

## Практическая работа №5

### Цель работы:

Получение начальных знаний в области тестирования документации и требований заказчика

### Свойства качественного требования

1. Завершённость (completeness) Требование должно быть полным и законченным, без пропусков информации.

Типичные проблемы:

- Отсутствие нефункциональных требований (например, алгоритм шифрования).
- Неполные перечисления (например, "и т.д.").
- Неоднозначные ссылки (например, "см. выше").

2. Атомарность (atomicity) Требование должно описывать одну и только одну ситуацию.

Типичные проблемы:

- Объединение нескольких независимых требований в одно.
- Грамматические особенности, допускающие разночтения.
- Описание нескольких ситуаций в одном требовании.

3. Непротиворечивость (consistency) Требование не должно содержать внутренних противоречий и противоречий другим требованиям.

Типичные проблемы:

- Противоречия внутри одного требования.
- Противоречия между требованиями.
- Использование разных терминов для одного объекта.

4. Недвусмысленность (unambiguousness) Требование должно быть описано четко, без двусмысленностей.

Типичные проблемы:

- Использование субъективных терминов (например, "большой объем данных").
- Неочевидные аббревиатуры без расшифровки.
- Предположение, что нечто "очевидно".

5. Выполнимость (feasibility) Требование должно быть реализуемо в рамках бюджета и сроков.

Типичные проблемы:

- "Озолочение" — требования, которые дорого реализовать, но они бесполезны.
- Технически нереализуемые требования.
- Нереализуемые в принципе требования.

6. Обязательность и актуальность (obligatoriness and up-to-date) Требование должно быть необходимым и актуальным.

Типичные проблемы:

- Требования, добавленные "на всякий случай".
- Неверные приоритеты требований.
- Устаревшие требования.

7. Прослеживаемость (traceability) Требования должны быть связаны с тест-планами, архитектурными решениями и т.д.

Типичные проблемы:

- Отсутствие нумерации, структуры, оглавления.
- Неполный набор требований.

8. Модифицируемость (modifiability) Требования должны легко изменяться без нарушения других свойств.

Типичные проблемы:

- Неатомарные и непрослеживаемые требования.
- Противоречивые требования.

9. Проранжированность по важности, стабильности, срочности (ranked for importance, stability, priority) Требования должны быть ранжированы по важности, стабильности и срочности.

Типичные проблемы:

- Неверное распределение приоритетов.
- Нестабильные требования, которые могут измениться.

10. Корректность и проверяемость (correctness and verifiability) Требование должно быть корректным и проверяемым.

Типичные проблемы:

- Опечатки, грамматические ошибки.
- Неаргументированные требования к дизайну.
- Требования к пользователю, а не к приложению.

### **Техники тестирования требований**

1. Взаимный просмотр (peer review)

- Беглый просмотр (walkthrough).
- Технический просмотр (technical review).
- Формальная инспекция (inspection).

2. Вопросы Задавание вопросов для уточнения требований.

3. Тест-кейсы и чек-листы Продумывание тест-кейсов для проверки требований.

4. Исследование поведения системы Моделирование работы пользователя с системой для выявления неоднозначностей.

5. Графическое представление (рисунки, схемы) Использование диаграмм для наглядного представления требований.

6. Прототипирование Создание прототипов интерфейсов для оценки решений.

### **Практическая часть**

Требования к разрабатываемому приложению:

1. Приложение должно выполнять математические вычисления.
2. Несколько приложений должны иметь возможность работать одновременно.

Анализ требований:

1. Завершённость:

- Требование 1 не уточняет, какие именно математические вычисления должны выполняться.
- Требование 2 не указывает, как приложения будут взаимодействовать или какие ресурсы будут использоваться.

2. Атомарность:

- Оба требования достаточно атомарны, но требуют уточнения.

3. Непротиворечивость:

- Противоречий между требованиями нет.

4. Недвусмысленность:

- Требование 1 слишком общее, неясно, какие именно вычисления должны выполняться.
- Требование 2 не уточняет, как именно приложения будут работать одновременно.

5. Выполнимость:

- Оба требования выполнимы, но требуют уточнения.

6. Обязательность и актуальность:

- Оба требования кажутся обязательными, но их актуальность зависит от контекста.

7. Прослеживаемость:

- Требования не пронумерованы и не структурированы.

8. Модифицируемость:

- Требования могут быть легко изменены, но это может привести к потере ясности.

9. Проранжированность:

- Приоритеты требований не указаны.

10. Корректность и проверяемость:

- Требования не проверяемы из-за их общности.

**Вопросы заказчику:**

1. Какие именно математические вычисления должно выполнять приложение?
2. Как приложения будут взаимодействовать при одновременной работе?
3. Какие ресурсы будут использоваться для поддержки одновременной работы приложений?
4. Каковы приоритеты данных требований?

### **Заключение**

Получили начальные знания в области тестирования документации и требований заказчика.