

УВАЖАЕМЫЕ СТУДЕНТЫ! Изучите теоретические сведения к лабораторной работе, выполните практическое задание, дайте ответы на контрольные вопросы.

Результаты работы, фотоотчет, предоставить преподавателю на e-mail: r.bigangel@gmail.com **до 13.02.2023.**

Требования к отчету:

Отчет предоставляется преподавателю в электронном варианте и должен содержать:

- название работы, постановку цели, вывод;
- ответы на контрольные вопросы, указанные преподавателем.

При возникновении вопросов по приведенному материалу обращаться по следующему номеру телефона: (072)111-37-59, (Viber, WhatsApp), vk.com: <https://vk.com/daykini>

ВНИМАНИЕ!!! При отправке работы, не забывайте указывать *ФИО студента, наименование дисциплины, дата проведения занятия (по расписанию).*

Лабораторная работа №28

Тема: Критерии качества тестирования программного обеспечения. Аксиомы тестирования.

Цель: изучить критерии качества требований, выполнить тестирование требований к программному обеспечению.

План занятия:

1. Изучить теоретические сведения.
2. Выполнить практическое задание по лабораторной работе.
3. Оформить отчёт и ответить на контрольные вопросы.

Теоретические сведения

Качество программного обеспечения во многом зависит от качества сформированных требований, т.к. требования к программному продукту являются базой для разработки и последующего тестирования.

Тестирование требований выполняется на предмет их соответствия

критериям качества требований (рисунок 28.1).



Рисунок 28.1 – Критерии качества требований

Завершённость (completeness). Требование является полным и законченным с точки зрения представления в нём всей необходимой информации, ничто не пропущено по соображениям «это и так всем понятно».

Типичные проблемы с завершённостью:

– Отсутствуют нефункциональные составляющие требования или ссылки на соответствующие нефункциональные требования (например: «пароли должны храниться в зашифрованном виде», а каков алгоритм шифрования?).

– Указана лишь часть некоторого перечисления (например: «экспорт осуществляется в форматы PDF, PNG и т.д.», а что следует понимать под «и т.д.»?).

– Приведённые ссылки неоднозначны (например: «см. выше» вместо «см. раздел 123.45.b»).

Атомарность, единичность (atomicity). Требование является

атомарным, если его нельзя разбить на отдельные требования без потери завершённости и оно описывает одну и только одну ситуацию.

Типичные проблемы с атомарностью:

– В одном требовании, фактически, содержится несколько независимых (например: «кнопка “Restart” не должна отображаться при остановленном сервисе, окно “Log” должно вмещать не менее 20-ти записей о последних действиях пользователя»: здесь в одном предложении описаны совершенно разные элементы интерфейса в совершенно разных контекстах).

– Требование допускает разночтение в силу грамматических особенностей языка (например: «если пользователь подтверждает заказ и редактирует заказ или откладывает заказ, должен выдаваться запрос на оплату»: здесь описаны три разных случая, и это требование стоит разбить на три отдельных требования во избежание путаницы). Такое нарушение атомарности часто влечёт за собой возникновение противоречивости.

– В одном требовании объединено описание нескольких независимых ситуаций (например: «когда пользователь входит в систему, должно отображаться приветствие; когда пользователь вошёл в систему, должно отображаться имя пользователя; когда пользователь выходит из системы, должно отображаться прощание»: все эти три ситуации заслуживают того, чтобы быть описанными отдельными и более детальными требованиями).

Непротиворечивость, последовательность (consistency). Требование не должно содержать внутренних противоречий и противоречий другим требованиям и документам.

Типичные проблемы с непротиворечивостью:

– Противоречия внутри одного требования (например: «после успешного входа в систему пользователя, не имеющего права входить в систему...»: а как пользователь вошёл в систему, если не имел такого права?).

– Противоречия между двумя и более требованиями, между таблицей и текстом, рисунком и текстом, требованием и прототипом и т.д. (например:

«712.a Кнопка “Close” всегда должна быть красной» и «36452.x Кнопка “Close” всегда должна быть синей»: так всё же красной или синей?).

– Использование неверной терминологии или использование разных терминов для обозначения одного и того же объекта или явления (например: «в случае, если разрешение окна составляет менее 800x600...»: разрешение есть у экрана, у окна есть размер).

Недвусмысленность (unambiguousness, clearness). Требование описано без использования жаргона, неочевидных аббревиатур и расплывчатых формулировок и допускает только однозначное объективное понимание. Требование атомарно в плане невозможности различной трактовки сочетания отдельных фраз.

