



Лабораторная работа «Разработка требований»

1. Высокоуровневые требования
 - Бизнес—требования
 - Основные функции
 - Системные требования
2. Пользовательские требования
3. Функциональные требования
4. Требования к данным
5. Нефункциональные требования

ЧувГУ, каф. Вычислительной техники
Доц. , к.т.н. Ржавин В.В.

Что посеешь, то и пожнешь
Из желудя вырастет дуб,
Из семени репейника — только репейник

Анализ требований. Цель анализа требований состоит в том, чтобы дать развернутое определение функциональных и нефункциональных требований к проектируемой системе. Требования определяются и специфицируются в виде "что система должна делать", но не в виде "как это реализовать" (ответы на вопросы как? будут получены на этапе проектирования). Для этого проводится подробное исследование бизнес-процессов (функций, определенных на предыдущем этапе) и информации, необходимой для их выполнения (сущностей, их атрибутов и связей (отношений)).



Шаблон требований

В данной лабораторной работе требуется разработать модель требований к проектируемой программной системе, взяв за основу предлагаемый ниже шаблон.

При решении любой проблемы всегда придерживайтесь последовательности действий:

Проблема → Цель → Задачи → Методы решения → Инструменты → Решение

А. Высокоуровневые требования

- Бизнес—требования
- Основные функции
- Системные требования

1. Введение

1.1. Назначение

В данном разделе необходимо определить продукт, требования для которого указаны в этом документе, в том числе его редакцию или номер выпуска.

1.2. Проблема

Смотрите лекционные материалы.

1.3. Решение (ставим общую цель). Цель создания программного продукта.

1.4. Бизнес-требования.

Смотрите лекционные материалы.

1.5. Бизнес-задачи. Продукт (приводится наименование продукта) предназначен для решения следующих бизнес-задач: ...

1.6. Соглашения, принятые в документах (Приложение А)

Здесь мы описываем все непонятные технические слова или термины, которые встречаются в документе. Заметьте, что описание непонятого слова не может содержать другое непонятое слово. Старайтесь расписать как можно подробнее термин, который Вы исполь-

зуете простым и понятным всем языком. Не экономьте на этой секции, потому что чем больше вы распишете непонятных вещей, тем проще будет потом проектировать.

1.7. Границы проекта

Для наглядного представления границ проекта чаще всего используют:

- контекстная диаграмма
- карта экосистемы
- дерево функций
- список событий

Но применяются и другие методы. Диаграммы вариантов использования позволяют проиллюстрировать границу между вариантами использования и действующими лицами.

1.8. Ссылки

В данной секции мы пишем ссылки на литературу, в которой можно найти основания использованных технологий и фактов. Ссылки и их описание должны быть максимально полными, чтобы в случае чего (линк умер просто) можно было найти этот материал.

2. Общее описание

2.1. Общий взгляд на продукт

Во многих случаях полезно начинать создание модели требований с составления диаграммы деятельности, на которой должны быть отображены основные задачи, выполняемые пользователем, как в системе, так и за ее пределами.

Обратите внимание на то, что диаграммы деятельности можно использовать для изображения алгоритмов, **действующих в программном обеспечении**, а вот при применении диаграмм для бизнес-процесса нужно отображать на них действия, видимые **вне** системы.

2.2. Классы и характеристики пользователей (заинтересованных лиц)

2.2.1 Сведения о пользователях

Так как требования — это желания нескольких людей, то анализ требований начинается с выявления лиц, чьи желания система должна учитывать

Пример

У системы существуют три основных пользователя: диспетчер, менеджер, мастер цеха. Менеджер – вводит данные о вновь поступивших заказах, контролирует их исполнение. Диспетчер – планирует работы цеха, следит за их исполнением. Мастер цеха – вводит данные о реальном состоянии работ (заказов).

2.2.2 Пользовательская среда

Пример

В настоящее время на предприятии имеется 2 производственных цеха (2 мастера), один диспетчер и 10 менеджеров продаж. Увеличение количества мастеров в ближайшие 8 лет – максимально 10, менеджеров – максимально 40, диспетчеров – максимально 2. Разумное время планирования одного заказа – 5 минут, для сложных заказов – не более 15 минут

2.2.3 Профили пользователей

Типичный представитель	Менеджер
Описание	Пользователь системы, наделенный правами на чтение информации и занесение данных о заказах.

Тип	Пользователь
Ответственности	Вводит данные о вновь поступивших заказах. Назначает срок исполнения заказа на основе информации, представленной в системе. Осуществляет контроль заказов.
Критерий успеха	Возможность определения реального срока исполнения заказа; получение данных о реальном состоянии заказа в производстве.

2.2.4 Ключевые потребности пользователей

Пример

Диспетчер затрачивает большое количество времени на составление очереди и внесение необходимых изменений. Менеджеры затрачивают значительное количество времени на регулирование накладок с диспетчером. Предприятие нуждается в системе, которая бы ускорила и оптимизировала вышеуказанные процессы.

