другой информации;
- разложение высокоуровневых требований до нужного уровня детализации;
- выведение функциональных требований из информации других требований;
- распределение требований по компонентам ПО;
- согласование приоритетов реализации;
- понимание относительной важности атрибутов качества;
- выявление пробелов в требованиях или излишних требований, не соответствующих заданным рамкам.

# Документирование

Документирование требований предусматривает представление и хранение совокупного знания о требованиях постоянным и хорошо организованным способом.

К ключевому действию относится:

преобразование собранных потребностей пользователей в письменные требования и диаграммы, пригодные для понимания, анализа и использования целевой аудиторией.



## Утверждение требований

На этом этапе подтверждается правильность имеющегося набора требований, которые позволят реализовать решение, удовлетворяющее бизнес-целям.

Основные действия:

- проверка задокументированных требований для устранения всех недостатков до принятия требований группой разработки;
- разработка приемочных тестов и критериев, которые должны подтвердить, что созданный на основе требований продукт будет отвечать потребностям заказчика и удовлетворять поставленным бизнес-целям.



# Управление требованиями

## К действиям по управлению требованиями относятся:

- определение основной версии требований, моментальный снимок, который представляет согласованный, проверенный и одобренный набор функциональных и нефункциональных требований, обычно для конкретного выпуска продукта или итерации разработки;
- оценка влияния предлагаемых требований и внедрение одобренных изменений в проект управляемым образом;
- обновление планов проекта в соответствии с изменениями в требованиях;
- обсуждение новых обязательств, основанных на оцененном влиянии изменения требований;
- определение отношений и зависимостей, существующих между требованиями;
- отслеживание отдельных требований до их проектирования, исходного кода и тестов;
- отслеживание состояния требований и действий по изменению на протяжении всего проекта.

Предмет управления требованиями заключается не в предотвращении изменения или усложнении их внесения — задача состоит в предугадывании и приспособлении к ожидаемым реальным изменениям, чтобы снизить их разрушительное влияние на проект.

# Выявление требований

## Источники требований

- ✓ Заинтересованные стороны
- ✓ Существующие системы
- ✓ Существующие документы
- ✓ Конкуренты и др. подобные системы
- ✓ Интерфейсы с системами
- ✓ Законы и стандарты
- ✓ Политика компании

## Кто такие заинтересованные стороны?

- **"Клиент"** – люди, которые платят за разработку системы.
- **Пользователи** - их потребности — это то, чему организация должна уделять первостепенное внимание.
- **Эксперты домена** – Это специалисты, которые знают среду, в которой будет использоваться продукт.
- **Инспекторы** – Они являются экспертами в области государственных норм и правил и безопасности, требуемой проектом.
- **Адвокаты** – Они являются экспертами, когда речь идет о законах, а также о стандартах, которые следует учитывать при разработке продукта.

