



# Управление требованиями

Введение в системный анализ



# Оглавление

<b>Введение</b>	<b>2</b>
<b>Термины, используемые в лекции</b>	<b>2</b>
<b>Общий принцип управления требованиями на основе понятия энтропии</b>	<b>3</b>
<b>Выявление требований</b>	<b>7</b>
<b>Согласование требований</b>	<b>15</b>
<b>Документирование требований</b>	<b>18</b>
<b>Управление требованиями</b>	<b>20</b>
<b>Контроль требований</b>	<b>23</b>
<b>Что можно почитать еще?</b>	<b>26</b>
<b>Используемая литература</b>	<b>26</b>

## Введение

Мы продолжаем цикл лекций по технической специализации «Системный анализ». В предыдущих лекциях мы познакомились с понятием системы, которая является объектом труда системного аналитика. Мы познакомились с приемами и методами системного анализа, его основными инструментами и подходами. Изучили понятие информации, и те особенности, которые существуют в информационных системах. Так же мы ознакомились как на практике управлять качеством информации.

На текущей лекции определим что такое требования в применении к системному анализу, и как требования помогают делать системы лучше, узнаем какими бывают требования, изучим процесс формирования требований и жизненный цикл требований. Научимся поддерживать требования в актуальном состоянии

## Термины, используемые в лекции

**Требования к системе** – это такие знания о системе которые позволяют воссоздать моделируемый объект с ожидаемой точностью.

**Управлением требованиями к системе** – это управление жизненным циклом знаний о системе для гарантированного достижения цели её создания.

**Управление требованиями** – это процесс формирования и изменения требований в целях снижения рисков и издержек процесса создания и развития системы.

## **Общий принцип управления требованиями.**

Давайте с вами разберем общие принципы управления требований на основе понятия энтропии.

Прежде чем приступить к изучению требований к проектируемым системам, как предмета труда системного аналитика, вернемся на шаг назад и ещё раз закрепим роль информации в современном мире.

Отправной точкой научного изучения информации по праву считается теория информации, которая появилась более полувека назад благодаря труду Клода Шеннона. Она достаточно сложна для неподготовленного восприятия, но в ней наряду с информационной определённой и упорядоченностью фигурирует достаточно простое, но ценное для целостного восприятия понятие энтропии.

Как правило, информационная энтропия - это среднее количество информации, передаваемой событием, при рассмотрении всех возможных исходов.

Энтропия это характеристика степени неопределённости состояний системы или проще говоря хаоса. Чем выше степень упорядоченности системы или чем проще, меньше и ограниченной система, чем ниже показатель её энтропии.

По мнению современников Клода Шеннона - ученых внесших значительный вклад в развитие математики и кибернетики Эрвина Шрёдингера и Ноберта Винера, энтропия это и есть основной показатель дезорганизованности системы.

Таким образом, в 1948 году произошел настоящий научный взрыв и появилась не только информатика, но и термин системный анализ в публикациях американской корпорации RAND (Research and Development – «Исследования и разработки»)



Эрвин  
Шредингер  
1948

Упорядоченность,  
неупорядоченность  
и энтропия



Клод  
Шеннон  
1948

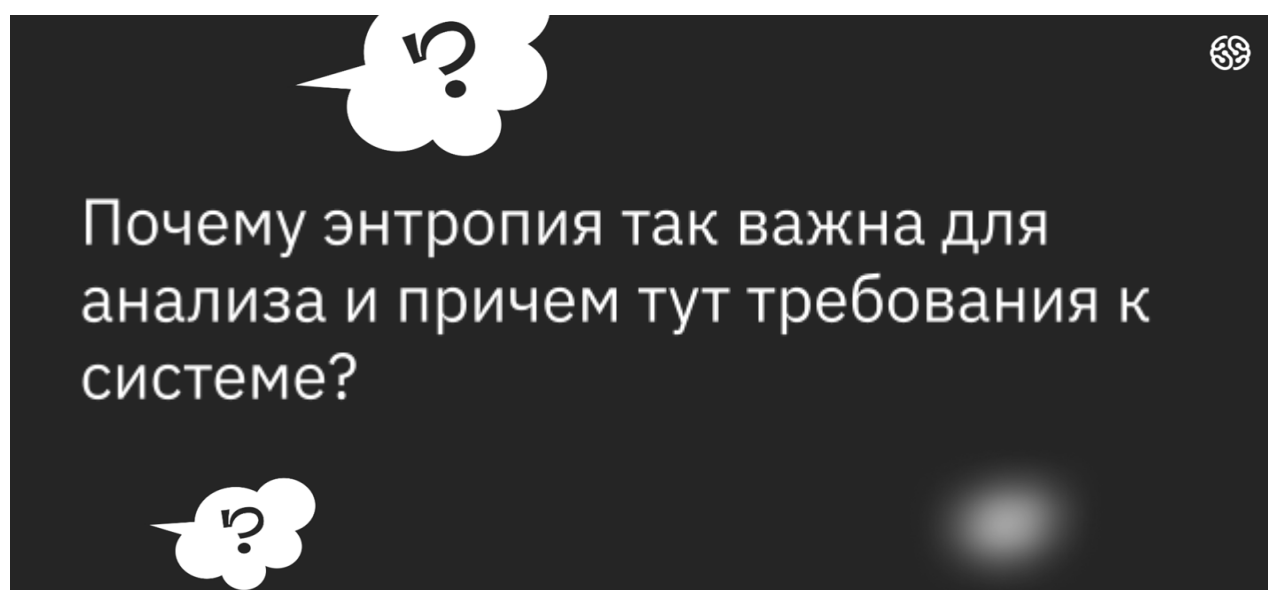
Математическая  
теория связи  
(Теория информации)



Ноберт  
Винер  
1948

Кибернетика

Почему же энтропия так важна для системного анализа? Попробуйте порассуждать ища ответ на этот вопрос.



