



Тема 2

Тестирование требований

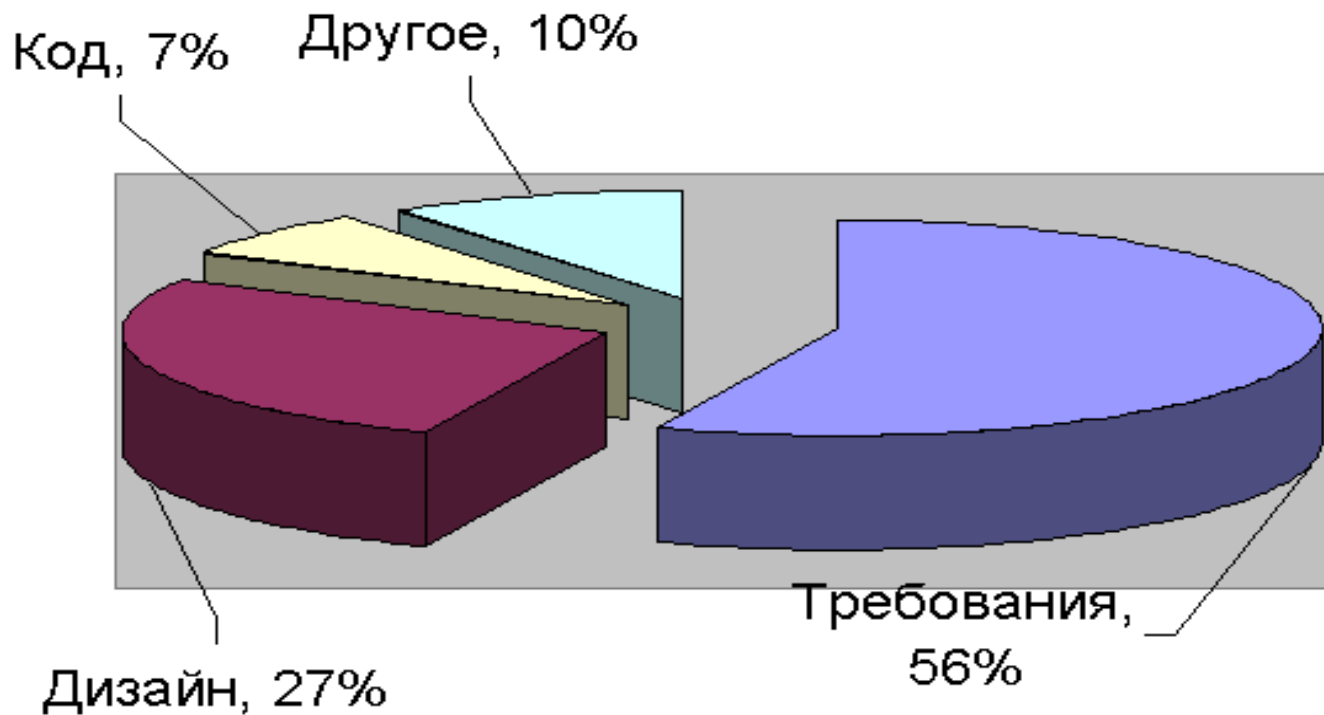
Вопросы:

1. Понятие , уровни и типы требований
2. Способы выявления и документирование требований
3. Характеристики требований

Важность тестирования документации



Требования. Там есть дефекты?



Получил от клиента письмо с правками в макет сайта: «То, чего здесь нет, оставляем так, как есть, но туда вносим правки». (С) bash.org.ru

Требования. Определение

Требования (requirements) – это подробное изложение функционального наполнения системы. Требования должны быть независимы от внутренней архитектуры приложения, т.е. должны описывать то, что необходимо заказчику без деталей реализации (принцип «**what, not how**»).

Требования к продукту и процессу

Требования к продукту. Цель заказчика - получить хороший конечный продукт: функциональный и удобный в использовании.

