

Разработка требований к программному продукту и их оформление в виде технического задания (ТЗ) или спецификации (SRS)

Структура и содержание технического задания (ТЗ) определяется специальным шаблоном – государственным или международным стандартом в зависимости от объекта разработки. Чаще всего таким объектом является:

- программа (программное изделие) - это данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма.

- автоматизированная система (АС) или информационная система (ИС), которая состоит из персонала и комплекса средств автоматизации его деятельности, реализующая информационную технологию выполнения установленных функций. АС включает различные виды обеспечения: организационное, методическое, техническое, программное (ПО), информационное, лингвистическое, математическое, правовое, эргономическое.

Сегодня большинство разработчиков и аналитиков считают отечественные ГОСТ'ы морально устаревшими и все чаще вместо них для оформления требований к ИС используют шаблоны зарубежных стандартов, которые регламентируют составление спецификаций требований к ПО.

Раздаточный материал № 24

ГОСТ 34.602-89 выделяет следующие разделы в ТЗ на создание АС	ГОСТ 19.201-78 в ТЗ на разработку программы	IEEE 830-1998
1. Общие сведения	1. Введение	1. Введение
2. Назначение и цели создания системы	2. Основания для разработки	a. Назначение
3. Характеристика объектов автоматизации	3. Назначение разработки	b. Область действия
4. Требования к системе	4. Требования к программе или программному изделию	c. Определения, акронимы и сокращения
a. Требования к системе в целом	5. Требования к программной документации	d. Ссылки
b. Требования к функциям, выполняемым системой	6. Техничко-экономические показатели (можно взять из ТЭО)	e. Краткий обзор
c. Требования к видам обеспечения	7. Стадии и этапы разработки	2. Общее описание
5. Состав и содержание работ по созданию системы	8. Порядок контроля и приемки	a. Взаимодействие продукта с другими продуктами и компонентами
6. Порядок контроля и приёмки системы	9. Приложения	b. Функции продукта (краткое описание)
7. Требования к составу и содержанию работ по		c. Характеристики пользователя
		d. Ограничения
		e. Допущения и зависимости

<p>подготовке объекта автоматизации к вводу системы в действие</p> <p>8. Требования к документированию</p> <p>9. Источники разработки</p>		<p>3. Детальные требования</p> <p>а. Требования к внешним интерфейсам</p> <p> i. Интерфейсы пользователя</p> <p> ii. Интерфейсы аппаратного обеспечения</p> <p> iii. Интерфейсы программного обеспечения</p> <p> iv. Интерфейсы взаимодействия</p> <p>б. Функциональные требования</p> <p>в. Требования к производительности</p> <p>г. Проектные ограничения (и ссылки на стандарты)</p> <p>д. Нефункциональные требования (надежность, доступность, безопасность и пр.)</p> <p>е. Другие требования</p> <p>4. Приложения</p> <p>5. Алфавитный указатель</p>
---	--	--

Оба стандарта допускают оформлять отдельные разделы ТЗ в виде приложений, вводить дополнительные, исключать или объединять подразделы в зависимости от особенностей и условий эксплуатации объекта разработки.

IEEE 830-1998 носит рекомендательный характер и сегодня не так часто используется на практике, как международный стандарт по инженерии требований ISO IEEE 29148-2011, который обеспечивает единую трактовку процессов и продуктов для всей системы и ПО. По сути, этот стандарт заменяет все выше приведенные стандарты.

Стандарт делит требования к ПО на:

- требования стейкхолдеров;
- системные требования, которые описывают определение, поведение и свойства каждой функции системы, ограничения по реализации, технические и качественные метрики.