Ответ: Изначально сталкиваясь с незнакомой системой её можно отождествлять с чёрным ящиком, да еще и в темной комнате, наполненной малознакомыми людьми и другими системами. Не правда ли зловещая история? Обычно в таких случаях зовут частного детектива. О да, он уже здесь, это вы!

Кстати, говоря о детективной работе системного аналитика, можно заметить что она во многом похоже на работу сыщика. Также как метод дедукции, применяемый известными детективами, служит для логического объяснения факта на основе теории, метод индукции служит построению теории на основе отдельных фактов, так и метод анализа, применяемый системным аналитиком, позволяет выделять и описывать отдельные элементы систем, а метод синтеза формулировать общее представление на основе частных сведений.

Итак, о системе ещё ничего не известно, также неизвестно о связях системы с внешним миром с ресурсами, потребителями и с другими системами. Возникшая неопределенность в самом начале изучения системы с точки зрения системного аналитика – это и есть энтропия её информационной модели. Но не стоит бояться, даже если в руках всего лишь чистый лист бумаги. С каждым новым фактом о системе повышается информационная определённость её модели. По мере роста детализации энтропия снижается. Так происходит до тех пор, пока система не станет в должной мере описанной. Но затем с развитием системы весь процесс повторяется вновь.

Таким образом, системный аналитик проектируя модель системы, изначально работает над снижением её энтропии, а в дальнейшем по мере развития системы, следит чтобы возрастающая сложность системы не привела к росту энтропии. Это может привести к гибели системы за счет абсолютной невозможности проверить какие-либо изменения.

Что плохого в излишней энтропии, почему она так губительна?

Можно привести одну доступную аналогию: на автомобильном производстве транспортное средство изготовлено по строгой модели, энтропия автомобиля сравнительно низка, каждый узел описан и многократно проверяется, но в процессе эксплуатации двигатель и подвеска получают повреждения: двигатель начинает потреблять больше топлива, подвеска греметь.



**На заводе автомобиль изготавливается по документации**

Экземпляр автомобиля при производстве многократно проверяется – это называется контроль качества

Энтропия низка



**При эксплуатации автомобиль получает повреждения**

Постепенно подержанный автомобиль преподносит все больше сюрпризов владельцу

Энтропия повышается



**Представьте автомобиль выпущенный без соблюдения технологического процесса**

о таком автопроизводителе говорят что он не качественный, понятно чего следует жать от такого автомобиля

Энтропия высокая

Мы говорим, у такой системы повышается энтропия. Если на производстве энтропия модели высока, мы говорим, что автомобиль некачественный, и даже, что у него может отвалиться колесо, пока он не сошел с конвейера. Системный аналитик, должен следить за энтропией модели системы, так как это в дальнейшем повлияет

на разработку, с тем фактом, что небольшая неопределенность в описании модели, будет многократно усилена при создании объекта по данной модели.

С другой стороны подобно, подобно подвеске автомобиля, энтропия требований повышает адаптивность системы. Если требования недостаточно формализованы, то это повышает возможность системы к адаптации под изменчивые условия внешней среды.

Здесь нужно отметить, что общесистемная среда значительно изменилась за последние 40 лет. Мировая бизнес-среда, влияющая на большинство проектируемых систем, с распространением глобальной сети Интернет из стабильного состояния SPOD 80-х годов прошлого века перешел в изменчивое состояние VUCA и по мнению Джамаиса Кашио начал дальнейшую качественную трансформацию в хрупкую и непредсказуемую BANI, от английских слов хрупкий, тревожный, нелинейный, непостижимый.



Ключевыми терминами становятся MVP – минимально жизнеспособный продукт, отличающийся отсутствием четких требований, но обладающий ценностью для потребителя и Legasy – наследие, нарицательное название подхода вызванного использованием устаревших технологий и как результат избыточного описания неактуальных требований.

Конечно, состояние среды зависит от сферы применения конкретной системы и возможно в госуправлении, образовании, медицине, обороне, науке остаются стабильные SPOD области, но среда для многих коммерческих систем уже находится на стадии VUCA или BANI, с высокой степенью изменчивости, что определяет более высокий целевой уровень энтропии требований, то есть более низкий уровень их определенности чем для SPOD среды.

Таким образом, аналитику требуется оценивать состояние среды с которой взаимодействует система и пристально следить за степенью определенности требований: избыточная их энтропия ухудшит качество, а недостаточная приведёт к параличу изменений.



Пристально следите за степенью определённости требований: избыточная их энтропия ухудшит качество, а недостаточная приведёт к параличу изменений

## Выявление требований

Описывая систему аналитик создает модель, так, чтобы по модели её можно было бы воспроизвести заново в соответствии с описанием, но без относительно того существует ли система в данный момент или нет.

Подготовка такого описания называется определением или спецификацией требований к системе. А управление жизненным циклом этих знаний называется управлением требованиями к системе.

Прежде всего, при определении требований для неподготовленного аналитика не обладающего знаниями о системе встает задача их выявления.

Выявление требований – первый этап процесса работы с требованиями.

Прежде всего нам следует разобраться с составом тех кто тесно связан с системой. Это первый шаг выявления требований. Включив свет в нашей воображаемой комнате мы увидим, что эти лица относительно рассматриваемой системы занимают определенное положение или как можно ещё сказать играют роли.

Важнейшая роль – потребитель или клиент. Разрабатывая систему мы всегда должны ориентироваться на её потребителя, и его требования являются важными для системного анализа, но не только его. Но давайте обо всем по порядку.

Потребители могут быть как внутренними, состоящим в штате компании выступающие заказчиком системы, так и внешними. Могут быть активными пользователями системы, а могут лишь получать результат работы системы. Вне зависимости от того какое место занимает потребитель относительно компании заказчика разработки, его слово является весомым при определении требований. А сам подход называется клиентоориентированным.

Вторая важнейшая роль – это распорядитель бюджета, или как его называют стейкхолдер (stakeholder), который буквально переводится как владелец доли компании. Сейчас же благодаря стандартам BABOK 3.0, ГОСТ 51897 от 2002 и ГОСТ 59194 от 2020 года трактовка термина расширена и теперь стейкхолдером стоит понимать любую причастную сторону, или как её еще именуют заинтересованную сторону.