3. Основные функции

Решения о том, каким образом проектируемая система будет решать проблемы заказчика, обычно сводят в список такого вида:

- У автомобиля будут автоматические стеклоподъемники
- Диаграммы динамики обнаружения неисправностей будут снабжены визуальными средствами оценки прогресса
- Необходимо предусмотреть возможность ввода заказов через Интернет
- Будет поддерживаться автоматический цикл двойной сварки

Эти описания называются **функциями** (features), **ключевыми возможностями** (key features) или **характеристиками** создаваемой системы.

Ключевая возможность — это набор логически связанных функциональных и нефункциональных требований, которые обеспечивают потребности пользователя и удовлетворяют «бизнес требованиям». В области коммерческого программного обеспечения «ключевая возможность» представляет собой узнаваемую всеми заинтересованными лицами группу требований, которые важны при принятии решения о покупке – элемент маркированного списка в описании продукта.

В отличие от бизнес-требований, которые описывают цели организации, ключевые возможности на высоком уровне абстракции **описывают функциональность и характеристики качества разрабатываемой информационной системы, т. Е. являются высокоуровневыми требованиями к системе.**

Желательные характеристики продукта, которые перечисляет клиент, не эквивалентны тем, что входят в список необходимых для решения задач пользователей.

В качестве примеров характеристик продуктов можно привести избранные страницы или закладки веб-браузера, средства проверки орфографии, запись макрокоманды, автоматическое обновление определений вирусов в антивирусной программе.

Характеристики могут охватывать множество пользовательских требований, и для каждого варианта необходимо, чтобы множество функциональных требований было реализовано для выполнения пользователем его задач.

1. Назначаем уникальный идентификатор.
2. Описываем детально характеристику (features) продукта. Для чего она? Что должна делать? Какой у нее приоритет выполнения? Из этого раздела человеку должно сразу стать понятно, зачем этот функционал присутствует в системе.

3. Триггер запуска features. Когда она запускается и как себя ведет при запуске? Например, HTML редактор показывается при нажатии пользователя на ссылку меню *Открыть HTML редактор*
4. Подробное и детальное описание характеристики. Описываем все: как работает, как реагирует на ошибки, что должно проверять, как отображать данные, как и куда что сохраняет и т.д.

Пример

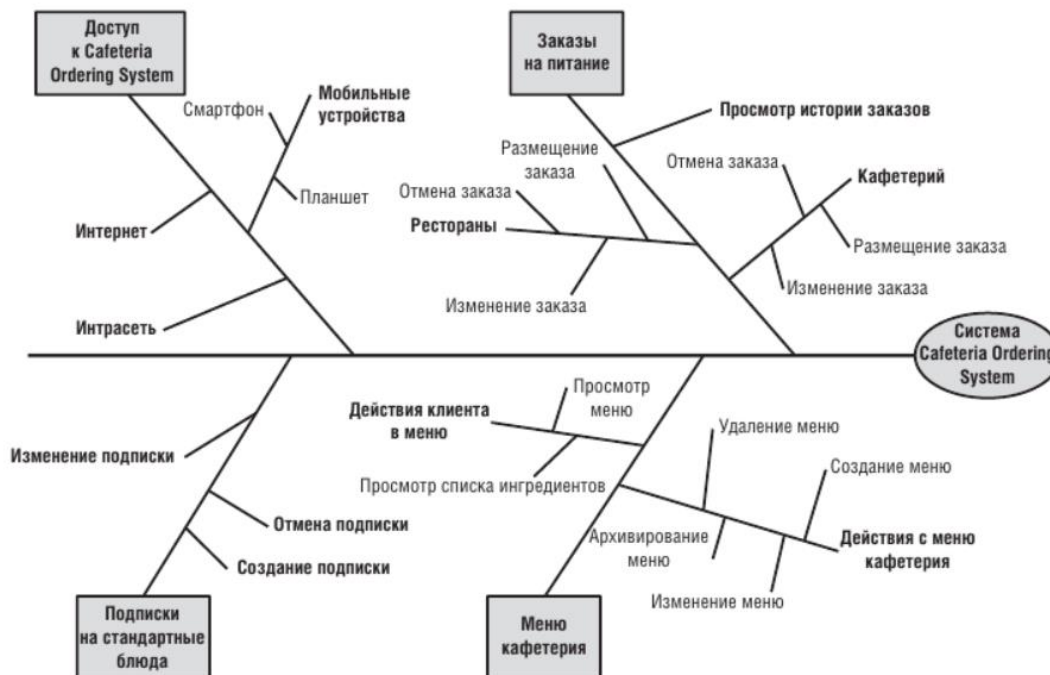
FE-1 Заказ и оплата блюд из меню кафетерия для получения в кафетерии или с доставкой.

