

Требования к программному обеспечению

Понятие требований

Чтобы подойти к требованиям, повторим **понятия жизненного цикла программного обеспечения**, т.к. требования должны быть согласованы с его этапами.

Жизненный цикл программного обеспечения (ПО) — период времени, который начинается с момента принятия решения о необходимости создания **программного** продукта и заканчивается в момент его полного изъятия из эксплуатации.

Требования формулируются в **техническом задании (ТЗ)**. Может быть и отдельный документ

Основные стандарты, методологии и своды знаний, где упоминается требования, Техническое задание (ТЗ) и **СПЕЦИФИКАЦИЯ (Specification)**:

- **ГОСТ 34**: Техническое задание на разработку автоматизированной системы
- **ГОСТ 19**: ЕСПД ЕДИНАЯ СИСТЕМА ПРОГРАММНОЙ ДОКУМЕНТАЦИИ
- **Стандарты разработки, сопровождения, тестирования и управления конфигурацией компонентов и программных средств**
Стандарты **ISO/IEC** (International Organization for Standardization International / Electrotechnical Commission - Международная электротехническая комиссия)

ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы регламентирует структуру ТЗ на создание именно СИСТЕМЫ, в которую входят ПО, аппаратное обеспечение, люди, которые работают с ПО, и автоматизируемые процессы. Согласно ГОСТ 34 техническое задание должно включать следующие разделы:

1. Общие сведения
2. Назначение и цели создания (развития) системы
3. Характеристика объектов автоматизации
- 4. Требования к системе**
5. Состав и содержание работ по созданию системы
6. Порядок контроля и приемки системы
- 7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие**
- 8. Требования к документированию**
9. Источники разработки

“ГОСТ 19.ххх Единая система программной документации (ЕСПД)” — это комплекс государственных стандартов, устанавливающих взаимосвязанные правила разработки, оформления и обращения программ (или ПО) и программной документации. Т.е. этот стандарт относится к разработке именно ПО.

Согласно [ГОСТ 19.201-78 Техническое задание, требования к содержанию и оформлению](#) техническое задание должно включать следующие разделы:

1. Введение;
2. Основания для разработки;
3. Назначение разработки;
4. Требования к программе или программному изделию;
5. Требования к программной документации;
6. Технико-экономические показатели;
7. Стадии и этапы разработки;
8. Порядок контроля и приемки;
9. Приложения.

В зарубежных стандартах (их очень много, поэтому неудобно пользоваться, но надо знать, т.к. в целом все подробно, более детально и может быть полезно):

Детальные требования (организованы по разному)

1. **Требования к внешним интерфейсам**

- ✓ Интерфейсы пользователя
- ✓ Интерфейсы аппаратного обеспечения
- ✓ Интерфейсы программного обеспечения
- ✓ Интерфейсы взаимодействия

2. **Функциональные требования**

3. **Требования к производительности**

4. **Проектные ограничения (и ссылки на стандарты)**

5. **Нефункциональные требования (надежность, доступность, безопасность и пр.)**

6. **Другие требования**

Требования к программному обеспечению — утверждения относительно атрибутов, свойств или качеств программной системы, подлежащей реализации. Создаются в процессе разработки и анализа требований.

Требования могут быть в виде текста и графических моделей.

Используются **на стадии проектирования ПО**, а также в процессе **проверки ПО**, так как **тестирование ПО** основано на требованиях.

Этапу разработки требований предшествует технико-экономическое обоснование (ТЭО), **концептуальная фаза** анализа проекта. Фаза разработки требований может быть разбита на **выявление требований** (сбор, понимание, рассмотрение и выяснение потребностей заинтересованных лиц), **анализ** (проверка целостности и законченности), **спецификация** (документирование требований) и **проверка** правильности.

Виды требований по уровням

Бизнес-требования — определяют назначение ПО, описываются в документе о видении и границах проекта

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

Пользовательские требования могут выражаться в виде фраз утверждений, в виде сценариев использования (англ. *use case*), сценариев взаимодействия - функций.