Заинтересованная сторона может быть включена в проект, и быть частью команды разработки. Это могут быть руководитель и члены проектной команды, другие исполнители, владелец продукта, различные рабочие комитеты от решения которых зависит судьба системы.

Также к заинтересованной стороне можно отнести тех кто только намерен или уже пользуется результатами работы системы. Здесь нужно сказать о заказчиках её создания, о руководителях разного уровня, бизнес партнёрах и потребителях. Да потребители также являются заинтересованной стороной.

Ещё к заинтересованной стороне можно отнести тех кто не вовлечен в процесс создания и развития системы и даже не используют и не планируют использовать результаты работы системы, но в силу определённых обстоятельств имеет влияние на команду разработки и волен определять правила и выставлять требования к системе. Это различные регулирующие органы как внутри компании и за её пределами, менеджеры высокого уровня — распорядители бюджета, акционеры, аудиторы бизнес-партнеры и другие стороны вплоть до конкурентов, в общем все кто может повлиять на процесс развития или конечный результат.

У каждой заинтересованной стороны есть фамилия имя отчество и контактные данные, но так же верно утверждение, что сторонняя система или может выступать в качестве заинтересованной стороны и выдвигать свои требования.

Выявление всех заинтересованных сторон является важным этапом работы с требованиями. Допустим на этом этапе мы пропустили представителям внутренней безопасности, накладывающего ограничения на техническую реализацию системы. После того как система будет создана без учёта мнения одной из заинтересованных сторон встанет вопрос, что делать когда требования будут предъявлены при вводе в системы в эксплуатацию. Соответственно, такой пропуск заинтересованной стороны приводит к тому, что готовую систему приходится перестраивать поэтому пропуск хотя бы одного значимого стейкхолдера несет в себе серьезные риски для будущего существования системы. Но следует также заметить, что заинтересованные стороны могут варьироваться от уровня зрелости продукта.



Например, если служба информационной безопасности может быть намеренно не включена в перечень заинтересованных сторон для ускорения и облегчения процесса подготовки прототипа системы, то игнорирование вопросов информационной безопасности на последующих этапах разработки информационной системы могут нести самые серьезные последствия и риски для системы.

Если список разных сторон изначально не известен, то следует его составить используя приемы их выявления, для чего стоит провести интервьюирование известных заинтересованных лиц, задав вопросы о наличии других заинтересованных сторон, изучить нормативную базу и технические регламенты, после чего описать для каждой заинтересованной стороны отношение к рассматриваемой системе и степень влияния. Отношение к системе может варьироваться от позитивного – развивающего, до негативного – ограничивающего. Степень влияния от слабой до сильной. Распределение полученного списка определяет степень упорядочения требований при их рассмотрении.

Второй шаг выявления требований это получение или извлечение непосредственно самих требований. Для выявления требований широко используются такие приемы как: интервью, анкетирование, мозговой штурм, наблюдение, анализ нормативной документации, сопоставление с аналогами.

Интервью это один из наиболее действенных методов сбора требований. Аналитик составляет список вопросов и назначает встречу или отправляет опросник по почте. Вопросы можно разделить условно на пять групп: что, где, когда, как и зачем или почему.

Например:

- Что функционал системы должен делать?
- Что является составной частью функционала?
- Где и кем он будет использоваться?
- Когда планируется его реализовать?
- Как определить готовность функционала?
- Какие составные части функционала можно выделить?
- Могут ли составные части функционала работает независимо?
- Почему этот функционал так важен для системы?

Какие вопросы вы бы задали в интервью эксперту по системе перевода денежных средств?



## Какие вопросы вы бы задали в интервью эксперту по системе перевода денежных средств?



**Ответ:** Вопросы могут быть любые, но они должны затрагивать пять групп: что, где, когда, как и зачем.



При проведении интервью предварительно четко и однозначно определите тему и повестку встречи, а после зафиксируйте информацию в протоколе встречи

Другой метод – **анкетирование**, является важным для извлечения требований большой группы заинтересованных лиц. Зачастую бывает так, что группа заинтересованных лиц, в особенности потребителей является очень обширной, но у каждого есть индивидуальное мнение по сути требований. В этом случае аналитик подготавливает стандартную анкету с вариантами ответов на вопросы интервью, направляет её для заполнения, после чего статистический обрабатывает результаты выявляя среднее значения и анализируя отклонения. К анализу отклонений следует подходить очень внимательно так как они могут свидетельствовать о разделении базы опрашиваемых на несколько групп или кластеров, что возможно потребует уточнения вопросов под каждую группу и проведения повторного анкетирования.



К анализу отклонений следует подойти внимательно, они могут свидетельствовать о разделении выборки на несколько групп, что потребует уточнения анкеты

Хотя интервью является одним из основных способов извлечения требований, не всегда существует центр компетенции это способный ответить на эти вопросы. Если такой центр отсутствует, то на помощь аналитику может прийти метод мозгового штурма. Метод мозгового штурма является групповым, обычно в таком мероприятии участвует вся команда проектирования. В ходе мозгового штурма сперва выставляются вопросы, затем собирается информация в виде вариантов ответов на вопросы по принципу «плохих идей не бывает», затем варианты обсуждаются, ранжируются путем голосования и выбираются те идеи, которые получили наибольшую поддержку.



Главный принцип мозгового штурма «плохих идей не бывает»

В случае если ответы на вопросы о требованиях содержится внутри уже созданной системы, кто следует произвести наблюдение, зафиксировав факторы таких требований. Допустим если известно с каким ресурсом работает система, то следует оценить все свойства ресурса с которым система может продолжать работать. Также, если это целесообразно и безопасно, изменять ресурс до тех пор, пока система не прекратит нормальное функционирование, для того чтобы определить фактические границы требований.