FE-2 Заказ и оплата блюд с доставкой из близлежащих ресторанов.

FE-3 Создание, просмотр, изменение и удаление одинарной или регулярной заявок на питание или на ежедневные специальные блюда.

FE-4 Создание, просмотр, изменение и удаление меню кафетерия.

...



Частичное дерево функций системы Cafeteria Ordering System.

Необходимо также отметить, что **features** обладают определенным дуализмом в своей интерпретации, зависимым от контекста конкретного продукта – с одной стороны это может быть «тот самый список характеристик, указанный на коробке продукта» в случае создания «коробочного ПО». С другой стороны, это может список высокоуровневых возможностей системы, например, при заказной разработке ПО автоматизации бизнес-процессов конкретной организации.

Features могут быть разного уровня детализации – от выражения высокоуровневых возможностей системы (например, «Расчет заработной платы ...»), до достаточно конкретных требований (например, «Автоматическое уведомление клиента по e-mail о резервировании товара на складе»)

Здесь мы будем придерживаться интерпретации **features**, как основной функции.

При определении требований важно специально установить не только функции, но и атрибуты, ограничения, предпочтения и ожидания продукта. Обычно в процессе получения информации функции определяются первыми. **Функции** описывают то, что продукт собирается выполнить. Также важно определить **атрибуты** продукта. Атрибуты являются характеристиками, желаемыми клиентом, и, хотя два продукта могут иметь схожие функции, они могут иметь совершенно разные атрибуты. После того, как все атрибуты были выяснены и прикреплены к функциям, мы должны определить **ограничения** для каждого из атрибутов. **Предпочтения**, которые являются желательным, но необязательным условием для атрибута, также могут быть определены в дополнение к его ограничениям. Наконец, мы должны определить ожидания клиента.

4. Системные требования

- Требования к архитектуре системы. Например, число и размещение хранилищ и серверов приложений.
- Требования к параметрам оборудования. Например, частота процессоров серверов и клиентов, объём хранилищ, размер оперативной и видеопамати, пропускная способность канала и т.д.
- Требования к параметрам системы. Например, время отклика на действие пользователя, максимальный размер передаваемого файла, максимальная скорость передачи данных, максимальное число одновременно работающих пользователей и т.д.
- Требования к программному интерфейсу.
- Требования к структуре системы. Например, масштабируемость, распределённость, модульность, открытость.
- Требования по взаимодействию и интеграции с другими системами.
- Требования к режимам функционирования системы.
- Требования к персоналу.
- Требования к надёжности.
- Требования к безопасности;
- Требования к эргономике и технической эстетике.
- Требования к транспортабельности (для подвижных АС).
- Требования к защите информации от несанкционированного доступа.

В. Пользовательские требования

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. К отличным способам представления этого вида требований относятся **варианты использования, пользовательские истории и таблицы «событие — отклик»**. В идеале эту информацию предоставляют реальные представители пользователей. Пользовательские требования описывают, что пользователь должен иметь возможность делать с системой.

С. Функциональные требования

Функциональные системные требования, как более детализированное описание пользовательских требований. Они обычно служат основой для заключения контракта на разработку программной системы и поэтому должны представлять максимально полную спецификацию системы в целом.

Пользовательское требование декомпозируется на какое-то число функциональных. К каждому функциональному требованию нужно продумать нефункциональные требования и ограничения.

Пример

Пользовательское требование

- “...система должна обеспечить возможность ввода информации о заказе, с последующим ее сохранением и возможностью передачи информации о заказе к любому сотруднику...”

Функциональные требования

- 1) “...необходимо разработать диалоговые средства (Web-интерфейс) для работы пользователя с данными о заказах на предприятии,
- 2) ... для каждого заказа данные могут быть: введены, отредактированы, удалены и / или сохранены в БД системы, с возможностью доступа для них одного или нескольких пользователей ...”

D. Требования к данным

1. Концептуальная (или логическая) модель данных

2. Словарь данных

Словарь данных описывает каждый поток данных, определенный в системе. Пример описания декомпозиции сложных потоков данных на составные части приведен ниже:

ПРЕДМЕТЫ = МОНЕТЫ + НЕ_ДЕНЬГИ

МОНЕТЫ = [РУБЛИ | КОПЕЙКИ | ГРИВЕННИКИ]

ТОВАРЫ = [ШОКОЛАДКИ | КОНФЕТЫ | ЛИМОНАД]

В словаре данных определяются все потоки данных и управления, а также хранилища данных и управления. Внешние сущности не описываются в словаре данных ввиду того, что они принадлежат окружению системы, а не самой системе. В словаре присутствуют описания двух типов потоков - примитивов и составных потоков. Примитивы в отличие от составных потоков не определяются через другие потоки и примитивы.