Виды требований по характеру

- **Функциональный характер** — требования к **поведению системы**
 - Бизнес-требования
 - Пользовательские требования
 - Функциональные требования
- **Нефункциональный характер** — требования к **характеру поведения системы**
 - Бизнес-правила — определяют ограничения, проистекающие из предметной области и свойств автоматизируемого объекта (предприятия)
 - Системные требования и ограничения — определения элементарных операций, которые должна иметь система, а также различных условий, которым она может удовлетворять. К системным требованиям и ограничениям относятся:
 - Ограничения на программные интерфейсы, в том числе к внешним системам
 - Требования к атрибутам качества
 - Требования к применяемому оборудованию и ПО
 - Требования к документированию
 - Требования к дизайну и юзабилити
 - Требования к безопасности и надёжности
 - Требования к показателям назначения (производительность, устойчивость к сбоям и т. п.)
 - Требования к эксплуатации и персоналу
- Прочие требования и ограничения (внешние воздействия, мобильность, автономность и т. п.).

Источники требований

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

Качество требований

Характеристика	Объяснение
Единичность	Требование описывает одну и только одну вещь.
Завершённость	Требование полностью определено в одном месте и вся необходимая информация присутствует.
Последовательность	Требование не противоречит другим требованиям и полностью соответствует внешней документации.
Атомарность	Требование «атомарно». То есть оно не может быть разбито на ряд более детальных требований без потери завершённости.
Отслеживаемость	Требование полностью или частично соответствует деловым нуждам как заявлено заинтересованными лицами и документировано.
Актуальность	Требование не стало устаревшим с течением времени.
Выполнимость	Требование может быть реализовано в пределах проекта.
Недвусмысленность	Требование кратко определено без обращения к техническому жаргону, акронимам и другим скрытым формулировкам. Оно выражает объективные факты, не субъективные мнения. Возможна одна и только одна интерпретация. Определение не содержит нечётких фраз. Использование отрицательных утверждений и составных утверждений запрещено.
Обязательность	Требование представляет определённую заинтересованным лицом характеристику, отсутствие которой приведёт к неполноценности решения, которая не может быть проигнорирована. Необязательное требование — противоречие самому понятию требования.
Проверяемость	Реализованность требования может быть определена через один из четырёх возможных методов: осмотр, демонстрация, тест или анализ.

Проверка требований

- при их выполнении, можно сказать вып/не вып. Методика проверки — тесты. Если проверка тестами невозможна, тогда должен использоваться другой метод (анализ, демонстрация, осмотр или обзор дизайна).
- Некоторые требования не поддаются проверке. Например, что **система *никогда* не должна или *всегда* должна показывать** специфическое свойство. Тестирование этих требований потребовало бы **бесконечного цикла тестирования**. Они должны быть переопределены так, чтобы они **стали поддающимися проверке**. Все требования должны поддаваться проверке.
- **Нефункциональные требования, которые не поддаются проверке** на программном уровне, все равно должны быть сохранены как документация намерений клиента. Такие требования к продукту **могут быть преобразованы в требования к процессу**. Например, сложные требования безопасности авиационного программного обеспечения могут быть удовлетворены следованием процессу разработки.

Анализ требований

- При разработке требований **часто возникают проблемы двусмысленности, неполноты, и несогласованности**. Устранение этих проблем на **этапе разработки требований стоит гораздо меньше**, чем на поздних стадиях разработки. Для решения и устранения этих проблем существует процесс разработки требований.
- При разработке требований **нужен компромисс** между слишком неопределёнными и требованиями столь детализированными, что они:
 - ✓ **требуют много времени** для разработки, иногда даже **рискуют устареть** к концу разработки;
 - ✓ **ограничивают** возможные способы реализации;
 - ✓ являются слишком **дорогостоящими**.
 - ✓ Их никто не дочитает

Документирование требований

- Требования - средство коммуникации между различными заинтересованными лицами. Это означает, что **требования должны быть просты и понятны для обычных пользователей и разработчиков**. Один общий способ задокументировать требование — это написать **спецификацию**, что должна сделать система:
- Спецификация требований программного обеспечения
- Спецификацию программного обеспечения часто называют техническим заданием. НО: Спецификация требований является частью технического задания.
- За создание спецификации программного обеспечения отвечает Системный аналитик, иногда — Бизнес-аналитик.
- Для **графических моделей требований** используются диаграммы или методологии

Изменение требований

Требования изменяются со временем.