Анализ нормативной базы и рабочей документации может также служить источником требований. Во многих экономико-социальных системах нормативная база служит основным источником требований. Например, платежная система коммерческого банка подлежит жесткому регулированию, другой пример система взаимоувязанного документооборота в федеральном органе исполнительной власти. Примеров можно привести достаточно много. В таких системах во главу угла поставлен нормативный акт, определяющий требования к реализации системы до сравнительно мелких деталей.



Во многих экономико-социальных системах нормативная база служит одним из основных источников требований

Если рабочая система ещё не создана или её изучения затруднительно то можно воспользоваться методом сравнения с аналогом если он функционирует и доступен. При этом выполняются те же действия, что при наблюдении за системы и при

анализе рабочей документации, с отражением выявленных требований на анализируемую систему.

Выделяют несколько видов требований:

**Бизнес-требования** — требования самого высокого уровня, которые определяют цель создания системы,

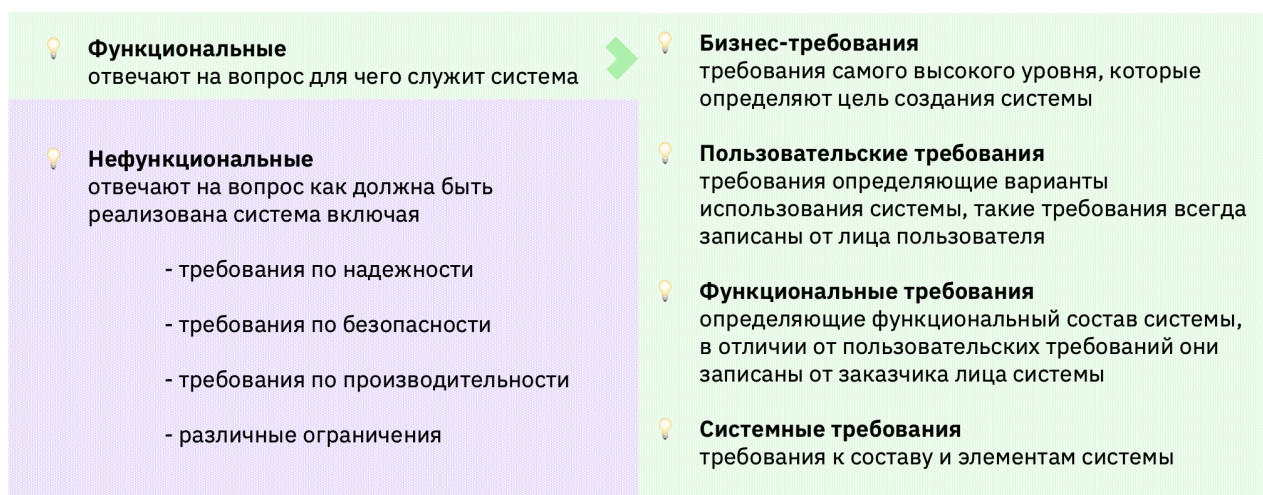
**Пользовательские требования** – требования определяющие варианты использования системы, такие требования всегда записаны от лица пользователя,

**Функциональные требования** – определяющие функциональный состав системы, в отличие от пользовательских требований они записаны от заказчика лица системы,

**Системные требования** – требования к составу и элементам системы, дополняют функциональные требования техническими аспектами реализации.

Выделяют два класса требований. Вышеперечисленные требования относятся к классу **функциональных требований**, то есть таких требований, которые отвечают на вопрос для чего служит система.

Другой класс требований называется **нефункциональными требованиями**, в нем относятся различного рода правила, бизнес или административные, регламенты, ограничения, требования по качеству и пользовательскому интерфейсу. Нефункциональные требования отвечают на вопрос как должна быть реализована система.



При выявлении все требования стоит приоритизировать, так чтобы задачи по их реализации были в порядке, который гарантирует первоочередное исполнение самых важных требований.

Одним из методов для категоризации требований является метод **MoSCoW**.

Методологию MoSCoW сегодня знают во всем мире и применяют широко в разных областях управления. С известной столицей технику ничего не связывает.

Согласные буквы в акрониме MSCW — это степени приоритетности:

**M** — задачи и требования, которые имеют самый высокий приоритет и должны быть первоочередно применимы к продукту в первую очередь. Без них релиз не будет выполнен (это *must*).

**S** — важные требования, но не с самой высокой приоритетностью. Обычно они не имеют решающего значения, но все равно обязательны к исполнению (это *should*).

**C** — требования и задачи, желательные для релиза (это *could*).

**W** — наименее критичные требования, их можно проигнорировать или перенести до следующих релизов (это *would*).

На примере задач веб приложения рассмотрим, как можно определить приоритеты, согласно методологии MoSCoW:

**Must Have** — реализовать функцию просмотра каталога услуг, с возможностью запуска соответствующих процессов.

**Should Have** — внедрить функцию информирования о составе услуг.

**Could have** — добавить раздел ленты операций, показывающей историю запуска услуг.

**Would Have** — обеспечить информирование о выполнении услуги по электронной почте.

Метод предлагает быстрое и простое решение для определения приоритетов. Однако часто такой классификации по категориям может быть недостаточно. Поэтому, считается, что MoSCoW лучше подходит для внутренних проектов, а не для продуктов с большим количеством клиентов.

**Модель Кано** — другая технология, позволяющая ранжировать требования была разработана японцем Нориаки Кано в 1984 году.

Кано предложил систему координат, где по оси Y измеряется удовлетворенность, по оси X — уровень выполнения.

В модели Кано выделены 3 основные составляющие профиля качества, влияющие на удовлетворенность потребителя:

- базовое,
- требуемое,
- вызывающие восхищение.

Ожидаемые свойства по Кано — это базовые свойства продукта или услуги. Они должны быть по умолчанию. Покупатель вряд ли будет задумываться об этих свойствах, поскольку воспринимает их как что-то само собой разумеющееся. Например, если вы купили билет на поезд, то поезд будет иметь места для пассажиров. Это естественно, но проектируя систему это даже не обсуждается. Но потребитель будет просто обескуражен, если базовые свойства будут нарушены. Такие свойства часто относятся к нефункциональным.