Примитивы определяются посредством указания их атрибутов. Существует два принципиально различных типа примитивов: дискретные и непрерывные, имеющие соответственно различные множества атрибутов. Примитив является дискретным, если он может принимать в каждый конкретный момент времени одно из конечного множества значений. Отличительной чертой непрерывных примитивов является то, что множество значений определяется некоторым диапазоном значений и не ограничено.

Элемент данных	Описание	Структура или тип данных	Длина	Значения
----------------	----------	--------------------------	-------	----------

Пример:

Элемент данных	Описание	Структура или тип данных	Длина	Значения
указания по доставке	куда и кому должно быть доставлено блюдо, если его не забирают в кафетерии	имя клиента + телефон клиента + дата доставки блюда + пункт назначения поставки + временной интервал доставки		
пункт назначения поставки	строение или комната, куда нужно доставить заказанное блюдо	алфавитно-числовое значение	50	разрешены дефисы и запятые

3. Отчеты

Если приложение будет генерировать отчеты, перечислите их здесь и опишите их характеристики. Если отчет должен соответствовать определенному готовому макету, можно указать это как ограничение, возможно с примером. В противном случае сосредоточьтесь на логических описаниях, порядке сортировки, уровнях суммирования и т.п., отложив подробный макет до этапа дизайна.

Пример:**Отчет о заказанных блюдах**

Характеристика	Описание
<i>Идентификатор отчета</i>	COS-RPT-1
<i>Заголовок отчета</i>	История заказов блюд
<i>Цель отчета</i>	Клиент хочет увидеть список всех блюд, которые он раньше заказывал в кафетерии или локальных ресторанах за определенный период времени вплоть от полугода до текущей даты, чтобы можно было повторно заказать понравившееся блюдо
<i>Приоритет</i>	Средний
<i>Пользователи отчета</i>	Постоянные клиенты кафетерия
<i>Источники данных</i>	База данных о ранее размещенных заказах блюд
<i>Частота и использование</i>	Отчет генерируется по запросу клиента. Данные в отчете статичны. Отчет отображается в окне веб-браузера пользователя на компьютере, планшете или смартфоне. Его можно распечатать, если устройство под-

	держивает печать
<i>Время доступа</i>	Готовый отчет должен отображаться в течение 3 секунд после его запроса
<i>Визуальный макет</i>	Альбомная ориентация
<i>Верхний и нижний колонтитулы</i>	Верхний колонтитул должен содержать заголовок отчета, имя клиента и заданный диапазон дат
	При печати в нижнем колонтитуле должен содержаться номер страницы

4. Получение, целостность, хранение и утилизация данных

Если это важно, опишите, как получают и обслуживают данные. Укажите все требования, относящиеся к защите целостности данных системы. Укажите все процедуры, которые могут потребоваться, например резервное копирование, создание контрольных точек, зеркальное отображение или проверка корректности данных. Укажите политики, которые должна применять система для хранения или утилизации данных, в том числе временных данных, метаданных, остаточных данных (таких как удаленные записи), данных в кеше, локальных копий, архивов и промежуточных архивов.

Е. Нефункциональные требования

1. Требования к внешним интерфейсам

- Пользовательские интерфейсы
- Интерфейсы ПО
- Интерфейсы оборудования
- Коммуникационные интерфейсы

2. Атрибуты качества

- Удобство использования
- Производительность
- Безопасность
- Техника безопасности
- Требования к производительности
- Требования к сохранности (данных)
- Требования к качеству программного обеспечения
- Требования к безопасности системы
- Требования на интеллектуальную собственность

3. Бизнес-правила

4. Ограничения

- Язык программирования, база данных
- Стандарты кодирования
- Стандарты обмена данными
- Ограничения, накладываемые операционной средой, включая операционную систему, интерфейс пользователя и аппаратную платформу
- Ограничения, которые могут быть наложены бизнес-логикой проекта



ЗАДАНИЯ

Общие рекомендации

- При создании документов требований применяйте эффективные приемы и средства управления версиями, чтобы все читатели четко понимали, какую версию они читают в тот или иной момент времени.
- Ведите журнал изменений, в котором фиксируются суть изменений, автор, дата и причина.
- Иногда фрагмент информации логически подходит для нескольких разделов шаблона. Выберите один раздел и используйте именно его для информации такого типа в своем проекте. Не дублируйте информацию в нескольких разделах, даже если логически она ложится в эти разделы.
- При разработке требований активно пользуйтесь шаблонами, приведенными в Приложении.

А. Высокоуровневые требования

- Бизнес—требования
- Основные функции
- Системные требования

1. ВВЕДЕНИЕ

- 1.1. Кратко опишите назначение системы.
- 1.2. Для иллюстрации работы системы приведите диаграммы деятельности.
- 1.3. Опишите своими словами как вы понимаете проблему. Описание завершите предлагаемым шаблоном.