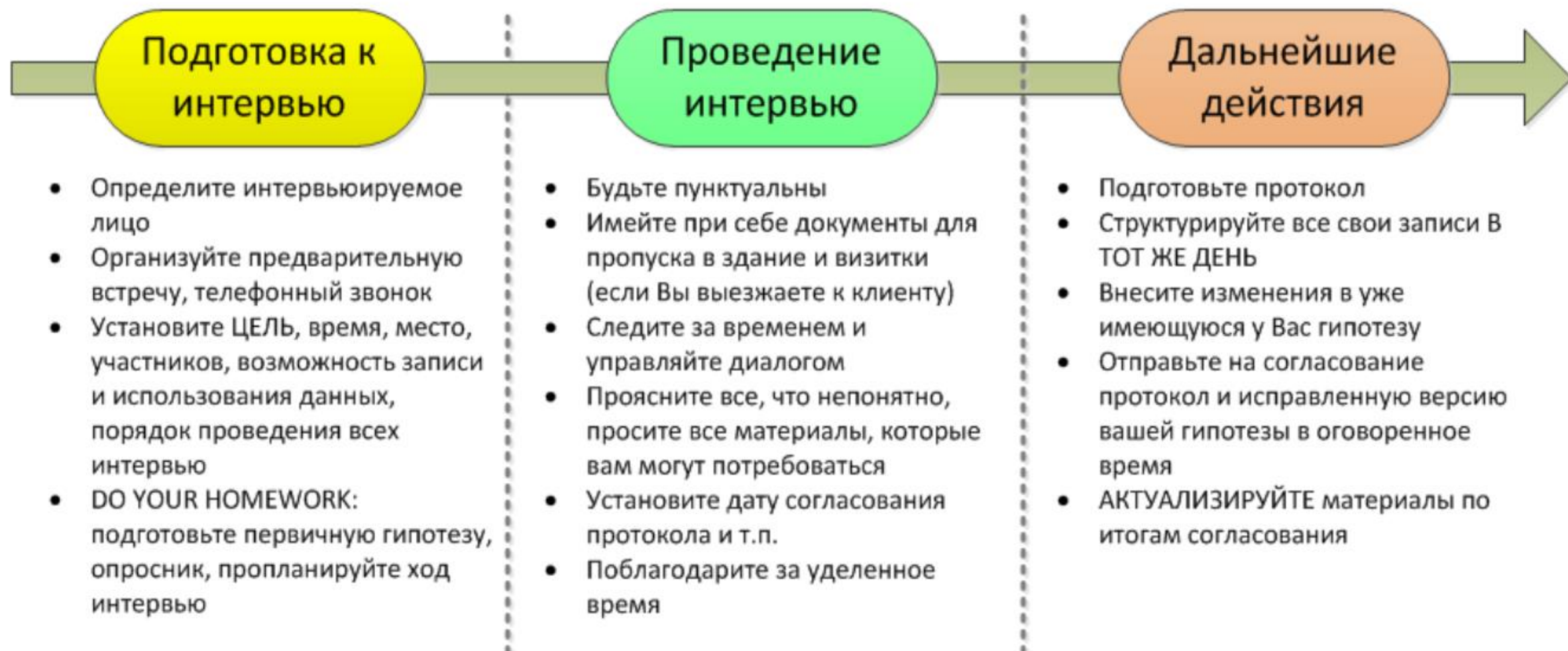
...

## Методы выявления требований

- ✓ Интервью, опросы, анкетирование
- ✓ Мозговой штурм, семинар
- ✓ Наблюдение за производств. деятельностью, «фотографирование» рабочего дня
- ✓ Анализ нормативной документации
- ✓ Анализ моделей деятельности
- ✓ Анализ конкурентных продуктов
- ✓ Анализ статистики использования предыдущих версий продукта

Заинтересованное лицо (**stakeholder**) — это человек, группа или организация, которая активно задействована в проекте, подвержена влиянию процесса или результата или может влиять на процесс или результат.

## Примерный чек-лист для специалиста по сбору требований



**? Как думаете, что лучше выбрать: интервью или анкетирование ?**



## **Сотрудничество клиентов и разработчиков**

Отличные программные продукты — результат правильно выполненного проектирования, основанного на требованиях, полученных в результате тесного взаимодействия разработчиков и клиентов. Совместная работа возможна только тогда, когда все участники процесса разработки знают, что именно необходимо им для успеха, и когда они понимают и уважают стремление их соратников к успеху.

«Билль о правах клиента ПО» содержит 10 положений, на выполнении которых клиенты могут на вполне законных основаниях настаивать при общении с аналитиками и разработчиками на этапе формулирования требований к проекту.

### **Билль о правах клиента ПО при формировании требований**

#### **У вас есть право**

1. Иметь дело с аналитиком, который разговаривает на вашем языке
2. Иметь дело с аналитиком, хорошо изучившим ваш бизнес и цели, для которых создается система
3. Потребовать, чтобы аналитик зафиксировал требования в надлежащей форме
4. Получить подробный отчет о будущих процедурах и результатах процесса формулирования требований
5. На изменение ваших требований
6. На взаимное уважение
7. Знать о вариантах и альтернативах требований и их реализации
8. Описать характеристики, упрощающие работу с продуктом
9. Узнать о способах корректировки требований для ускорения разработки за счет повторного использования
10. Получить систему, функциональность и качество которой соответствует вашим ожиданиям.