Требуемое качество — это те свойства, за которые потребитель готов платить. Они оговариваются при заключении контракта. Их выполнение влияет напрямую на удовлетворенность потребителя. Именно на основных свойствах продукта создаются конкурентные преимущества.

В примере с поездом, требуемым качеством может стать скорость доставки до пункта назначения. Относительно веб сайта — это перечень разделов и услуг, запрашиваемых пользователем. Например, для личного кабинета банка это возможность открытия счета и денежного перевода. Эти свойства составляют основу функциональных требований.

Свойства, вызывающие восхищение, по Кано — это желаемое качество. Потребитель не готов за них платить или отдельно их обговаривать, но был бы рад, если бы эти свойства были у услуги или системы.

В примере с поездом это может быть дополнительной услугой wifi.

Оценив принадлежность требований к профилю качества, можно определить их приоритет и понять как требования будут влиять на удовлетворенность потребителя.



На этапе выявления следует выявить и упорядочить по степени важности все требования

Частой ошибкой выявления требований является невнимание к нефункциональным требованиям. Зачастую на старте проекта не понятно насколько система будет

востребована, что приводит к игнорированию большого числа важных нефункциональных требований. Это зачастую приводит к серьезным проблемам эксплуатации созданных систем.

Чтобы проиллюстрировать важнейшее значение нефункциональных требований, представьте, что программное приложение рассчитано на определенную, ожидаемую нагрузку по обращениям клиентов в момент времени. Система проверена на данной нагрузке и никаких отрицательных факторов не выявлено. Вы, как системный аналитик, не ждёте сюрпризов, так как система прошла все испытания согласно выявленным функциональным и нефункциональным требованиям.

Но в реальной жизни пиковая нагрузка на систему может стать настоящим сюрпризом. Представьте, что создаваемая система оказалась на несколько часов недоступна потребителям. Допустим рассматриваемая система является частью сервиса дистанционного банковского обслуживания реализующего функции личного кабинета и некоторое время не работал весь личный кабинет. За время недоступности приложения возник отложенный спрос на сервис предоставляемый системой.

Когда все заработало, то в начальный момент времени количество клиентских сессий возросло до 10 раз к его обычному значению. Выглядит эта ситуация как цунами, если приложение не будет держать нагрузку, то оно окажется просто “затоптанным” клиентами. В случае, если приложение не справится с нагрузкой, то оно либо станет недоступным на некоторое время, либо его производительность деградирует настолько, что его использование станет затруднительным.

Если это веб приложение, то пользователи не получая обновление страницы в ожидаемое время будут думать, что проблема локализована в их браузере, и не найдут решения лучше, чем перезагрузить страницу сайта с данным приложением. Это вызовет дополнительный рост трафика к тому отложенному спросу, который уже сформирован.



Рассматривая нефункциональные требования, учитывайте возможность повышения спроса на приложение на порядок от нормальной нагрузки и закладывайте повышенные значения от предоставленных бизнесом расчетов в требования по надежности

# Согласование требований

Второй этап заключается в согласовании требований. На этом этапе аналитику для разрозненных сведений следует определить индивидуальные характеристики качества информации, а также риски, которые влечет как исполнение, так и не исполнение этих требований.

Популярный стандарт бизнес анализа BABOK 3.0 предоставляет девять характеристик качества требований, которые могут быть использованы в качестве чек листа:

Российский стандарт управления требованиями ГОСТ 59194 от 2020 дополняет этот перечень:	
💡 Атомарность	💡 <u>Приоритизированность</u>
💡 Выполнимость	💡 Тестируемость
💡 Полнота	💡 Недвусмысленность
💡 Краткость	💡 Понятность
💡 <u>Консистентность</u>	
	💡 Адекватность
	💡 <u>Прослеживаемость</u>
	💡 Единичность
	💡 <u>Проверяемость</u>
	💡 Завершенность
	💡 Обязательность
	💡 Законность
	💡 Абстрактность

- атомарность — означающее, что требование должно быть сформулировано таким образом, чтобы его невозможно было разделить на части,
- выполнимость,
- полнота,
- краткость,
- консистентность,
- приоритизированность,
- тестируемость,
- недвусмысленность,
- понятность.

Российский стандарт управления требованиями ГОСТ 59194 от 2020 дополняет этот перечень такими понятиями как:

- актуальность требования — то есть актуальность для этапа и уровня разработки системы,
- единичность требования — означающее, что каждое требование должно относиться только к одной характеристике разрабатываемой системы,
- завершенность требования — означающее полноту определения требования,
- однозначность требования — означающее его недвусмысленность,



- прослеживаемость требования — подразумевающее обязательную логическую связь с источником требования,
- проверяемость требования — означающее возможность проверки или тестирования требования известными способами,
- обязательность, законность и абстрактность.

Понятие абстрактности требования подразумевает, что требования к системе не должны включать детальные требования к составу элементов и наоборот.

Как вы можете заметить качество требований несколько расширяют понятие качества информации. Это происходит из-за необходимости сравнивать требования друг с другом и сопоставлять с реальным окружением. Это важно, потому что исполнение избыточных требований чревато увеличением сложности системы, как следствие трудозатрат на ее создание.

Игнорирование существенных требований несет в себе риски вызванные не удовлетворением ожиданий отдельных заинтересованных лиц, так и нарушением межсистемного взаимодействия или нормативного документа.



При наличии требований низкого качества их нужно отбросить, если они не несут существенных рисков или в противном случае проработать повторно согласно шагам первого этапа, применив дополнительные методы извлечения.

В итоге итерационного, то есть пошагового процесса согласования и сопоставления требований остаются лишь непротиворечивые требования, которые ложатся в основу проектирования системы и служат основой для ее создания. Все требования должны быть разложены в порядке приоритетов исполнения.



При выявлении новых требований их необходимо также пропустить через процесс согласования, оценив риски и приняв решение об их месте.

