

Тема 2 «Документирование и тестирование требований»

Вопросы лекции:

1. Понятие требований. Виды, уровни и типы требований
2. Источники требований
3. Документирование требований
4. Тестирование требований

1-й вопрос: «Понятие требований. Виды, уровни и типы требований»

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»).

Определения понятия “требование” (на основе стандарта IEEE Standard Glossary of Software Engineering Terminology, 1990):

1. Условие или возможность, требуемая пользователем для решения задач или достижения целей.
2. Условие или возможность, которые должны удовлетворяться системой/компонентом системы или которыми система/компонент системы должна обладать для обеспечения условий контракта, стандартов, спецификаций или др. регулируемыми документами.
3. Документальная репрезентация (зафиксированное определение, описание) условий или возможностей, перечисленных в предыдущих пунктах.

Виды требований:

Требования к продукту. В своей основе требования - это то, что формулирует заказчик. Цель, которую он преследует – получить хороший конечный продукт: функциональный и удобный в использовании. Поэтому требования к продукту являются основополагающим классом требований.

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:

Обычно выделяют три уровня требований:

1 уровень. Бизнес-требования (*business requirements*). Пример бизнес-требования: система должна сократить срок оборачиваемости обрабатываемых на предприятии заказов в три раза. Бизнес-требования обычно формулируются топ-менеджерами, либо акционерами предприятия.

2 уровень. Уровень требований пользователей (*user requirements*). Пример требования пользователя: система должна представлять диалоговые средства для ввода исчерпывающей информации о заказе, последующей фиксации информации в базе данных и маршрутизации информации о заказе к сотруднику, отвечающему за его планирование и исполнение.

3 уровень. Функциональный (*functional requirements*). Пример функциональных требований (или просто функций) по работе с электронным заказом: заказ может быть создан, отредактирован, удалён и перемещён с участка на участок.

Типы требований:

1. Функциональные требования определяют, что ПС должно делать.
2. Нефункциональные требования определяют, с соблюдением каких условий ПС должно работать.

Группа функциональных требований (по уровням):

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.

Функциональные требования (*functional requirements*) – определяют функциональность (поведение) программной системы, которая должна быть создана разработчиками для предоставления возможности выполнения пользователями своих обязанностей в рамках бизнес-требований и в контексте пользовательских требований.

Группа нефункциональных требований:

Бизнес-правила (*business rules*) – включают или связаны с корпоративными регламентами, политиками, стандартами, законодательными актами, внутрикорпоративными инициативами, учётными практиками, алгоритмами вычислений и т.д.

Внешние интерфейсы (*external interfaces*) – «пользовательские интерфейсы.

Атрибуты качества (*quality attributes*) – описывают дополнительные характеристики продукта в различных измерениях, важных для пользователей и/или разработчиков.

Ограничения (*constraints*) – формулировки условий, модифицирующих требования или наборы требований, сужая выбор возможных решений по их реализации.

2-й вопрос: «Источники требований»

Основным источником требований являются соображения, высказанные представителями заказчика. В соответствии с иерархической моделью требований данная информация структурируется как минимум на два уровня: бизнес-требования и требования пользователей.

Другим важным источником информации, помимо выявления требований, являются артефакты, описывающие предметную область. Это могут быть документы с описанием бизнес-процессов предприятия, выполненные консалтинговым агентством, либо просто документы (должностные инструкции, распоряжения, своды бизнес-правил), принятые на предприятии.

Основными способами выявления требований являются:

- Интервью
- Наблюдение
- Самостоятельное описание
- Семинары
- Прототипирование

Один из классов методов, которые могут быть использованы для выявления требований, получил название *технологии FAST (Facilitated Application Specification Techniques* – упрощенной спецификации приложения).

Обычно во время сеанса технологии FAST рекомендуется следовать тому или иному набору условий из числа приведенных ниже:

- И заказчики, и разработчики принимают участие в совещании, посвященном вопросам определения требований.
- Установлены правила подготовки и участия в совещании, которым следуют обе стороны. Должна быть установлена атмосфера, содействующая успеху совещания.
- Совещание проводится под управлением посредника. Таким посредником может быть заказчик, разработчик или кто-то посторонний, приемлемый для обеих сторон.
- Для фиксации требований используются такие средства, как лекционные плакаты с рейкой, настенные индикаторные панели или простая ленточная бумага.
- Цель совещания — определить задачи или потребности заказчика, предложить решение, обсудить различные подходы и определить набор требований.

3-й вопрос: «Документирование требований»

Документ определения требований содержит все требования, предъявляемые к системе и сформулированные на естественном языке, благодаря чему они становятся понятными и заказчику, и тем, кто принимает участие в разработке проекта.