Типичные проблемы с недвусмысленностью:

– Использование терминов или фраз, допускающих субъективное толкование (например: «приложение должно поддерживать передачу больших объёмов данных»: насколько «больших»?) Вот лишь небольшой перечень слов и выражений, которые можно считать верными признаками двусмысленности: адекватно, быть способным, легко, обеспечивать, как минимум, быть способным, эффективно, своевременно, применимо, если возможно, будет определено позже, по мере необходимости, если это целесообразно, но не ограничиваясь, быть способно, иметь возможность, нормально, минимизировать, максимизировать, оптимизировать, быстро, удобно, просто, часто, обычно, большой, гибкий, устойчивый, по последнему слову техники, улучшенный, результативно.

– Использование неочевидных или двусмысленных аббревиатур без расшифровки (например: «доступ к ФС осуществляется посредством системы прозрачного шифрования» и «ФС предоставляет возможность фиксировать сообщения в их текущем состоянии с хранением истории всех изменений»: ФС здесь обозначает файловую систему или какой-нибудь «Фиксатор Сообщений»?)

– Формулировка требований из соображений, что нечто должно быть

всем очевидно (например: «Система конвертирует входной файл из формата PDF в выходной файл формата PNG» и при этом автор считает совершенно очевидным, что имена файлов система получает из командной строки, а многостраничный PDF конвертируется в несколько PNG-файлов, к именам которых добавляется

«page-1», «page-2» и т.д.). Эта проблема перекликается с нарушением корректности.

Выполнимость (feasibility). Требование технологически выполнимо и может быть реализовано в рамках бюджета и сроков разработки проекта.

Типичные проблемы с выполнимостью:

– Так называемое «озолочение» (gold plating) — требования, которые крайне долго и/или дорого реализуются и при этом практически бесполезны для конечных пользователей (например: «настройка параметров для подключения к базе данных должна поддерживать распознавание символов из жестов, полученных с устройств трёхмерного ввода»).

– Технически нереализуемые на современном уровне развития технологий требования (например: «анализ договоров должен выполняться с применением искусственного интеллекта, который будет выносить однозначное корректное заключение о степени выгоды от заключения договора»).

– В принципе нереализуемые требования (например: «система поиска должна заранее предусматривать все возможные варианты поисковых запросов и кэшировать их результаты»).

Обязательность, нужность (obligation) и актуальность (up-to-date). Если требование не является обязательным к реализации, оно должно быть просто исключено из набора требований. Если требование нужное, но «не очень важное», для указания этого факта используется указание приоритета. Также исключены (или переработаны) должны быть требования, утратившие актуальность.

Типичные проблемы с обязательностью и актуальностью:

– Требование было добавлено «на всякий случай», хотя реальной потребности в нём не было и нет.

– Требованию выставлены неверные значения приоритета по критериям важности и/или срочности.

– Требование устарело, но не было переработано или удалено.

Прослеживаемость (traceability). Прослеживаемость бывает вертикальной и горизонтальной. Вертикальная позволяет соотносить между собой требования на различных уровнях требований, горизонтальная позволяет соотносить требование с тест-планом, тест-кейсами, архитектурными решениями и т.д.

Для обеспечения прослеживаемости часто используются специальные инструменты по управлению требованиями и/или матрицы прослеживаемости.

Типичные проблемы с прослеживаемостью:

– Требования не пронумерованы, не структурированы, не имеют оглавления, не имеют работающих перекрёстных ссылок.

– При разработке требований не были использованы инструменты и техники управления требованиями.

– Набор требований неполный, носит обрывочный характер с явными «пробелами».

Модифицируемость (modifiability). Это свойство характеризует простоту внесения изменений в отдельные требования и в набор требований. Можно говорить о наличии модифицируемости в том случае, если при доработке требований искомую информацию легко найти, а её изменение не приводит к нарушению иных описанных в этом перечне свойств.

Типичные проблемы с модифицируемостью:

– Требования неатомарны (см. «атомарность») и непрослеживаемы, а потому их изменение с высокой вероятностью порождает противоречивость.

– Требования изначально противоречивы. В такой ситуации внесение изменений (не связанных с устранением противоречивости) только

усугубляет ситуацию, увеличивая противоречивость и снижая прослеживаемость.

– Требования представлены в неудобной для обработки форме (например, не использованы инструменты управления требованиями, и в итоге команде приходится работать с десятками огромных текстовых документов).

Проранжированность по важности, стабильности, срочности (ranked for importance, stability, priority). Важность характеризует зависимость успеха проекта от успеха реализации требования. Стабильность характеризует вероятность того, что в обозримом будущем в требование не будет внесено никаких изменений. Срочность определяет распределение во времени усилий проектной команды по реализации того или иного требования.

Типичные проблемы с проранжированностью состоят в её отсутствии или неверной реализации и приводят к следующим последствиям.

Проблемы с проранжированностью по важности повышают риск неверного распределения усилий проектной команды, направления усилий на второстепенные задачи и конечного провала проекта из-за неспособности продукта выполнять ключевые задачи с соблюдением ключевых условий.