Давайте совместно решим задачу: пользователи ожидают что система будет всегда 24/7, руководитель клиентской службы с этим согласен, бизнес заказчик тоже и поддерживает это требование, при этом служба эксплуатации сообщает что с 20-00

до 21-00 технический перерыв в работе серверов на резервное копирование баз данных и выполнение работ на оборудовании. Как быть аналитику?



Вопрос

Пользователи ожидают что система будет всегда, при этом служба эксплуатации сообщает что с 20-00 до 21-00 необходим технологический перерыв в работе оборудования.  
Как быть аналитику?

**Ответ:** Необходимо приоритизировать требования, оценив риски: с одной стороны отказ от клиентоориентированности может быть недопустим, с другой стороны нарушение обслуживания серверов ведет к риску ущерба вследствие отсутствия резервной копии и невозможность восстановить систему в актуальном состоянии после аварийного события.

В этом случае системный аналитик может совместно со службой эксплуатации разработать требования по использованию теневого копирования работающей системы без необходимости приостановки на время резервного копирования или выполнения копирования в наименее нагруженные дни. С этими расчетами аналитик финализирует у заказчика требования, ведь требования уже согласованы, заказчику осталось выбрать цену решения и принять риск.

## Документирование требований

Третий этап — это документирование требований. Собранные и согласованные сведения должны быть записаны в текстовой форме.

Текстовое требование представляет собой утверждение, которое объясняет или выражает потребность и связанные с ней ограничения и условия. Данное утверждение должно включать субъект, его функцию (действие) и дополнение (объект, значение, ограничение).

Субъект требования идентифицирует тот объект, к которому задано требование (например, система, программное обеспечение и т. д.). Функция описывает некоторую полезную деятельность, которую должен выполнять субъект (например, взаимодействовать по протоколу ..., обеспечивать проведение платежа ...).

Например:

Отчет [это **Субъект требования**] должен содержать [это **Действие требования**] таблицу строк деталей операции [это **Объект требования**] с постраничной разбивкой и нумерацией начиная с 1 [это **Значение требования**] до 20 строк на лист [это **Ограничение требования**].

или

Например:

*На последнем листе отчета [Субъект] должен быть сформирован [Действие] штамп в формате банка и электронная подпись [Значение]*

По схожему принципу формулируются требования заинтересованных сторон в виде пользовательских историй, в которых в дополнительно указывается источник требований.

Например:

Для меня как для пользователя с правом аудитор [Субъект] необходим доступ к просмотру [Действие] деталей реестра [Объект].

Для закрепления попробуйте сформулировать свои требования по указанному шаблону Субъект — Действие — Объект.

Системный аналитик в своей деятельности не одинок. Сейчас все чаще ему помогает бизнес-аналитик – эксперт по той предметной области в которой должна функционировать система.

Бизнес-аналитик выявляет бизнес требования, формализует их в виде пользовательских историй, выявляет и согласовывает пользовательский путь, то есть последовательность взаимодействия пользователя с системой, а также определяет варианты использования создаваемых функций, после чего системному аналитику остается выполнить функциональное проектирование, описать системные требования, затем перевести требования бизнеса на язык

исполнителей, предметно, полно, понятно и непротиворечиво описать детальные требования.

После того как требования собраны, согласованы и документированы их проверяют и утверждают.

## **Утверждение требований**

Четвертый этап — утверждение требований, представляет собой визирование документа содержащего требования к системе всеми заинтересованными сторонами. Это самый опасный этап управления требованиями, если он будет длиться долго, то это может сказаться существенным образом на сроках реализации системы.

Поэтому в современных гибких итерационных процессах разработки стараются исключить эффект полной блокировки процесса разработки при ожидании утверждения за счет деления требований по отдельным элементам или принятием рисков продолжения работы в условиях не согласованных требований.

## **Управление требованиями**

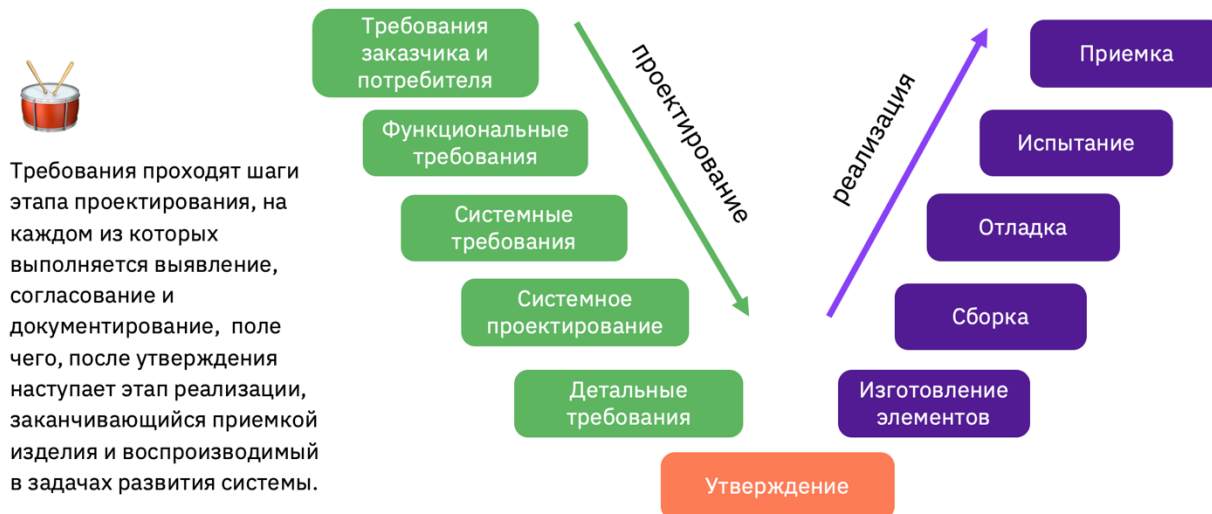
После того как требования заинтересованных сторон, функциональные требования, системные требования выявлены, согласованы, документированы и утверждены для системы в целом или для ее элемента, этап проектирования заканчивается и наступает стадия разработки, при которой последовательно производится изготовление элементов, их интеграция (сборка), отладка (тестирование) системы, ее испытание и приемка в эксплуатацию.