Определения требований должны обладать следующими свойствами:

- Каждое требование должно быть снабжено уникальным идентификатором, чтобы можно было однозначно ссылаться на него при планировании тестового покрытия, при проектировании тестовых случаев и в отчетах по результатам тестирования.
- Требования должны быть представлены с точки зрения пользователя системы. Этот принцип препятствует появлению требований, затрагивающих внутренние свойства системы и требующих детальнх знаний программного кода для успешного тестирования. Такие требования должны возникать на поздних стадиях разработки и охватываться модульным тестированием и проверкой взаимодействия и функционирования компонентов системы.
- Должны быть включены как *функциональные (functional)*, так и *нефункциональные (nonfunctional)* требования.
- Документ определения требований должен находиться под управлением конфигурациями. Это, по меньшей мере, означает, что данный документ подпадает под управление версиями и что все версии документа должны быть помещены в безопасное хранилище, подобное, например, каталогу, содержимое которого обычно дублируется. Если требования подвергаются изменениям, мы должны иметь возможность проследить, чтобы соответствующие изменения были внесены в тестовые случаи системных и приемочных испытаний.

Оглавление одного из возможных вариантов документа, содержащего системные требования, согласно стандарту *IEEE Standard 830-1998: The IEEE Guide to Software Requirements Specifications* – Руководящие принципы IEEE по составлению спецификации требований к программному обеспечению, представлено на рисунке 1.

Документ определения требований (Requirements Definition Document)	
Оглавление (Table of content)	
1. Введение (Introduction)	4
1.1. Назначение (Purpose).....	4
1.2. Область применения (Scope).....	4
1.3. Обзор (Overview).....	4
2. Общее описание (General Description)	4
2.1. Перспектива продукта (Product perspective).....	4
2.2. Функция продукта (Product function).....	4
2.3. Характеристика пользователя (User Characteristics).....	4
2.4. Ограничения общего характера (General Constraints).....	4
2.5. Допущения и зависимости (Assumption and Dependencies).....	5
3. Особые требования (Specific Requirements)	5
3.1. Функциональные требования (Functional Requirements).....	5
3.2. Требования внешнего интерфейса (External Interface Requirements).....	5
3.3. Требования к производительности (Performance Requirements).....	5
3.4. Проектные ограничения (Design constraints).....	5
3.5. Атрибуты (Attributes).....	6
3.6. Другие требования (Other Requirements).....	6
Ссылки (References)	7
Приложение 1. Сокращения (Appendix 1 — Acronyms)	7
Приложение 2. Терминология (Appendix 1 — Definition of Terms)	7

Рисунок 1 – Пример оглавления документа определения требований к программному продукту

Спецификация требований описывает то же, что и документ определения требований, однако спецификация предназначена для системных разработчиков и представлена на языке или в обозначениях, ориентированных на разработчиков. Спецификацию требований часто называют системной функциональной спецификацией, несмотря на то что ее область действия распространяется за пределы на функциональных требований системы.

Назначение матрицы прослеживаемости требований заключается в отображении каждого требования на проектные компоненты, программные коды и тестовые случаи. Пример матрицы прослеживаемости требований показан на рисунке 2. В этом примере каждое требование, фигурирующее в документе определения требований, отображается на одно или большее количество требований из спецификации, на проектные компоненты и программные коды, а также на тестовые случаи модульного тестирования, проверки взаимодействия и функционирования компонентов программного продукта, системных и приемочных испытаний. Для нумерации требований используется "десятичная нотация Дьюи". Эта нотация позволяет отобразить исходные требования из документа определения требований на производные требования и различные виды тестов в отношении "один-ко-многим" (например, одно требование соотносится с некоторым множеством системных тестов). Даже если проектная информация и данные о программных кодах не включены в эту матрицу, полезно поддерживать ее упрощенный вариант, который соотносит каждое требование с соответствующими системными и приемочными тестами.

Эти идентификаторы отображают требования из документа определения требований на требования из спецификации требований.

Эти идентификаторы отображают все тесты обратно на требования.

Идентификатор требования в документе определения требований	Идентификатор требования в спецификации требований	Идентификатор проектного компонента	Идентификатор программного компонента	Идентификатор тестового случая на этапе модульного тестирования	Идентификатор тестового случая на этапе проверки взаимодействия компонентов продукта	Идентификатор тестового случая на этапе системных испытаний	Идентификатор тестового случая на этапе приемочных испытаний
RD2 2 4	RS2 2 4 1	D 2 2 4 1	CC2 2 4 1	UT2 2 4 1	IT2 2 4	ST2 2 4	AT2 2 4
RD2 2 4	RS2 2 4 2	D2 2 4 2	CC2 2 4 2	UT2 2 4 2	IT2 2 4	ST2 2 4	AT2 2 4
RD2 2 4	RS2 2 4 3	D2 2 4 3	CC2 2 4 3	UT2 2 4 3	IT2 2 4	ST2 2 4	AT2 2 4
RD2 2 4	RS2 2 4 4	D2 2 4 4	CC2 2 4 4	UT2 2 4 4	IT2 2 4	ST2 2 4	AT2 2 4

Эти идентификаторы прослеживают происхождение всех проектных и программных компонентов до соответствующих требований.