Так как обратной стороной прав являются обязанности, «Билль об обязанностях клиента ПО», напротив, содержит 10 положений, определяющих ответственность клиента перед аналитиком и разработчиком на этапе формулирования требований. Возможно, его стоит назвать «Билль о правах разработчика».

## Билль об обязанностях клиента ПО при формировании требований

### Клиент обязан

1. Ознакомить аналитиков и разработчиков с особенностями вашего бизнеса
2. Потратить столько времени, сколько необходимо на уточнение требований
3. Точно и конкретно описать требования к системе
4. Принимать своевременные решения относительно требований
5. Уважать определенную разработчиком оценку стоимости и возможности реализации ваших требований
6. Определять реалистичные приоритеты требований совместно с разработчиками
7. Проверять требования и оценивать прототипы
8. Определить критерии приемки
9. Своевременно сообщать об изменениях требований
10. Уважительно относиться к процессам создания требований



# Use Case (вариант использования, прецедент использования)

- Use Case описывает сценарий взаимодействия участников (как правило — пользователя и системы).
- Участников может быть 2 и больше.
- Пользователем может выступать как человек, так и другая система.

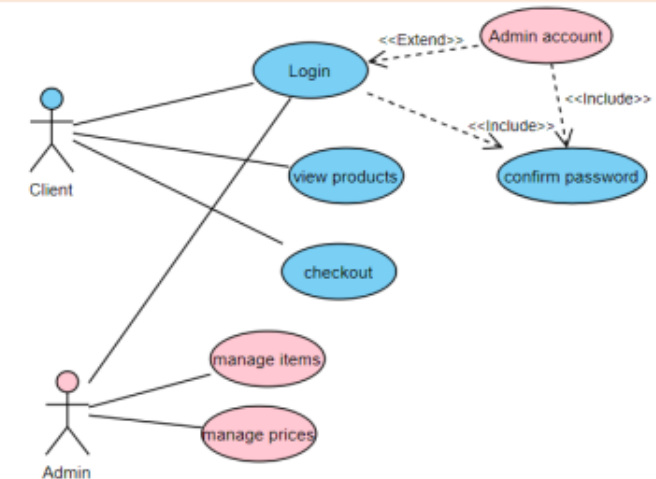
## Текстовое описание

**Пример 1.** Разблокировать учетную запись пользователя (простой короткий пример, без альтернативного потока событий):

Действующие лица	Администратор, Система
Цель	Изменить статус учетной записи пользователя на «активный».
Предусловие	Учетная запись пользователя не активна.
Успешный сценарий:	<ol style="list-style-type: none"><li>Администратор выбирает пользователя и активирует «Разблокировать».</li><li>Система переключает учетную запись пользователя в статус «активный», и посылает сообщение (тут можно сослаться на текст сообщения из списка сообщений, см. примечание ниже) пользователю на email (если «User Account → email» не пусто).</li></ol>
Результат	Учетная запись пользователя была переведена в статус «активный».

или

## UML диаграмма вариантов использования



Полезная статья о том КАК !!!  
<https://habr.com/ru/post/566218/>

## Словесное описание Use Case

Пример 1. Разблокировать учетную запись пользователя (простой пример):

<b>Действующие лица</b>	Администратор, Система
<b>Цель</b>	Изменить статус учетной записи пользователя на «активный».
<b>Предусловие</b>	Учетная запись пользователя не активна.

**Успешный сценарий:**

1. Администратор выбирает пользователя и активирует «Разблокировать».
2. Система переключает учетную запись пользователя в статус «активный», и посылает сообщение пользователю на email (если «User Account → email» не пусто).

<b>Результат</b>	Учетная запись пользователя была переведена в статус «активный».
------------------	--

## Пример 2. Авторизация пользователя:

## Словесное описание Use Case

**Действующие лица** Пользователь, Система

**Цели** Пользователь: авторизоваться в системе и начать работать; Система: идентифицировать пользователя и его права.

### Успешный сценарий:

1. Пользователь запускает систему. Система открывает сессию пользователя, предлагает ввести логин и пароль.
2. Пользователь вводит логин и пароль.
3. Система проверяет логин и пароль.
4. Система создает запись в истории авторизаций (IP адрес пользователя, логин, дата, рабочая станция).
5. Система выдает пользователю сообщение по поводу успешной авторизации (ссылка на сообщение).