Элемент	Описание
Проблема	[Описание проблемы]
воздействует на	[Лица, на которых она влияет]
результатом чего является	[Каково влияние этой проблемы на заинтересованных лиц и деятельность организации]
Выигрыш от	[Предлагаемое решение]
может состоять в следующем:	[1. Преимущество предлагаемого решения №1; 2. ... 3. Преимущество предлагаемого решения №n.]

- 1.4. **Цель.** Сформулируйте основную цель создания программного продукта.
- 1.5. **Бизнес-требования.** Бизнес-требованиями создания продукта (приводится наименование продукта) являются (формулируются в количественном и измеряемом виде):

Идентификатор	Статус	Версия	Бизнес-требование
---------------	--------	--------	-------------------

BR-xx		1.0	

1.6. **Бизнес-задачи.** Продукт (приводится наименование продукта) предназначен для решения следующих бизнес-задач:

Идентификатор	Статус	Версия	Бизнес-задача
BT-xx		1.0	

1.7. **Основные функции системы.** Определите основные функции системы

Идентификатор	Статус	Версия	Характеристика
FE-xx		1.0	

1.7.1. По возможности для каждой функции определите

- Атрибуты
- Ограничения
- Предпочтения и ожидания

1.8. **Системные требования.** Определите системные требования

№	Иден.	Приор.	Требование
1	СТ1	средний	Система должна фиксировать ...
2	СТ2	средний	Система должна иметь ...
3	СТ3	низкий	Система должна иметь четкое ограничение ...

1.9. **Соглашения, принятые в документах.**

1.10. **Границы проекта.** Тут желательно разместить контекстную DFD-диаграмму, которая покажет общее взаимодействие системы.

1.11. **Ссылки.**

2. ОБЩЕЕ ОПИСАНИЕ

2.1. **Классы и характеристики пользователей (заинтересованных лиц)**

2.2. **Сведения о пользователях.** Дайте краткую характеристику основным пользователям системы.

2.2.1 **Пользовательская среда.** При необходимости опишите пользовательскую среду.

2.2.2 **Профили пользователей.** Составьте профиль для каждого пользователя.

Типичный
представитель
Описание
Тип
ответственности
Критерий успеха

В. Пользовательские требования

№	Иден.	Приор.	Предок	Название
1	ПТ1.	высокий	БТ-1	«Пользователь» должен иметь возможность ...
2	ПТ2.	средний		
3	ПТ3.	средний		

* Замените абстрактного «Пользователь» реальным объектом из вашего списка заинтересованных лиц.

С. Функциональные требования

Требуется в качестве примера одно произвольное пользовательское требование декомпозировать на какое-то число функциональных требований.

Д. Нефункциональные требования

- Требования к внешним интерфейсам
- Атрибуты качества
- Бизнес-правила
- Ограничения

Е. Требования к данным

Е.1. Концептуальная (или логическая) модель данных. Приведите концептуальную ER-диаграмму.

Е.2. Словарь данных. Приведите несколько примеров описания декомпозиции сложных потоков данных на составные части. См. «Приложение Д: Потоки данных».

Е.3. Отчет. Приведите перечень требуемых отчетов.

№	Иден.	Приор.	Название
1	RPT-1	высокий	Средняя успеваемость студентов ИВТ по группам в 2024 году
2			
3			

Для одного из отчетов (по вашему выбору) опишите его характеристики согласно приведенной выше таблице «Отчет о заказанных блюдах».



ТРЕБОВАНИЯ К ОТЧЕТУ

1. Краткое описание предметной области
2. Диаграммы деятельности, отражающие основные процессы предметной области
3. Проблемы предметной области
4. Цель создания программного продукта

5. Высокоуровневые требования
 - a. Бизнес—требования
 - b. Основные функции
 - c. Системные требования
6. Пользовательские требования
7. Функциональные требования
8. Требования к данным
 - a. Концептуальная (или логическая) модель данных
 - b. Словарь данных
 - c. Перечень требуемых отчетов
9. Нефункциональные требования



КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Приведите классификация требований по К.Вигерсу
2. Приведите классификация требований по Д. Леффингуэллу
3. Приведите примеры ограничений
4. Назовите основные атрибуты качества
5. Какие средства используют для демонстрации границ системы?
6. Назовите методы выявления требований
7. Перечислите задачи, которые решаются на стадии анализа требований.
8. Что такое «управление требованиями»?
9. В чем заключается анализ проблемы?
10. Какие виды ограничений на создаваемое ПО необходимо выявить в процессе работы над требованиями?
11. Каковы существующие методы выявления требований к ПО?
12. Что надо сделать для обеспечения тестируемости требования?
13. Зачем вводится приоритетность требований? Какие уровни приоритетов вы знаете?
14. Сравните процессы формирования требований и анализе требований. Чем они схожи? В чем отличаются? Поясните причины отличий.
15. Поясните методы, применяемые при аттестации требований.
16. Поясните основные шаги по работе с требованиями (выявление, анализ, ...)
17. Что такое верификация и валидация?