Требования к процессу. Вопросы формулирования требований к процессу, т.е. к тому, как разработчик будет выполнять работы по созданию целевой системы. Заказчик, вступая в договорные отношения с разработчиком, несёт различные риски, главными из которых является риск получить продукт с опозданием, либо ненадлежащего качества. Основные мероприятия по контролю и снижению риска - регламентация процесса создания программного обеспечения и его аудит.

Уровни требований

Уровень	Требования <i>В чём (как) выражаются</i>	Что описывают
1	<u>Бизнес требования</u> <i>Общее видение и обзорная документация</i>	Зачем вообще нужен ПП и что с его помощью будет делаться. Например "Нам нужен инструмент, извлекающих бизнес-информацию из различных источников и представляющий её в виде диаграмм и таблиц"
2	<u>Пользовательские требования</u> <i>Use Cases</i>	Задачи, которые пользователь может выполнять с помощью ПП, и реакцию ПП. Например: "Когда пользователь заходит в систему, должен появляться текст приветствия"
3	<u>Функциональные и нефункциональные требования</u> <i>Требования к ПО</i>	Функциональные требования описывают поведение системы, например: "В процессе инсталляции программа должна проверять остаток свободного места на всех несъёмных накопителях". Нефункциональные требования описывают специфические свойства системы, такие как: требования к удобству использования, безопасности, надёжности, расширяемости и т.д. Например: "При одновременной работе с системой 1000 пользователей, минимальное время между возникновением сбоя должно быть более или равно десяти тысячам часов".

Основные типы требований

- 1) **Функциональные** требования определяют, «что система должна делать».
- 2) **Нефункциональные** требования определяют, «с соблюдением каких условий» должны происходить действия системы (например, скорость отклика при выполнении заданной операции).

Типы: группа функциональных требований

Бизнес-требования (business requirements) – определяют высокоуровневые цели организации или клиента (потребителя) – заказчика разрабатываемого программного обеспечения.

Пользовательские требования (user requirements) – описывают цели/задачи пользователей системы, которые должны достигаться/выполняться пользователями при помощи создаваемой программной системы.

Способы выявления требований

Основными **способами выявления** требований являются:

- Интервью
- Наблюдение
- Самостоятельное описание
- Семинары
- Прототипирование

Документирование требований

Стандарт IEEE 830-1998 (см. стандарт «IEEE Standard 830-1998, «IEEE Recommended Practice for Software Requirements Specifications» в файле "IEEE Std 830-1998.pdf) содержит развёрнутое описание требований, которое может быть оптимизировано для нужд конкретной организации.

В описание требований рекомендуется включать:

1. Введение

1.1 Назначение документа.

1.2. Поддерживаемые соглашения.

1.3. Предполагаемая аудитория и рекомендации по последовательности работы с документом для каждого класса читателей.

1.4. Границы проекта. Здесь содержится ссылка на документ «Концепция», если таковой имеется, либо краткое резюме продукта.

1.5. Ссылки.

Документирование требований

2. Общее описание

2.1. **Общий взгляд на продукт.** Здесь необходимо определить, является ли описываемый продукт новым членом растущего семейства продуктов, новой версией существующей системы, заменой существующего приложения или совершенно новым продуктом. Если спецификация требований определяет компонент более крупной системы, укажите, как это ПО соотносится со всей системой и определите основные интерфейсы между ними.

2.2. **Особенности продукта.** Перечисляются ключевые особенности продукта или его главные свойства. Здесь уместно поместить контекстную диаграмму (в виде диаграммы вариантов использования, потоков данных или др. спецификаций).

2.3. **Классы и характеристики пользователей.** Выявляются все пользователи системы и осуществляется обобщение (выделение классов) пользователей. Найденные классы описываются (например — уровень квалификации, доступный функционал и т.д.).

Документирование требований

2. Общее описание (продолжение)

2.4. **Операционная среда.** Рассматривается среда функционирования программного продукта, включая аппаратные средства, операционные системы, для распределённых систем – географическое расположение пользователей и серверов, топология сети.