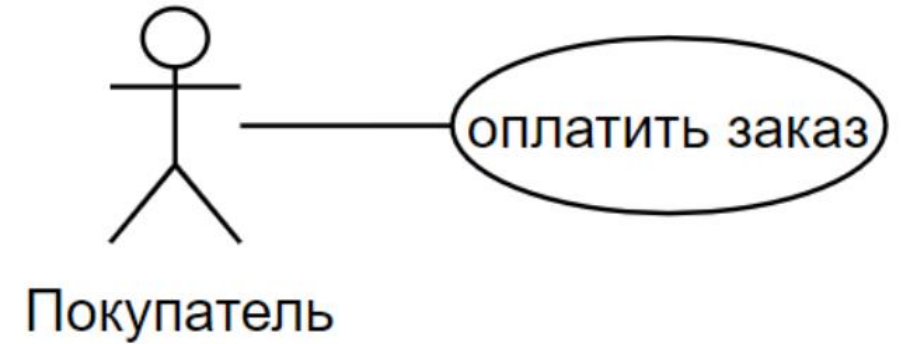
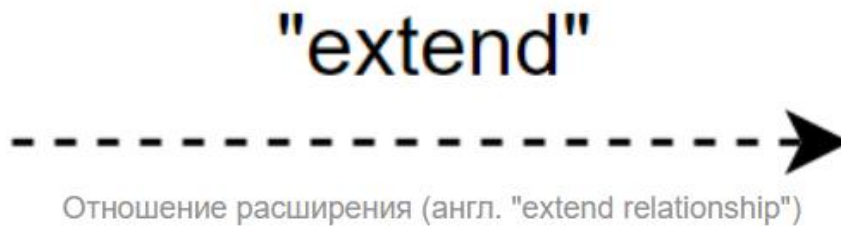
**Результат** Пользователь успешно авторизирован и может работать с системой.

### Расширения:

*a	Нет доступа к БД. Система выдает сообщение (ссылка на сообщение). Результат: пользователь не может войти.
1a	В настройках безопасности для данного IP адреса существует запрет на вход в систему. <b>Результат:</b> форма логина не предоставляется, система выдает сообщение пользователю (ссылка на сообщение).
2a	Пользователь выбирает: «Напомнить пароль». Вызывается сценарий «Напомнить пароль».
3a	Пользователь с введенными логином и паролем не найден. <b>Результат:</b> отказ в авторизации. Система выдает сообщение (ссылка на сообщение). Переход на шаг 2.
3б	Количество неудачных попыток авторизоваться достигло максимального, установленного в настройках. <b>Результат:</b> пользователь не может войти. Выдается сообщение: (ссылка на сообщение). Вход с IP адреса Пользователя заблокирован на время, установленное в настройках.