ИСТОЧНИКИ

1. К. Вигерс. Разработка требований к программному обеспечению.
2. Моделирование требований пользователей
<https://docs.microsoft.com/ru-ru/visualstudio/modeling/model-user-requirements?view=vs-2015&redirectedfrom=MSDN>

3. Как писать функциональные требования <https://habr.com/ru/company/retailrocket/blog/431572/>

ПРИЛОЖЕНИЕ

Приложение А. Словарь терминов

Термин, сокращение	Описание
Лицевой счет	– это специальный счёт для ведения учета расчетов с физическими и юридическими лицами, на котором отражаются все финансово-кредитные операции с определенным клиентом.

Приложение Б. Бизнес-правила

Рассмотрим различные типы правил, с которыми можно столкнуться:

1. **Факты** — это всего лишь верные утверждения о бизнесе. Факты также называют *инвариантами* – неизменными истинами о сущностях и их атрибутах.
 - *Со стоимости доставки налог с продаж не берется.*
 - *Стоимость билетов не возвращается, если покупатель изменяет маршрут.*
2. **Ограничения** определяют, какие операции может выполнять система и её пользователи. Часто применяемые слова: *может, не может, должен, только*.
 - *При отгрузке заказа менеджер должен запросить у бухгалтера товарно-транспортную накладную и счет-фактуру.*
3. **Активаторы операций** – правило, при определенных условиях приводящих к какому-либо действию. Выражение вида «Если <некоторое условие верно или наступило определенное событие>, то <что-то произойдет>», — это ключ, который описывает активатор операции.
 - *Если заказанный товар отсутствует на складе, то заказ передается в производственный отдел.*
4. **Вычисления.**
 - *Цена единицы товара снижается на 10% при заказе от 6 до 10 единиц, на 20% — при заказе от 11 до 20 единиц и на 30% — при заказе свыше 20 единиц;*
5. **Выводы.** Вывод создает новый факт на основе других фактов или вычислений. Выводы зачастую записывают в формате «если — то», применяемом также при записи бизнес-правил, активирующих операции; тем не менее, раздел «то» вывода заключает в себе факт или предположение, а не действие.

Пример

Идентификатор	Определение правила	Тип правила	Статичное или динамическое	Источник
BR-1	Периоды доставки — это 15-минутные интервалы, начинающиеся каждую четверть часа	Факт	Динамическое	Менеджер кафетерия
BR-2	Доставка всех заказов должна быть завершена между 10:00 и 14:00 по местному времени	Ограничение	Динамическое	Менеджер кафетерия

Приложение В: Уровни и типы требований

Из-за такого большого числа разных типов информации требований нам требуется единый набор описаний, изменяющих слишком перегруженный термин требование. В этом разделе приводятся определения, которые будут использоваться для терминов, наиболее часто применяемых в такой сфере, как разработка требований.

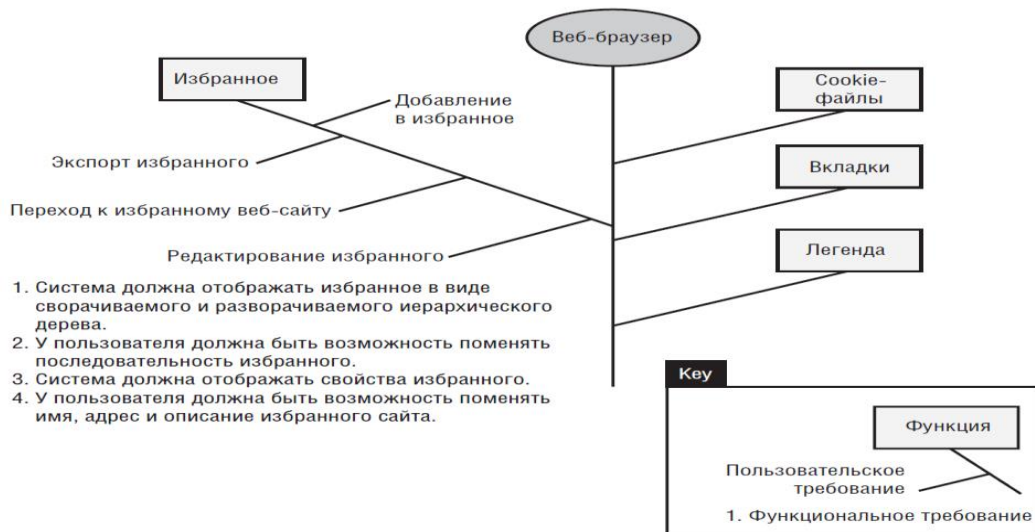
Табл. П-1. Информация о некоторых типах требований