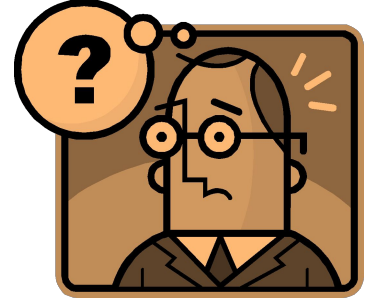
После того, как требования определены и одобрены, **изменения должны попадать под контроль внесения изменений.** Для многих проектов требования изменяются до **завершения создания системы.** Это происходит частично из-за сложности программного обеспечения и того факта, что пользователи не знают, что им нужно на самом деле. Эта особенность требований привела к появлению процесса [управления требованиями.](#)

Методы обследования – выявления требований (организационно– функционального анализа):

- 1) Анкетирование,
- 2) метод интервью,
- 3) метод наблюдения, анализ конкурентных продуктов,
- 4) документальный анализ,
- 5) метод личного участия, работа совместно с пользователями,
- 6) работать в команде заказчика, учиться у него,
- 7) опыт, повторное использование спецификаци
- 8) встречи и совещания,
- 9) анализ моделей деятельности



Анкетирование



Составление **листа-опросника** (анкеты, брифа), который может содержать **открытые** (требуют от опрашиваемого сформулировать его ответ) и **закрытые** (требуют от опрашиваемого выбрать ответ из предложенных вариантов) **вопросы**.

Анкетирование используется для того, чтобы **подтвердить** или **детализировать** ранее известные требования, **выбрать** параметры для решений.

Самый известный пример анкетирования - **“Бриф на разработку сайта”** — анкета содержащая список основных требований и информацию о будущем сайте.

Этапы проведения анкетирования

Шаг 1. Выявление цели анкетирования.

Шаг 2. Подбор целевой аудитории.

Шаг 3. Составление анкеты.

Шаг 4. Тестирование анкеты.

Шаг 5. Проведение анкетирования и повторная проверка анкеты.

+

1. Высокая скорость получения результатов.

2. Небольшие материальные затраты.

-

1. Не подходит для выявления неявных требований.

2. При составлении опросника невозможно учесть все необходимые вопросы.

Интервью - своего рода беседа "по душам"

с заинтересованным лицом

Задавать **открытые вопросы** для получения информации и **закрытые** для того, чтобы подтвердить или опровергнуть конкретные варианты требований. Способ **применяется для получения информации по какой-либо конкретной теме, уточнения требований**. Провести хорошее интервью достаточно сложно. Вы должны гибко реагировать на реакцию интервьюируемого и в случае необходимости изменять порядок заготовленных вопросов или их формулировку. Не забудьте **включить диктофон во время интервью или вести заметки**.

+ Возможность задавать вопросы в произвольной последовательности.

Возможность использовать вспомогательный материал.

Анализ невербальной реакции опрашиваемого человека, позволит сделать дополнительный вывод о достоверности его ответов.

- Интервью отнимает достаточно много времени и сил

Метод наблюдения за деятельностью и процессами

+

1. Позволяет наглядно увидеть проблему и разработать наиболее оптимальный вариант ее решения.
2. Помогает наиболее точно собрать требования, наблюдая за работой сотрудников.

-

1. В процессе наблюдения могут быть упущены некоторые альтернативные сценарии бизнес процесса.
2. Трудно применим на секретных предприятиях или опасных (вредных) производствах.
3. От сотрудников мы получаем видение систему As_Is (как есть). То, как они сейчас работают. А на выходе нам надо систему под цели заказчика To_Be (как должно быть).

Не делать 1 в 1, как просят сотрудники. Задаем вопрос «Зачем?» и «Почему?», вместо «Что?» и «Как?»

Метод документального анализа и Автозапись



Изучение существующей документации. Примеры документации: регламенты, описания процессов, структура организации, спецификации продукта, стандарты и инструкции, шаблоны документов, отчеты, распоряжения, квитанции ...

Применимо при автоматизации устоявшихся в организации бизнес-процессов.

+ быстрое получение информации.

- не применим при наличии неактуальных документов или их полном отсутствии.

Автозапись подразумевает работу с **документом, автором которого является сам Заказчик** или конечный пользователь. Примером такого метода может быть работа с **концепцией или видением проекта (сам Заказчик любит называть это — «ТЗ»)**, которую он прислал вам на момент начала работ по аналитике.

- + Помогает лучше понять сложные процедуры или процессы.
- Метод зависит от опыта Заказчика, а также от его умения формулировать и выражать свои мысли (редко ТЗ заказчика не соответствует тому, что он имеет в виду, но его надо полностью учесть, т.к. это документ).