Рисунок 2 – Примерный формат матрицы отслеживания требований

4-й вопрос: «Тестирование требований»

Завершающим действием процесса формулирования требований является статическое тестирование требований. Одна из целей статического тестирования предусматривают проверку требований на полноту, непротиворечивость, осуществимость, а также на возможность тестирования (статического или динамического) в процессе реализации. Дополнительная цель такого статического тестирования заключается в проверке того, что требования, представленные в окончательном виде, соответствуют пожеланиям заказчика, выраженных им на стадии выявления требований. Правильно выполненное статическое тестирование оборачивается серьезным выигрышем в смысле экономии средств и времени. Статическое тестирование требований представляет собой наиболее эффективный способ обнаружения дефектов еще до того, как они окажут неблагоприятное влияние на планы выполнения работ и сметные расходы.

Характеристики требований:

- Полнота.
- Ясность.
- Корректность.
- Согласованность (непротиворечивость).
- Верифицируемость (проверяемость).
- Необходимость.
- Полезность при эксплуатации.
- Осуществимость (выполнимость).
- Модифицируемость.
- Трассируемость (прослеживаемость).
- Упорядоченность по важности стабильности.
- Наличие количественных метрик.

Полнота отдельного требования – свойство, означающее, что текст требования не требует дополнительной детализации, то есть в нём предусмотрены все необходимые нюансы, особенности и детали данного требования.

Полнота системы требований – свойство, означающее, что совокупность артефактов, описывающих требования, исчерпывающим образом описывает всё то, что требуется от разрабатываемой системы.

Требование обладает свойством **ясности**, если оно сходным образом воспринимается всеми участниками проекта.

Корректность требования определяется через точность описания функциональности. В этом смысле корректность в определённой степени конкурирует с полнотой. Но есть и различие: если свойство полноты носит скорее качественный характер: абсолютная полнота представляет недостижимый идеал, к которому можно приближаться, то свойство корректности носит оценочный характер и задаёт дихотомию: **каждое из требований либо корректно, либо нет.**

Свойство **верифицируемости** существенно связано со свойствами **ясности** и **полноты**: если требование изложено на языке, понятном и одинаково воспринимаемым участниками проекта, причём оно является полным, т.е. ни одна из важных для реализации деталей не упущена – значит, это требование **можно проверить.**

Необходимыми следует считать свойства, без выполнения которых невозможно, либо затруднено выполнение автоматизированных бизнес-функций пользователей; **полез-**

ными при эксплуатации следует считать любые свойства, повышающие эргономические качества продукта.

Выполнимость требования на практике определяется разумным балансом между ценностью (степенью необходимости и полезности) и требуемыми ресурсами.

Модифицируемость означает, что требования должны быть грамотно сгруппированы таким образом, чтобы при изменении какого-то требования (или группы требований) не приходилось изменять требования (или группы требований) в иных местах списка требований.

Трассируемость требования определяется возможностью отследить связь между ним и другими артефактами проекта (документами, моделями, текстами программ и пр.).

Приоритет требования представляет собой количественную оценку степени **значимости (важности)** требования. **Стабильность** требования характеризует прогнозную оценку неизменности требований во времени. Упорядоченность по стабильности позволяет проектной команде в первую очередь заниматься реализацией тех требований, вероятность пересмотра (изменения или даже исключения) которых минимальна.

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 выполняться приложение.

Недвусмысленным (unambiguous). Требование не должно допускать разночтений.

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

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).

Техники работы с требованиями.

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

Литература

Основная:

1. Браун, К. Быстрое тестирование / К. Браун, Р. Калбертсон, Гэри Кобб. – М.: изд-во «Вильямс». – 2002. – 384 с.
2. Котляров, В.П. Основы тестирования программного обеспечения / В.П. Котляров. – Интернет-университет информационных технологий – ИНТУИТ.ру. – 2006. – 285 с.
3. Бейзер, Б. Тестирование черного ящика. Технологии функционального тестирования ПО. Серия: Библиотека программиста. / Борис Бейзер. – Спб.: Питер. 2004. – 320 с.

Дополнительная:

1. Савин, Р. Тестирование Дот Ком, или Пособие по жестокому обращению с багами в интернет-стартапах / Роман Савин. – Дело, 2007. – 312 с.
2. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – М.: ДиаСофт. – 2001. – 544 с.
3. Куликов, С.С. Основы тестирования программного обеспечения. / С.С. Куликов, О.Г. Смолякова. – Мн.: ЦОТ «Белхард». – 2008.–

Контрольные вопросы по теме:

1. Дайте определение понятия «требование».
2. В чём разница требований к продукту и процессу? Дайте определения этих видов требований.
3. Перечислите основные источники требований.
4. Назовите уровни требований.
5. Какие типы требований вы знаете?
6. Какие требования относятся к группе функциональных требований?
7. Какие требования относятся к группе нефункциональных требований?
8. Назовите основные способы выявления требований. Каковы преимущества и недостатки каждого способа?
9. Назовите ключевые разделы требований согласно стандарту IEEE 830-1998.
10. Назовите характеристики хорошего требования.
11. Перечислите характеристики хороших наборов требований.
12. Какие техники работы с требованиями Вы знаете?