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 (функция)	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

Сложной задачей при выявлении требований является идентификация их типов. На практике, особенно при проведении интервью, требования, формулируемые пользователем, могут относиться к различным типам требований. Существуют следующие рекомендации о том, как «понимать» пользователей и классифицировать требования:

- ☐ **Бизнес требования** - любая информация, описывающая рынок или какие-либо интересы бизнеса.
- ☐ **Ключевые возможности** - предположения пользователя о выполняемых системой задачах.
- ☐ **Вариант использования** - пользователь говорит о том, что ему необходимо что-то сделать.
- ☐ **Характеристика качества** - необходимые характеристики системы, насколько хорошо работает система.

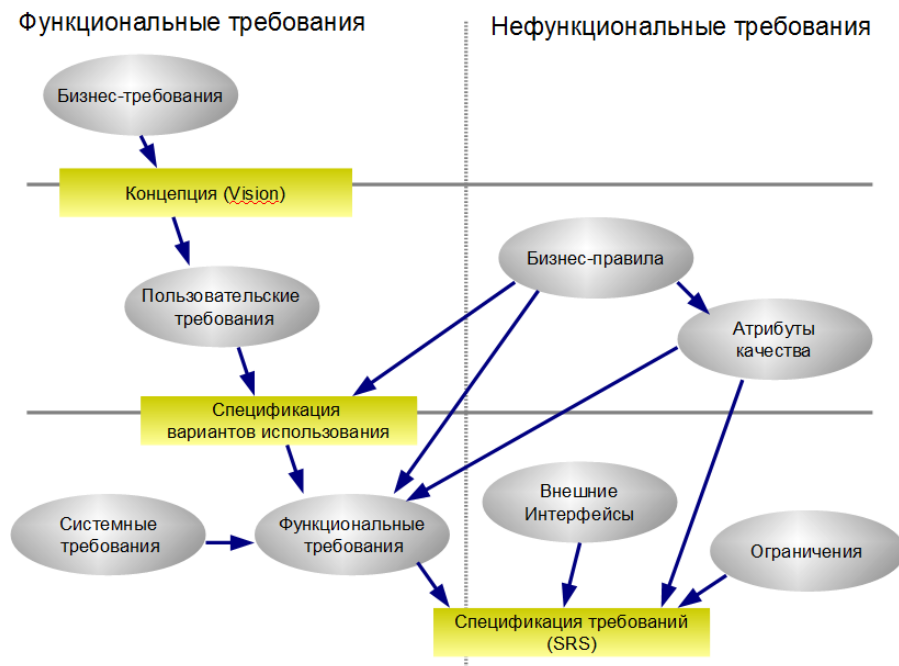
- ❑ **Ограничения** - пользователь говорит о том, что система или функция должна соответствию стандарту или правилу, реализована на определенной платформе.
- ❑ **Функциональные требования** - передаваемая информация, допустимые значения данных, поведенческие характеристики системы.



Взаимоотношения между функциями, пользовательскими и функциональными требованиями

Рисунок иллюстрирует дерево функций (feature tree) — модель анализа, которая показывает, как функцию можно разложить на иерархию более мелких функций, которые связаны с конкретными пользовательскими требованиями и ведут к определению наборов функциональных требований.

Классификация требований по Вигерсу



Иерархия требований



Приложение Д: Поток данных

Потоки, хранящиеся в словаре, могут быть:

- простыми или групповыми;
- внутренними или внешними;
- потоками данных или потоками управления;
- непрерывными или дискретными.

Атрибуты потока данных:

- имена-синонимы потока данных в соответствии с узлами изменения имени;
- БНФ-определение;
- единицы измерения потока;
- диапазон значений;
- список значений;
- список номеров диаграмм различных типов;
- список потоков;
- комментарий.

@ИМЯ = ВВЕДЕННАЯ КРЕДИТНАЯ КАРТА

@ТИП = управляющий поток

@БНФ = /указывает, что кредитная карта введена/

@ИМЯ = ДАННЫЕ КЛИЕНТА

@ТИП = поток данных

@БНФ = ФИО + адрес + телефон + ИНН

@ИМЯ = СООБЩЕНИЕ

@ТИП = поток данных

@БНФ = e-mail ! факс ! письмо

@ИМЯ = ДЕНЬГИ

@ТИП = дискретный поток

@БНФ = /деньги, выдаваемые клиенту/

@ЕДИНИЦА ИЗМЕРЕНИЯ = доллар

@НОРМА = 5...1000

@КОММЕНТАРИЙ Сумма выдаваемых денег должна делиться на 5

Приложение Е: Шаблоны требований

Каждому или большинству требований можно назначить шаблон. Шаблон имеет некоторый идентификатор, скажем номер. Т. е. все требования со схожим шаблоном будут иметь соответственно тот же самый номер шаблона и будут выглядеть структурно одинаковыми.