2.5. **Ограничения проектирования и реализации.** Рассмотрим классификацию ограничений:

- определённые технологии, средства, языки программирования и базы данных, которые следует использовать или избегать;
- ограничения, налагаемые операционной средой продукта;
- обязательные соглашения или стандарты разработки;
- обратная совместимость с продуктами, выпущенными ранее;
- ограничения, налагаемые бизнес-правилами;
- ограничения, связанные с оборудованием, например требования к быстродействию, ограничения памяти или процессора;
- соглашения, связанные с пользовательским интерфейсом существующего продукта, которые необходимо соблюдать при его улучшении;
- форматы и протоколы обмена данными.

2.6 **Документация для пользователей.**

2.7 **Предположения и зависимости**

Документирование требований

3. Функции системы

Для каждой i -й функции составляется следующее описание.

3.1 Наименование i -й функции системы.

3.1.1 Описание и приоритеты. Приводится краткое описание функции и указывается её приоритет (степень важности/очередности реализации).

3.1.2 Последовательности «воздействие – реакция». Необходимо перечислить последовательность воздействий, оказываемых на систему (действия пользователей, сигналы внешних устройств и др.), и отклики системы, определяющие реакцию конкретной функции.

3.1.3 Функциональные требования. Необходимо дать детализацию i -й функции, перечислить детализированные функциональные требования, включая реакцию на ожидаемые ошибки и неверные действия. Каждому детальному функциональному требованию присваивается уникальный идентификатор.

Документирование требований

4. Требования к внешнему интерфейсу

4.1 Интерфейсы пользователя (user interface, UI)

Основные характеристики UI:

- ссылки на стандарты графического интерфейса пользователей или стилевые рекомендации для семейства продукта, которые необходимо соблюдать;
- стандарты шрифтов, значков, названий кнопок, изображений, цветовых схем, последовательностей полей вкладок, часто используемых элементов управления и т.д.;
- конфигурация экрана или ограничения разрешения;
- стандартные кнопки, функции или ссылки перемещения, одинаковые для всех экранов (например, кнопка справки);
- быстрые клавиши («клавиатурные шорт-каты», «short-cut»);
- стандарты отображения сообщений;
- стандарты конфигурации для упрощения локализации ПО;
- специальные возможности для пользователей с проблемами со зрением.

Документирование требований

4. Требования к внешнему интерфейсу (продолжение)

4.2 Интерфейсы оборудования

Описываются характеристики каждого интерфейса между компонентами ПО и оборудованием системы. В описание могут входить типы поддерживаемых устройств, взаимодействие данных и элементов управления между ПО и оборудованием, а также протоколы взаимодействия.

4.3 Интерфейсы ПО

Описываются соединения продукта и других компонентов ПО (идентифицированные по имени и версии), в том числе базы данных, операционные системы, средства, библиотеки и интегрированные коммерческие компоненты.

Документирование требований

4. Требования к внешнему интерфейсу (продолжение)

4.4 Интерфейсы передачи информации

Указываются требования для любых функций взаимодействия, которые будут использоваться продуктом, включая электронную почту, веб-браузер, протоколы сетевого соединения и электронные формы. Определяются соответствующие форматы сообщений. Описываются особенности безопасности взаимодействия или шифрования, частоты передачи данных и механизмов синхронизации.

Документирование требований

5. Другие нефункциональные требования

В этом разделе описываются остальные нефункциональные требования, не относящиеся к требованиям к интерфейсу, которые представлены в разделе 4, и к ограничениям, описываемым в разделе 2.5.

5.1 Требования к производительности

Указываются специальные требования к производительности для различных системных операций. Обосновывается их необходимость для того, чтобы помочь разработчикам принять правильные решения, касающиеся дизайна.

Документирование требований

6. Приложения

Приложение А. Словарь терминов (гlossарий).

Приложение Б. Модели анализа. В этот раздел помещаются все модели, построенные в процессе анализа требований.

Приложение В. Список вопросов. Это динамический список ещё не решённых проблем, связанных с требованиями. Это могут быть элементы, помеченные как "TBD" («to be determined», «to be defined» – «будет определено», «необходимо определить»), отложенные решения, необходимая информация, неразрешённые конфликты и т.д.

Характеристики требований

Каждое требование должно быть:

Завершённым (complete). Все важные аспекты должны быть включены. Ничто не должно быть оставлено «для будущего определения» (2BD – to be defined).