1. Введение
 - 1.1. Цели
 - 1.2. Соглашения о терминах
 - 1.3. Предполагаемая аудитория и последовательность восприятия
 - 1.4. Масштаб проекта
 - 1.5. Ссылки на источники
2. Общее описание
 - 2.1. Видение продукта
 - 2.2. Функциональность продукта
 - 2.3. Классы и характеристики пользователей
 - 2.4. Среда функционирования продукта (операционная среда)
 - 2.5. Рамки, ограничения, правила и стандарты
 - 2.6. Документация для пользователей
 - 2.7. Допущения и зависимости
3. Функциональность системы
 - 3.1. Функциональный блок X (таких блоков может быть несколько)
 - 3.2. Описание и приоритет
 - 3.3. Причинно-следственные связи, алгоритмы (движение процессов, workflows)
 - 3.4. Функциональные требования
4. Требования к внешним интерфейсам
 - 4.1. Интерфейсы пользователя (UX)
 - 4.2. Программные интерфейсы
 - 4.3. Интерфейсы оборудования
 - 4.4. Интерфейсы связи и коммуникации
5. Нефункциональные требования
 - 5.1. Требования к производительности
 - 5.2. Требования к сохранности (данных)

5.3. Требования к качеству программного обеспечения

5.4. Требования к безопасности системы

5.5. Требования на интеллектуальную собственность

6. Прочее

6.1. Приложение А: Глоссарий

6.2. Приложение Б: Модели процессов и предметной области и другие диаграммы

6.3. Приложение В: Список ключевых задач

Все эти разделы дополняют и уточняют пункты, предусмотренные отечественными ГОСТ'ами. С практической точки зрения структура SRS по ISO IEEE 29148-2011 дает следующие преимущества по сравнению с ГОСТ 34.602-89 и ГОСТ 19.201-78:

- глубокий уровень детализации, в частности, подробное описание различных категорий пользователей, интерфейсов, разделение допущений и ограничений;
- разделение требований на функциональные и нефункциональные;
- рекомендации представлять функциональные требования в форме сценариев (вариантов) использования (use case);
- структурированный и детальный набор разных требований легко разделяется по итерациям гибкой разработки (Agile) через определение их приоритета и очередности реализации по релизам;
- тесная связь с бизнесом благодаря детальному описанию контекста, правил, процессов, организационной среды и других аспектов, описываемых в спецификации требований стейкхолдеров в разделах «Видение» и «Общее описание»;
- наглядность трассировки требований разных уровней друг к другу, а также к бизнес-правилам и регламентирующим документам.

Недостатком SRS по ISO IEEE 29148-2011 можно назвать большой объем этого документа, что отражается в длительности и трудоемкости его разработки.

При составлении требований необходимо понимать, что:

- требование должно быть понятным;
- требование должно быть конкретным;
- требование должно быть тестируемым;

Раздаточный материал № 26

Вид требования	Неправильная формулировка	Комментарий и как можно было сформулировать
Функциональность	Сумма затрат должна корректно распределяться по	Понятное ли это требование? В общем-то понятное, речь идет о распределении неких затрат по группе товаров. Конкретное ли это требование? Не сказано, как должна распределяться затрата, по

	соответствующи м товарам	<p>сумме, по количеству, равномерно или как-то иначе?</p> <p>Тестируемое ли это требование? Вроде бы простая вещь, но как ее проверять, если нет конкретики?</p> <p>Как можно было бы это переформулировать: «Сумма затрат, указанная в документе, должна распределиться на все товары, указанные в данном документе пропорционально стоимости этих товаров». Получилось и понятно, и конкретно. Как проверить тоже не составит труда.</p>
Эргономичность	Программа должна иметь удобный интерфейс	<p>Тут нет не конкретики, ни возможность проверить это требование. Хотя, безусловно, понятное (субъективно). Тут переформулировать никак нельзя, надо подробно расписывать каждый элемент «удобности». Например,:</p> <ul style="list-style-type: none"> - Строки в документ должны добавляться как по нажатию на кнопку «Добавить», так и при нажатии на клавиши «insert», а также вводе пользователем части наименования; - При просмотре списка товаров должна быть возможность поиска по наименованию, штрихкоду и артикулу; - и пр.
Разграничение прав доступа	Доступ к данным по прибыли должен быть доступен только финансовому директору	<p>Понятно? Почти. Правда, прибыль бывает разная, надо уточнить.</p> <p>Конкретно? Конечно нет. Как это видится в реализации? Если речь идет о валовой прибыли, то значит необходимо ограничивать доступ к данным о стоимости закупки, т.к. в противном случае валовую прибыль вычислить не составит труда, поскольку данные о стоимости реализации известны широкому кругу лиц. К тому, что относится к правам доступа, надо относиться очень аккуратно. А если у менеджеров по продажам мотивация построена на валовой прибыли, так эти требования еще и противоречат друг другу, т.к. менеджеры никогда не смогут это проверить. Если уж включать такое требование, то нужно указывать конкретные отчеты и объекты системы, в которых указывать, какая часть данных должны быть доступна отдельным</p>