■ **Типичное требование**

<Тип пользователя> должен иметь возможность <описание возможности>

■ **Требование с ограничениями и условиями**

<Тип пользователя> должен иметь возможность <описание возможности> с <показатель производительности> от <момент отсчета>, находясь в <условия эксплуатации>

Оператор должен иметь возможность произвести выстрел в течение 3 секунд с момента обнаружения цели радаром, находясь в сложных морских условиях.

■ **Требование - ограничение**

<Тип пользователя> не должен попадать под действие <соответствующее законодательство>

■ **Системное требование**

<Система> должна <выполняемая функция> не менее чем <количество> <объект> функционируя в <условия эксплуатации>

Телекоммуникационная система должна обеспечивать телефонную связь не менее чем с 10 абонентами, функционируя в условиях отсутствия источника внешнего электрического питания

■ **Периодическое требование**

<Система> должна <выполняемая функция> <объект> каждые <показатель производительности> <единица измерения>

Кофе-машина должна производить горячий напиток каждые 10 секунд

Примеры шаблонов требований с ограничением

■ **Производительность/возможность**

<Система> должна <выполняемая функция> <объект> не менее чем «производительность» раз в <единица измерения>

■ **Производительность/возможность_**

<Система> должна <выполняемая функция> <объект> типа <характеристика> в течение <производительность> <единица измерения>

■ **Производительность/мощность**

<Система> должна <выполняемая функция> не менее чем <количество> <объект>

■ **Производительность/своевременность**

<Система> должна <выполняемая функция> <объект> в течение <производительность> <единица измерения> с момента <событие>

■ **Производительность/периодичность**

<Система> должна <выполняемая функция> не менее чем <количество> <объект> в течение <производительность> <единица измерения>

■ **Способность к взаимодействию/мощность**

<Система> должна <выполняемая функция> <объект> состоящий из не менее чем <производительность> <единица измерения> с <внешняя сущность>

■ **Устойчивость/периодичность**

<Система> должна <выполняемая функция> <объект> с <производительность> <единица измерения> каждые <производительность> <единица измерения>

■ **Окружение/работоспособность**

<Система> должна <выполняемая функция> <объект> функционируя в <условия эксплуатации>

Приложение 3: Требования и спецификации

Существует четкое различие между требованиями и спецификациями.

Требование — это условие, необходимое пользователю для решения проблемы или достижения цели.

Спецификация — это документ, в котором в полной, точной и поддающейся проверке форме указываются требования, дизайн, поведение или другие характеристики системы, а зачастую и процедуры определения того, были ли выполнены эти положения.

Например, **требование** к автомобилю может заключаться в том, чтобы максимальная скорость составляла не менее 120 миль в час. Спецификация для этого требования будет включать техническую информацию о конкретных аспектах проектирования. Другой термин, который обычно встречается в книгах и бумагах, — это **спецификация требований**, которая представляет собой документ, который определяет требования к системе или компоненту. Включает функциональные требования, требования к производительности, требования к интерфейсу, требования к дизайну и стандарты разработки. **Таким образом, спецификация требований — это просто требования, записанные на бумаге.**

Спецификация — это требование, которое прошло технико-экономическое обоснование и готово к реализации. Это требование, которое дошло до стадии проектирования.

Другими словами:

Требование — это поведение (или не поведение) «как запланировано» или «как желаемо».

Спецификация — это поведение (или не поведение), которое «должно быть построено» или «как построено».

Пример:

требование: 1. пользователь нажимает кнопку ОК 2. система печатает счет

спецификация: 1. пользователь нажимает кнопку ОК 2. система печатает счет

Как видите, содержание обоих может быть одинаковым. Разница в том, что требование является артефактом анализа. Спецификация является артефактом дизайна.

В окончательной готовой документации вы обычно найдете слово «спецификация» вместо «требование», поскольку требования были преобразованы в спецификации.

Таким образом, спецификация — это документ, полный требований, плюс некоторая другая вспомогательная и дополнительная информация.

При определении требований важно специально установить функции, атрибуты, ограничения, предпочтения и ожидания продукта. Обычно в процессе получения информации функции определяются первыми. **Функции** описывают то, что продукт собирается выполнить. Также важно определить **атрибуты** продукта. Атрибуты являются характеристиками, желаемыми клиентом, и хотя два продукта могут иметь схожие функции, они могут иметь совершенно разные атрибуты. После того, как все атрибуты были выяснены и прикреплены к функциям, мы должны определить **ограничения** для каждого из атрибутов. **Предпочтения**, которые являются желательным, но необязательным условием для атрибута, также могут быть определены в дополнение к его ограничениям. Наконец, мы должны определить ожидания клиента.