Проблемы с проранжированностью по стабильности повышают риск выполнения бессмысленной работы по совершенствованию, реализации и тестированию требований, которые в самое ближайшее время могут претерпеть кардинальные изменения (вплоть до полной утраты актуальности).

Проблемы с проранжированностью по срочности повышают риск нарушения желаемой заказчиком последовательности реализации функциональности и ввода этой функциональности в эксплуатацию.

Корректность (correctness) и проверяемость (verifiability). Фактически эти свойства вытекают из соблюдения всех вышеперечисленных (или можно сказать, что они не выполняются, если нарушено хотя бы одно из вышеперечисленных). В дополнение можно отметить, что проверяемость

подразумевает возможность создания объективного тест-кейса (тест-кейсов), однозначно показывающего, что требование реализовано верно и поведение приложения в точности соответствует требованию.

К типичным проблемам с корректностью также можно отнести:

- опечатки (особенно опасны опечатки в аббревиатурах, превращающие одну осмысленную аббревиатуру в другую также осмысленную, но не имеющую отношения к некоему контексту; такие опечатки крайне сложно заметить);

- наличие неаргументированных требований к дизайну и архитектуре;

- плохое оформление текста и сопутствующей графической информации,

- грамматические, пунктуационные и иные ошибки в тексте;

- неверный уровень детализации (например, слишком глубокая детализация требования на уровне бизнес-требований или недостаточная детализация на уровне требований к продукту);

- требования к пользователю, а не к приложению (например: «пользователь должен быть в состоянии отправить сообщение»: мы не можем влиять на состояние пользователя).

Техники тестирования требований.

1. Одной из наиболее активно используемых техник анализа требований является просмотр или рецензирование. Данная техника может быть реализована в форме:

- беглого просмотра (показ автором своей работы коллеге; самый быстрый, самый дешёвый и наиболее широко используемый вид просмотра);

- технического просмотра (выполняется группой специалистов, каждый из которых представляет свою область знаний: просматриваемый продукт не может считаться достаточно качественным, пока хотя бы у одного просматривающего остаются замечания);

- формальной инспекцией (структурированный, систематизированный и документируемый подход к анализу документации, для выполнения

которого привлекается большое количество специалистов, само выполнение занимает достаточно много времени, и потому этот вариант просмотра используется достаточно редко: как правило, при получении на сопровождение и доработку проекта, созданием которого ранее занималась другая компания).

2. Следующей техникой тестирования и повышения качества требований является (повторное) использование такой техники выявления требований, как формулировка вопросов. Если хоть что-то в требованиях вызывает непонимание или подозрение – задавайте вопросы.

Хорошее требование является проверяемым, а значит, должны существовать объективные способы определения того, верно ли реализовано требование. Продумывание чек-листов или даже полноценных тест-кейсов в процессе анализа требований позволяет определить, насколько требование проверяемо. Помимо использования для тестирования требований в дальнейшем такие чек-листы и тест-кейсы могут составить основу тестовой документации.

3. Рисунки, схемы.

Чтобы увидеть общую картину требований целиком, очень удобно использовать рисунки, схемы, диаграммы, интеллект-карты и т.д. Графическое представление удобно одновременно своей наглядностью и краткостью (например, UML-схема базы данных, занимающая один экран, может быть описана несколькими десятками страниц текста).

4. Исследование поведения и прототипирование.

Можно сказать, что прототипирование часто является следствием создания графического представления и анализа поведения системы. С использованием специальных инструментов можно очень быстро сделать наброски пользовательских интерфейсов, оценить применимость тех или иных решений и даже создать не просто «прототип ради прототипа», а заготовку для дальнейшей разработки, если окажется, что реализованное в прототипе (возможно, с небольшими доработками) устраивает заказчика.

Практическое задание:

1. Найти в сети интернет спецификацию с требованиями к программному продукту.
2. Протестировать спецификацию методом просмотра на предмет соответствия критериям качества требований.
3. Для обнаруженных дефектов указать, какой критерий качества нарушен, и аргументировать свою точку зрения.
4. Для обнаруженных дефектов сформулировать уточняющие вопросы к заказчику для выработки качественных требований.

Содержание отчета:

1. Цель работы.
2. Отчет по тестированию спецификации.
3. Выводы по работе.

Контрольные вопросы:

1. Как выглядит жизненный цикл проекта?
2. Какие выделяют критерии качества?
3. Какие требования считаются проверяемыми?
4. Какие требования считаются модифицируемыми?
5. Какие требования считаются корректными?
6. Какие требования считаются недвусмысленными?
7. Какие требования считаются полными?
8. Какие требования считаются непротиворечивыми?
9. Какие требования считаются упорядоченными по важности и стабильности?
10. Какие требования считаются трассируемыми?
11. Какие существуют методы тестирования требований?