**Анализ и тестирование требований.**

**Анализ требований:**

Анализ требований — это процесс детального изучения и понимания требований к программному обеспечению (ПО), предоставленных заказчиком или пользователем. Цель анализа — выявить неясности, противоречия, неполноту или нереалистичность требований.

Задачи анализа требований:

* Сбор требований: Получение информации о потребностях заказчика и пользователей.
* Документирование требований: Формализация требований в виде спецификаций, пользовательских историй, диаграмм и т.д.
* Проверка требований: Убеждение в том, что требования являются полными, непротиворечивыми, тестируемыми и реализуемыми.
* Уточнение требований: Внесение изменений в требования на основе анализа и обратной связи от заинтересованных сторон.
* Приоритизация требований: Определение приоритетности требований для планирования разработки и тестирования.

**Тестирование требований:**

Тестирование требований — это процесс проверки того, что требования к ПО являются тестируемыми и могут быть реализованы. Тестирование требований начинается на ранних этапах разработки, еще до написания кода.

Задачи тестирования требований:

* Определение тестовых сценариев: Разработка тестовых сценариев на основе требований.
* Проверка реализуемости требований: Убеждение в том, что требования могут быть реализованы с использованием доступных технологий и ресурсов.
* Выявление дефектов в требованиях: Обнаружение неясностей, противоречий, неполноты или нереалистичности требований.
* Оценка покрытия требований тестами: Определение, насколько полно требования покрыты тестами.
* Сообщение об ошибках в требованиях: Передача информации об ошибках в требованиях разработчикам для исправления.

**Анализ требований включает:**

* Сбор требований: обсуждения с клиентами и пользователями для выявления их потребностей и ожиданий.
* Документирование требований: создание спецификаций, диаграмм и других документов, которые подробно описывают требования.
* Анализ на предмет выполнимости: оценка того, насколько требования реалистичны и осуществимы.
* Приоритизация требований: определение наиболее важных и критичных требований.

**Тестирование требований предполагает:**

* Верификацию требований: проверка на соответствие требованиям стандартам и спецификациям.
* Валидацию требований: обеспечение того, что требования правильно отражают потребности пользователей.
* Создание тест-кейсов: разработка сценариев тестирования, основанных на требованиях.
* Исполнение тест-кейсов: выполнение тестов и документирование результатов.

\_\_\_\_\_

Анализ требований - это процесс изучения и понимания требований, предъявляемых к программному обеспечению.

Тестирование требований - это процесс проверки требований на соответствие их описанию и правильность их реализации.

\_\_\_\_\_

**1. Анализ Требований (Requirements Analysis)**

**Цель:**

Понять потребности заинтересованных сторон (заказчиков, пользователей, бизнеса).

Собрать, задокументировать и детализировать требования к системе.

Убедиться, что требования полные, непротиворечивые, однозначные, реализуемые и тестируемые.

**Процесс:**

Сбор требований (Requirements Elicitation):

Идентификация заинтересованных сторон.

Проведение интервью, опросов, семинаров (workshop), мозговых штурмов.

Анализ существующих систем, документации, бизнес-процессов.

Документирование требований:

Создание спецификации требований (SRS - Software Requirements Specification) или другого документа, описывающего функциональные и нефункциональные требования.

Использование различных форматов: текстовое описание, Use Case Diagram, User Stories, Acceptance Criteria.

Анализ и уточнение требований:

Проверка требований на полноту, непротиворечивость, однозначность.

Разрешение конфликтов между требованиями.

Приоритизация требований (например, с использованием MoSCoW – Must have, Should have, Could have, Won’t have).

Валидация требований:

Представление требований заинтересованным сторонам для получения обратной связи и подтверждения.

Проведение обзоров требований (Requirements Reviews).

**Типы требований:**

Функциональные требования (Functional Requirements): Описывают, что система должна делать. (Например, “Система должна позволять пользователю зарегистрироваться”).

Нефункциональные требования (Non-Functional Requirements): Описывают как система должна работать, её характеристики. (Например, “Система должна отвечать на запросы пользователей в течение 2 секунд”). К ним относятся:

Производительность (Performance)

Надежность (Reliability)

Безопасность (Security)

Удобство использования (Usability)

Масштабируемость (Scalability)

Переносимость (Portability)

Поддержка (Maintainability)

Бизнес-требования (Business Requirements): Описывают цели и задачи бизнеса, которые система должна поддерживать.

Пользовательские требования (User Requirements): Описывают потребности пользователей системы.

**2. Тестирование Требований (Requirements Testing)**

**Цель:**

Проверить качество требований: полноту, непротиворечивость, однозначность, тестируемость, реализуемость.

Убедиться, что требования соответствуют бизнес-потребностям.

Выявить дефекты в требованиях на ранних стадиях разработки.

**Процесс:**

Разработка тест-кейсов на основе требований: Каждое требование должно быть покрыто тест-кейсами.

Проведение статического тестирования:

Обзоры требований (Requirements Reviews): Участники (аналитики, разработчики, тестировщики) читают и анализируют требования, выявляя недостатки.

Инспекции требований: Более формальный процесс обзора, с четкими ролями и этапами.

Проведение динамического тестирования (на более поздних этапах):

Выполнение тест-кейсов для проверки реализации требований.

Использование трассировки требований (Requirements Traceability Matrix) для отслеживания связи между требованиями, тест-кейсами и компонентами системы.

**Методы тестирования требований:**

Обзоры требований (Requirements Reviews): Наиболее распространенный и эффективный метод.

Разработка прототипов: Создание прототипа системы для проверки понимания требований.

Матрица трассировки требований (Requirements Traceability Matrix - RTM): Инструмент для отслеживания связи между требованиями, проектной документацией, кодом и тест-кейсами. Обеспечивает полное покрытие требований тестами.

Use Case Testing: Тестирование на основе Use Case диаграмм, фокусируясь на взаимодействии пользователя с системой.

Boundary Value Analysis (BVA): Анализ граничных значений для проверки валидации данных. Используется для функциональных требований, связанных с вводом данных.

Equivalence Partitioning: Разбиение входных данных на классы эквивалентности и тестирование только одного значения из каждого класса.

**Что тестировать в требованиях:**

Полнота: Убедиться, что все необходимые требования указаны.

Непротиворечивость: Убедиться, что требования не противоречат друг другу.

Однозначность: Убедиться, что каждое требование имеет только одно толкование.

Тестируемость: Убедиться, что для каждого требования можно разработать тест-кейсы.

Реализуемость: Убедиться, что требование может быть реализовано с учетом доступных ресурсов и технологий.

Прослеживаемость: Убедиться, что каждое требование можно отследить от его источника до реализации и тестирования.

\_\_\_\_\_

**Разработать модульные тесты для программного модуля ReadAndWrite, содержащий метод чтения всех строк из файла. Код для программного модуля находится «Resorce/ModuleC#/». Оформить тест-кейсы для модульного тестирования.**