Непротиворечивым (consistent). Требование не должно содержать противоречий как внутри себя, так и с другими требованиями.

Корректным (correct). Требование должно чётко указывать на то, что должно выполнять приложение.

Недвусмысленным (unambiguous). Требование не должно допускать разночтений.

Проверяемым (verifiable). Требование должно быть сформулировано так, чтобы существовали способы однозначной проверки – выполнено требование или нет.

Характеристики набора требований

Наборы требований должны быть:

Модифицируемыми (modifiable). Структура и стиль набора требований должны быть такими, чтобы набор требований можно было легко модифицировать. Должна отсутствовать избыточность. Должно быть построено корректное содержание всего документа.

Прослеживаемыми (traceable). У каждого требования должен быть уникальный идентификатор, по которому на это требование можно сослаться.

Проранжированными по важности, стабильности и срочности (ranked for importance, stability and priority). Для каждого требования должен быть указан уровень его важности (насколько оно важно для заказчика), стабильности (насколько высока вероятность, что это требование ещё будет изменено в процессе обсуждения деталей проекта) и срочности (как быстро требование должно быть реализовано).

Проблемы с требованиями

Проблемы незавершённости (неполноты). Упущены нефункциональные составляющие требования. Присутствуют скрытые предположения. Присутствуют слишком общие утверждения. **Примеры требований?**

Проблемы противоречивости. Требования противоречат явным образом друг другу (**пример?**) или части требований противоречат друг другу. Требование противоречит здравому смыслу

Проблемы некорректности. К некорректности приводят опечатки, последствия использования сору-paste. Некорректными являются требования, приводящие к избыточным расходам на разработку, при этом не дающие соответствующей отдачи. Технически невыполнимые требования.

Проблемы с требованиями

Проблемы двусмысленности. Требование должно однозначно трактоваться.

Проблемы непроверяемости. «В приложении должно быть ноль ошибок» – это гарантировать невозможно.

Проблемы непроранжированности. «Все требования одинаково важны».

Хорошие требования (и наборы требований) всегда проранжированы по трём показателям:

- **важности** (насколько требование важно);
- **приоритетности** выполнения (как быстро его надо реализовать);
- **стабильности** (какова вероятность, что требование изменится).

Проблемы с требованиями

Ещё раз самое важное

Если мы не можем проверить требование – это проблема. В конечном итоге именно тестировщики отвечают за то, чтобы требование было проверено.

Если мы не можем проверить требование по объективным причинам, мы уточняем его до тех пор, пока оно не станет проверяемым. В зависимости от ситуации, мы расспрашиваем заказчика, разработчиков, своих более опытных коллег и т.д.

В случае очень важных и нерешаемых простыми способами проблем, решение будет приниматься менеджером проекта и заказчиком.



Техники работы с требованиями

Одна из наиболее распространённых техник работы с требованиями – взаимный перепросмотр.

Суть взаимного перепросмотра требований проста: **после того, как один человек создал требование, другой человек это требование проверяет.**

Обычно, выделяют три уровня перепросмотра:

- 1) **Неформальный перепросмотр.** Двое коллег просто обмениваются листиками (файликами) и правят найденные ошибки.
- 2) **Технический перепросмотр.** Это немного более формализованный процесс, требующий подготовки, выделенного времени, участия некоторой группы специалистов (желательно, из различных областей).
- 3) **Формальная инспекция.** Проводится редко и в случае очень больших проектов и крайней необходимости. Описывается специальными стандартами, требует соблюдения широкого спектра правил и протоколирования результатов.

Важность тестирования требований

Хорошо проработанные требования позволяют:

- Выработать **общее понимание** между заказчиком и разработчиком.
- Определить **рамки проекта**.
- Более точно **определить финансовые и временные характеристики** проекта.
- **Обезопасить заказчика** от риска получить продукт, в котором он не сможет работать.
- **Обезопасить разработчика** от риска попасть в ситуацию «неконтролируемого размытия границ», которое может привести к непредвиденным затратам ресурсов сверх начальных ожиданий.

Тестирование требований в действии :)