При этом важно отметить, что при изготовлении отдельных элементов системы проверяются детальные требования, при отладке и тестировании системы верифицируются системные требования и при вводе в эксплуатацию валидируются бизнес требования.

Такая модель, облегчающая понимание жизненного цикла требований в системе называется V-образной, а деятельность по валидации системы, то есть подтверждение того, что система удовлетворяет потребностям заинтересованных сторон и валидации системы, то есть подтверждение того, что система разработана в соответствии с заданными требованиями - относятся к контролю требований.

## V-модель управления требованиями

Лучше всего последовательность этапов жизненного цикла описывает V-модель



Управление требованиями представляет собой процесс формирования и изменения требований в целях снижения рисков и издержек процесса создания и развития системы. Общая цель такого процесса – выявлять значимые требования как можно раньше, и поддерживать качество требований используемых для разработки системы.



Цель процесса – выявлять значимые требования как можно раньше, и поддерживать качество требований используемых для разработки системы.

В литературе V-модель определяется чаще как модель процесса разработки, но такое определение не совсем корректно с точки зрения системного анализа, для которого разработка является своеобразной вершиной айсберга, и не раскрывает всех подготовительных этапов связанных с проектированием.

В качестве альтернативы управления требованиями на основе V-модели выступает каскадная модель.

Каскадная модель или модель водопада является самой простой и самой распространенной классической методологией создания системы. Ее суть заключается в том, что все стадии жизненного цикла следуют последовательно:

начинается процесс с разработки требований к системе, далее следует проектирование, затем идет непосредственно разработка, далее — тестирование, а последней ступенью является эксплуатация и поддержка.

В каскадной модели ключевым фактором является строгая последовательность выполнения стадий. Сначала определяются требования к создаваемой системе. Только когда требования полностью определены, начинается этап проектирования продукта.

На этапе проектирования создается документация, подробно описывающая план разработки на основе ранее разработанных требований к данному продукту. Когда этап проектирования завершится и вся документация для программистов будет готова, программный продукт переходит в стадию разработки. На этой стадии происходит непосредственно реализация ранее спроектированного функционала или частей функционала, которые на этом же этапе интегрируются в общую систему.

Подготовка к тестированию, включающая в себя составление тестовой документации, может начинаться и до окончания стадии разработки, поскольку требования к приложению, на основе которых эта документация составляется, уже разработаны, однако в фазу непосредственного тестирования программный продукт все равно вступает после окончания стадии разработки.

На этапе тестирования проверяется весь разработанный функционал программы, находятся всевозможные дефекты, чтобы свести к минимуму возможность возникновения ошибок после вывода программного продукта в эксплуатацию. Только после того, как команда тестирования принимает решение о нецелесообразности дальнейшего тестирования, программный продукт переходит на стадию выпуска в эксплуатацию и дальнейшей поддержки.

Как и V-модель каскадная модель проста для понимания. В отличие от V-модели, каскадная модель содержит большое число блокеров, условий, которые должны быть исполнены чтобы процесс двинулся дальше. В итоге именно это обстоятельство существенно удлинняет сроки реализации проекта при условии изменчивости требований.

Почему это так важно? Каскадная модель применяется при реализации крупных проектов с длительными сроками планирования. Если за время реализации проекта по созданию системы требования меняются, то согласно каскадной методологии проект подлежит перезапуску с прохождением полного цикла от выделения дополнительного финансирования до повторной реализации.



Каскадный подход эффективен только в условиях стабильных не изменчивых требований и среды

Каскадная модель больше подходит для систем со стабильными не изменчивыми требованиями, в то же время V-модель лучше адаптирована к инкрементальному производственному процессу.

При подходе, когда во главу угла ставятся экономические показатели расходования средств, а не конечный результат, говорить о создании адекватного продукта или системы не стоит.



Можно сказать что V-модель является развитием каскадной модели, и при определенных допущениях может быть воплощена в качестве каскадной модели производственного процесса

В свою очередь каскадная модель не может быть сведена к V-модели без дополнительных договоренностей.

## Контроль требований

Контроль требований — завершающий этап жизненного цикла требований. Контроль это управление изменениями требований и обеспечение повторной проверки системы. Такая проверка называется регрессионной. Может происходить так, что требования изменяются на последних этапах работы над системой. В этом случае следует произвести оценку влияния изменений и принять решение о необходимости такой проверки.

Контроль требований производится на этапах изготовления элементов, тестирования (отладки) и приемки.

Качество продукта — разрабатываемой системы имеет много аспектов, включая соответствие спецификациям, отсутствие дефектов и удовлетворение продукта требованиям пользователей. В ИСО/МЭК 25010 "Модели качества систем и программного обеспечения" определено восемь показателей качества, которые могут быть измерены или оценены путем тестирования.

Тестирование программного обеспечения должно быть направлено на предоставление информации о программном продукте и нахождение максимально возможного числа дефектов требований и программного кода на возможно ранних этапах процесса разработки при заданных ограничениях стоимости и графика разработки.