# Use Case

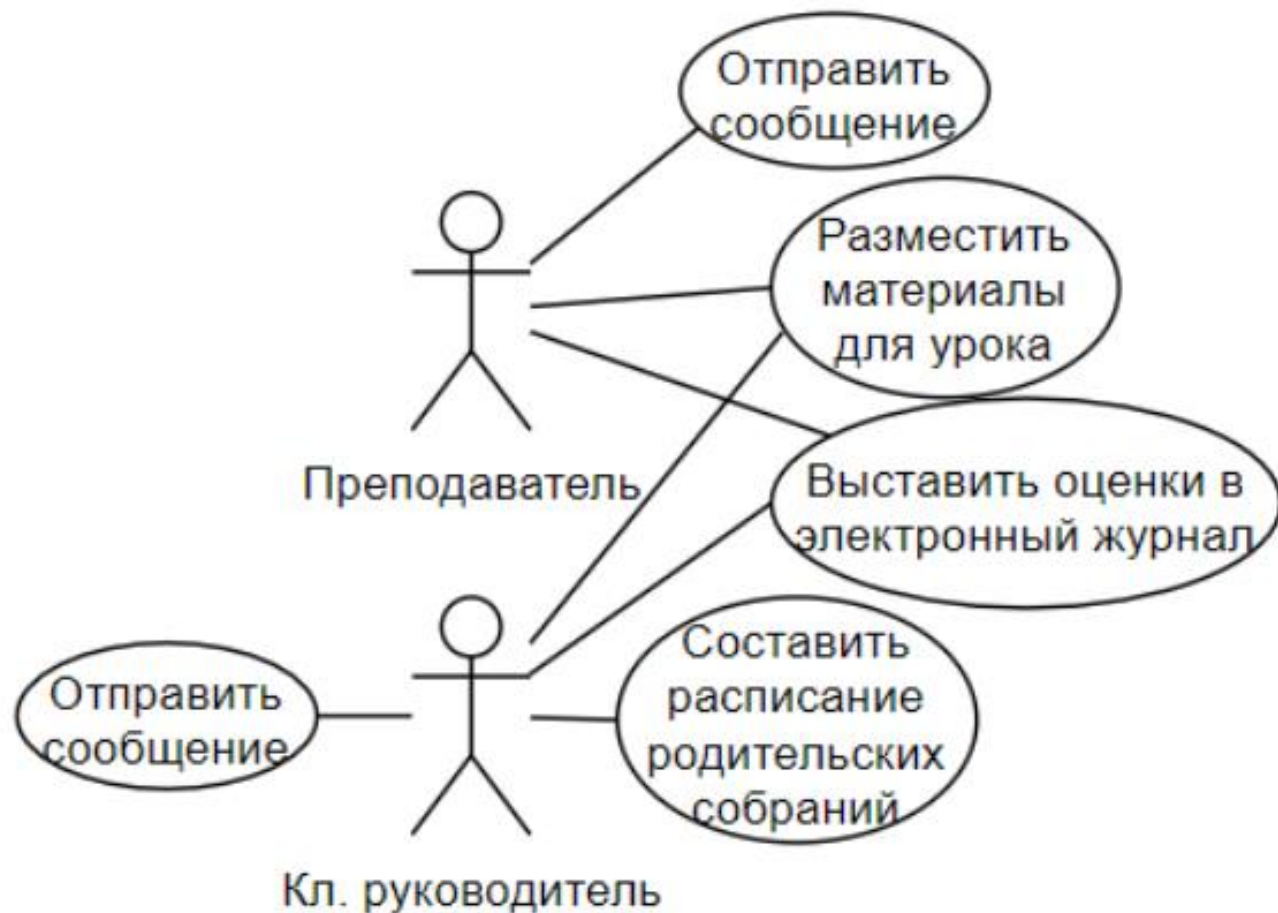
## Связи между элементами



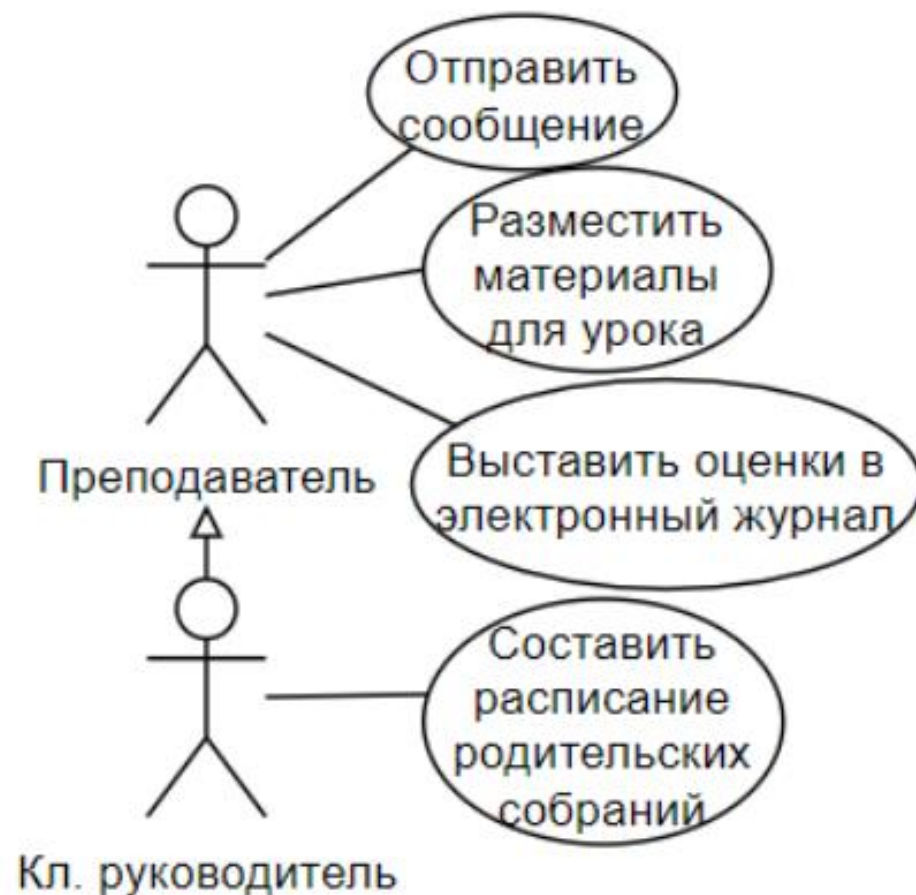




Отношение обобщения (англ. "generalization relationship")