Обучение

это процесс, в котором Заказчик, знающий процесс, обучает аналитика **по принципу учитель — ученик.**

Метод полезен, когда процессы и деятельность сотрудников Заказчика трудно описать с помощью других методов или **Заказчик не может предоставить адекватное описание требований.**

Обучение **позволяет лучше понять сложные бизнес-процессы**, а также преодолеть трудности, связанные с нехваткой абстрактного мышления и самовыражения у будущих пользователей системы.

+

Позволяет понять сложный бизнес процесс и предложить наилучшее решение.

-

Высокая стоимость и длительность.

Метод неприменим на опасных (вредных) производствах.



Повторное использование спецификации

Повторно использовать спецификации можно в том случае, если **есть уже завершенные** один или несколько подобных проектов.

ТЗ, предыдущего проекта может быть использовано с целью сократить продолжительность сбора, анализа и разработки требований, что позволит быстрее начать разработку.

Например, **ТЗ для интернет магазинов похожи друг на друга** и содержат одинаковые требования.

Но часто лишь часть документации актуальна для нового проекта, поэтому потребуется тщательная проверка требований на соответствие текущим целям и задачам нового Заказчика.

+

Сокращение времени на разработку документации.

-

Высокая стоимость первого проекта.

Излишняя детализация требований, может привести к их

разработчика - работа совместно с будущими

пользователями

изучение предметной области «изнутри», **выполняя определенные функции.** Один из **наиболее эффективных** методов, позволяет получать от представителя Заказчика своевременную оценку корректности реализации, в короткие сроки получать обратную связь и дополнительную информацию для корректировки и разработки требований. Метод применяется для сбора и управления требованиями **при итерационной разработке**, позволяет оперативно собирать, согласовывать и дорабатывать требования. Наличие представителя заказчика в компании разработчика является одним из главных правил Agile 'ædʒl.

+

Быстрое получение обратной связи и информации от Заказчика.

-

Достаточно высокая цена для Заказчика. Затраты по времени

Совещание - встреча

Обсуждение конкретных **вопросов, которые были определены и озвучены участникам заранее**. Привлекаются люди, которые придерживаются **разных точек зрения** по проблеме и могут помочь описать требования, **с разных сторон**. В процессе совещания уточняется список требований, выявляются скрытые требования и решаются конфликты требований.

Совещания являются одной из ключевых практик в Agile (гибкий), в них участвуют все стороны развития проекта и решения проблемы - Спринт, скрам (схватка), канбан - технологии.

+

Позволяет **развить и детализировать** требования, определить приоритеты.

-

Сложности в организации встречи, **если команда географически разделена**.



Мозговой штурм

наиболее **часто используемый метод** получения требований, которые связаны с новыми или плохо изученными направлениями деятельности или функциями Заказчика. Он позволяет собрать множество идей от различных заинтересованных лиц в кратчайшие сроки и практически бесплатно. Во время мозгового штурма **участники «накидывают» любые идеи**, касающиеся решения данной проблемы.

С помощью этой методики можно проработать **несколько различных вариантов** решения заданной проблемы, а также **разрешить конфликты требований**.

+

Позволяет генерировать множество (в том числе и нестандартных) **вариантов решений**, а также позволяет участникам развивать идеи друг друга.

-

Участники мозгового штурма **должны быть мотивированы** на идеи.

Трудно применим в распределенных командах.

Use cases или варианты использования (или IDEF0)

позволяют собрать и сформировать **функциональные требования участников**: Диаграммы вариантов использования, определяют границы решения и показывают **связи** с внешними системами и участниками.

Метод позволяет **детализировать требования** с точки зрения пользователей, а также помогает **уточнить и систематизировать функционал**, который требуется реализовать.

+

Позволяет проработать **все варианты развития сценария** (основной и альтернативные сценарии)

-

Метод **не применим для сбора нефункциональных требований**

Зависит от владения заказчиками этим методом.

Автоматизация сбора

В самом простом случае вручную заносятся собранные данные в таблицы для обработки одновременно с их сбором.

Автоматический сбор данных:

использование сканеров, датчиков. Работу по передаче данных в компьютер берут на себя драйвера устройства, а программы предоставляют драйверу информацию.

Накопители данных – переносные мини-компьютеры в эргономичных и защищенных корпусах, поддерживают как ручной ввод данных с помощью мини-клавиатуры, так и

использование портов ввода/вывода