		категориям лиц. И рассматривать каждый такой случай индивидуально.
Производительность	Отчет по продажам должен формироваться за 1 минуту.	Да, понятно. И даже есть конкретное ограничение по времени: 1 минута. Но не известно, какая детализация при этом предполагается: по каждому товару, группам товаров, клиентам или как-то еще? Можно сформулировать примерно так: «Отчет по продажам в разрезе клиентов с детализацией до каждой товарной позиции должен выводиться не более, чем за 1 минуту при условии, что количество товаров в выборке не превышает 5000 строк».

<https://www.klerk.ru/soft/articles/333385/>
<https://babok-school.ru/blogs/srs-and-other-standards-to-specify-requirements-as-technical-task/>
<https://babok-school.ru/blogs/how-to-write-srs-and-technical-task-without-standards/>
<https://babok-school.ru/blogs/srs-development-common-mistakes-of-analyst/>

Анализ требований

Анализ требований подразумевает их детализацию, гарантирующую, что требования понимают все заинтересованные лица, а также тщательное исследование требований на предмет ошибок, пробелов и других недостатков. Кроме того, анализ включает создание прототипов, анализ осуществимости и согласование приоритетов. Цель анализа — достаточно качественно и подробно описать требования, позволяющие менеджерам реалистично оценить все затраты на проект, а техническому персоналу — начать проектирование, сборку и тестирование.

Зачастую отдельные требования стоит представить несколькими способами, например, в текстовой и графической форме. Это позволит выявить их особенности и проблемы, не заметные при представлении одним способом. Также это помогает всем заинтересованным лицам выработать согласованное представление об итогах разработки продукта.

Рассмотрим основные виды деятельности на этом этапе.

Анализ осуществимости требований. Проанализируйте, насколько реально реализовать каждое требование при разумных затратах и с приемлемой производительностью в предполагаемой среде. Рассмотрите риски, связанные с реализацией каждого требования, включая конфликты с другими требованиями, зависимость от внешних факторов и препятствия технического характера.

Определение приоритетов требований. Определите относительные приоритеты реализации функций продукта, решаемых задач или отдельных требований. На основании приоритетов установите, в какой версии будет реализована та или иная функция или набор требований. Подтверждая изменения, распределите все их по конкретным версиям и включите в план выпуска этих версий затраты, необходимые на внесение изменений. В ходе работы над проектом стоит периодически переоценивать и корректировать приоритеты в соответствии с потребностями клиента, условиями рынка и бизнес-целями.

Моделирование требований. В отличие от подробной информации, представленной в спецификации требований к ПО или пользовательского интерфейса прототипа, графическая модель анализа отображает требования на высоком уровне абстракции. Модели позволяют выявить некорректные, несогласованные, отсутствующие и избыточные требования. К таким моделям относятся диаграммы потоков данных, диаграммы «сущность-связь», диаграммы конечных автоматов, карты экранов, диаграммы классов, диаграммы последовательностей, диаграммы взаимодействий, таблицы и деревья решений и многое другое.

Создание словаря терминов. В этом документе соберите определения всех элементов и структур данных, связанных с системой, что позволяет всем участникам проекта использовать согласованные определения данных. На стадии работы над требованиями словарь должен содержать определения элементов данных, относящихся к предметной области, чтобы клиентам и разработчикам было проще общаться.

Создание контекстной диаграммы. Контекстная диаграмма — простая модель анализа, отображающая место новой системы в соответствующей среде. Она определяет границы и интерфейсы между разрабатываемой системой и сущностями, внешними для этой системы, например, пользователями, устройствами и прочими информационными системами.

Создание пользовательского интерфейса и технических прототипов. Если разработчики или пользователи не совсем уверены насчет требований, создайте прототип — частичную, возможную или предварительную версию продукта, которая сделает концепции и возможности более осязаемыми. Оценка прототипа поможет всем заинтересованным лицам достичь взаимопонимания по решаемой проблеме.