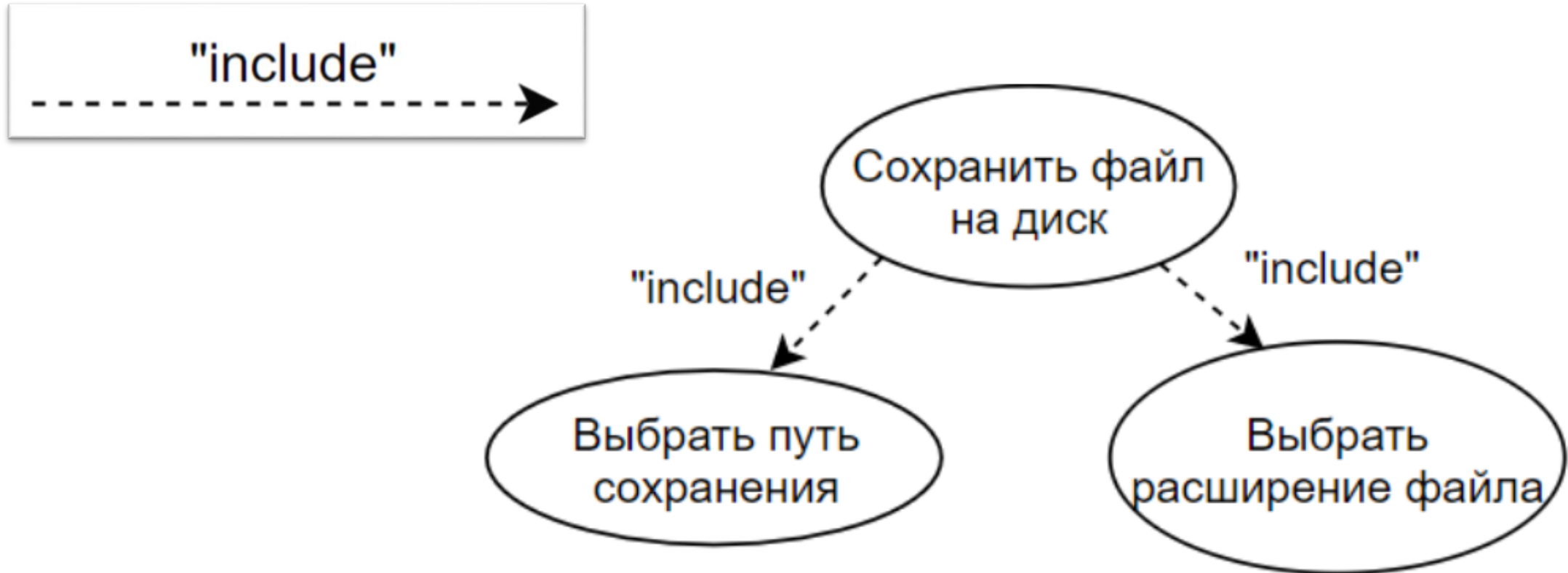


До использования отношения обобщения



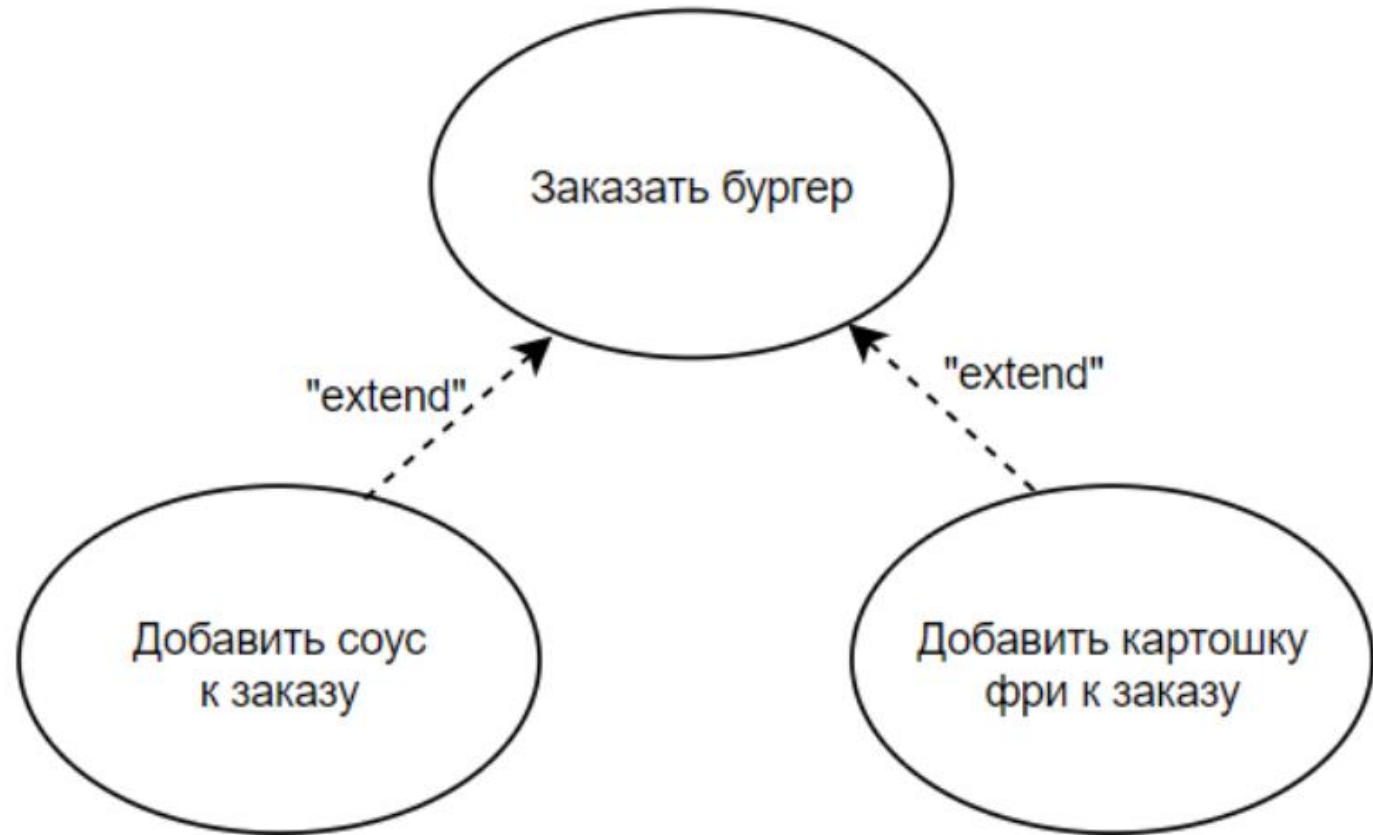
После добавления отношения обобщения

Когда мы используем отношение включения, мы подразумеваем, что составные варианты использования **ОБЯЗАТЕЛЬНО** входят в состав общего варианта использования.



Отношение включения используется для изображения составного действия

Когда мы используем отношение расширения, мы подразумеваем, что составные варианты использования **МОГУТ** входит в состав общего варианта использования.



На диаграмме предполагается, что к заказу **МОЖЕТ БЫТЬ** добавлена картошка фри или соус (необязательно)