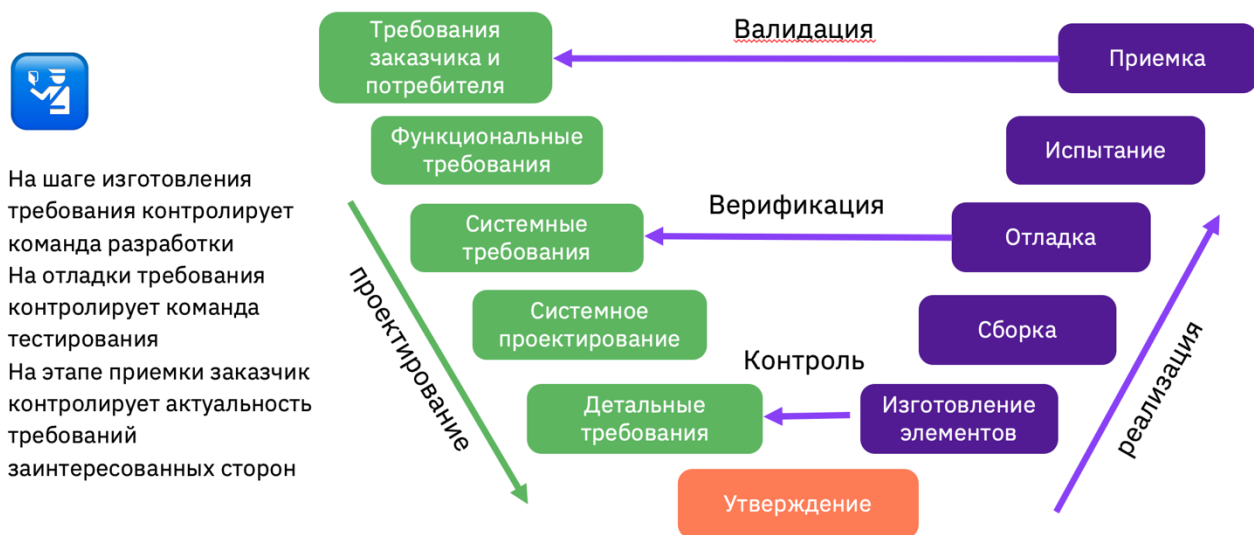
Основные положения тестирования заключаются в следующем:

- тестирование — это процесс, представляющий собой совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы в выходы;
- организационный процесс тестирования устанавливает и поддерживает политики тестирования и стратегии тестирования, которые повсеместно применяются в проектах и функциях организации;
- тестирование необходимо планировать, контролировать и управлять им. Процессы тестирования, описанные в ИСО/МЭК/ИИЭР 29119-2, включают в себя процесс менеджмента тестирования и могут быть применены к тестированию во всех жизненных циклах разработки и менеджменте исследовательского тестирования;
- процессы и подпроцессы тестирования применимы для любой фазы или уровня тестирования (например, тестирование системы) и для любого типа тестирования (например, нагрузочное тестирование);
- тестирование предполагает исследование элемента тестирования;
- статическое тестирование может также включать в себя использование инструментов статического анализа;
- динамическое тестирование представляет собой нечто большее, чем "просто" выполнение исполнимых элементов тестирования, сюда входят также как действия подготовки, так и последующие действия. Процессы динамического тестирования, описанные в ИСО/МЭК/ИИЭР 29119-2, охватывают каждое из действий, которые будут выполняться в ходе динамического тестирования;
- верификация — это подтверждение путем представления объективных доказательств выполнения данным рабочим элементом установленных требований;
- валидация демонстрирует, что рабочий элемент может использоваться пользователями для решения определенных ими задач;



- тестирование, как статическое так и динамическое, должно быть направлено на получение обоих типов подтверждения, хотя и должно допускать, что подтверждение не будет получено немедленно из-за обнаружения дефектов.

Например, при приемке раздела сайта из-за несогласованности между командами разработки выяснилось, что изменился дизайн общих элементов сайта по причине возникновения дополнительных требований пользователей и теперь раздел выбивается из общего приложения. Заказчик должен принять сложное решение, доработать приложение в течении одного месяца либо принять риски и вывести важную функциональность точно в срок, но с проблемой дизайна. Так в случае ошибки на начальном этапе проектирования обойтись без потерь крайне сложно.



Стоит отметить, что в модели управления требованиями, требования заинтересованных сторон являются самыми верхнеуровневыми, а по времени возникновения самыми ранними, относительно их системных спецификаций и подготовленных детализированных описаний на языке исполнителя.

Ошибка таких требования иногда может быть выявлена только при валидации или после того, как система разработана, протестирована и передана на эксплуатацию, что делает такую ошибку наиболее значимой с точки зрения времени исправления. Ошибка в системных требованиях – это вторая по стоимости исправления ошибка, зачастую требующая изменения реализации. Но ошибки в детализированном описании требований зачастую выявляются на момент ознакомления исполнителя с

заданием или на этапе модульного тестирования отдельного элемента. Такие ошибки легко и сравнительно быстро исправляются.

Мы ознакомились с вами с понятием требований к системе, узнали о возможных источниках требований и принятой классификации требований и их качеств. Ознакомились с V-образной моделью дающей наглядное представление о жизненном цикле требований и управлении ими.

На этом мы завершаем вводную часть курса технической специализации системный анализ. Лекции в этом курсе закончились. Мы завершили курс “Введение в системный анализ”. Вы познакомились с понятием системного анализа, научились использовать инструментарий системного анализа, узнали о важности качества информации и управлении данными, научились извлекать и управлять требованиями.

В следующих курсах вы рассмотрите более подробно процессы проектирования и разработки.

## Что можно почитать еще?

1. Для лучшего изучения темы можно почитать JIRA: управление требованиями <https://habr.com/ru/company/lanit/blog/486754/>
2. Тестирование требований: как я нахожу ошибки в бизнес-логике, фичи прежде, чем их закодят <https://habr.com/ru/company/plesk/blog/550550/>
3. Некоторые замечания по вопросу сбора требований при разработке программного обеспечения <https://habr.com/ru/post/409471/>
4. ГОСТ Р 59194-2020 Управление требованиями.

## Используемая литература

1. Горлушкина Н.Н. Системный анализ и моделирование информационных процессов и систем

2. Абрамов Г.В., Медведков И.Е., Коробова Л.А. Проектирование информационных систем
3. Грекул В. И. Проектирование информационных систем
4. Методики оценки эффективности информационных систем и информационных технологий в бизнесе
5. ГОСТ ISO/IEC/IEEE 29119-1:2013 "Программная и системная инженерия. Тестирование программного обеспечения"