Лабораторная работа №2 Определение требований и создание сценариев использования.

Цель работы:

Научиться выявлять пользовательские требования к веб-ресурсу и на их основе формулировать варианты использования, пользовательские истории и сценарии.

# Общие сведения

При тестировании веб-ресурсов важно уметь выявлять и документировать пользовательские требования, которые определяют, как и для чего конечный пользователь взаимодействует с сайтом. Эти требования могут быть не задокументированы заранее, поэтому их анализ и формулировка становятся первыми шагами для разработки тест-кейсов и построения тестовой стратегии.

## Основные определения

**Пользовательские требования** — это описание потребностей и ожиданий пользователей от веб-ресурса. Они помогают лучше понять, какие функции и действия наиболее важны для пользователя.

**Пользовательская история** — краткое, емкое описание одной функции или действия, которое пользователь хочет выполнить на сайте, чтобы достичь своей цели. Обычно оформляется в формате: "Как [тип пользователя], я хочу [действие], чтобы [цель]".

**Вариант использования (Use Case)** — это структурированное описание одного из сценариев взаимодействия пользователя с веб-системой. Включает основные шаги, которые предпринимает пользователь, а также реакции системы. Дополнительно может содержать альтернативные сценарии, описывающие, что происходит в случае ошибок или отклонений.

**Пользовательский сценарий** — это пошаговое описание действий пользователя и откликов системы в рамках одного сценария использования. Это более детализированное описание взаимодействия, которое помогает учесть все нюансы и исключительные ситуации.

## Стоп-слова при написании требований

При написании требований следует избегать слов и фраз, которые могут привести к неоднозначности, непониманию или неправильной интерпретации. Вот некоторые категории слов, которых стоит избегать:

**1. Неопределенные и двусмысленные термины**

Часто используемые: "возможно", "примерно", "где-то", "иногда", "по возможности", "при необходимости"

Пример: "Система должна работать примерно так, чтобы пользователь был доволен."

Почему избегать: Эти слова не дают четкого представления о том, как должна работать система, оставляя место для различных интерпретаций.

**2. Субъективные и оценочные слова**

Часто используемые: "удобный", "легкий", "быстрый", "дружелюбный", "эффективный"

Пример: "Приложение должно быть удобным для пользователя."

Почему избегать: Эти термины субъективны и могут иметь разное значение для разных людей. Лучше использовать конкретные метрики или стандарты для описания.

**3. Неоднозначные количественные оценки**

Часто используемые: "много", "мало", "достаточно", "недостаточно", "максимально"

Пример: "Система должна обрабатывать много данных."

Почему избегать: Такие слова не предоставляют конкретных чисел или диапазонов, что делает невозможным объективную оценку выполнения требования.

**4. Термины, указывающие на будущее**

Часто используемые: "будет", "может быть сделано"

Пример: "В будущем система будет поддерживать несколько языков."

Почему избегать: Требования должны быть конкретными и касаться того, что система должна делать, а не того, что может быть реализовано в будущем.

**5. Излишне обобщенные слова**

Часто используемые: "и т.д.", "и прочее", "и тому подобное"

Пример: "Система должна поддерживать возможность экспорта данных в CSV, Excel и прочие форматы."

Почему избегать: Такие фразы не указывают конкретные параметры и могут вызвать неясности относительно объема и границ разработки.

**6. Слова с неясной степенью важности**

Часто используемые: "должен", "следует", "рекомендуется"

Пример: "Система должна предоставлять отчет по запросу."

Почему избегать: "Должен" и "следует" имеют разные уровни обязательности. Рекомендуется четко разделять обязательные и необязательные требования, используя такие термины, как "обязан" для обязательных функций и "может" для необязательных.

**7. Технический жаргон и термины, непонятные пользователям**

Часто используемые: "API-интерфейс", "SOAP-запросы", "латентность"

Пример: "Приложение должно поддерживать интеграцию через REST API."

Почему избегать: Если требования пишутся для конечных пользователей или менеджеров, избегайте сложных технических терминов, которые могут быть им непонятны. Требования должны быть написаны на понятном языке для всех участников.

**Примеры корректных формулировок:**

Вместо "Приложение должно быть быстрым" — "Приложение обязано загружать страницу в течение не более чем 3 секунд при 1000 активных пользователях."

Вместо "Система должна быть удобной" — "Система обязана требовать не более 3 кликов для выполнения действия 'Добавить товар в корзину'."

# Практическая часть

1. **Анализ веб-ресурса:**

Проведите анализ веб-сайта, который был вам назначен. Определите основные цели сайта и ключевые действия, которые может выполнять пользователь. Например, это может быть сайт интернет-магазина, где главные цели — поиск и покупка товаров.

1. **Пользовательские требования:**

На основе анализа составьте **не менее трех пользовательских требований** к сайту. Это должны быть функциональные ожидания пользователей, основанные на вашем понимании цели веб-ресурса.

**Пример:**

Пользователи обязаны иметь возможность искать товары по ключевым словам.

Пользователи хотят просматривать отзывы других покупателей перед покупкой.

1. **Пользовательские истории:**

Для каждого требования сформулируйте по **одной пользовательской истории**. Это будет краткое описание потребностей пользователей в формате "Как [кто], я хочу [что], чтобы [зачем]".

**Пример:**

Как пользователь, я хочу искать товары по ключевым словам, чтобы найти интересующий меня продукт.

Как покупатель, я хочу видеть отзывы других пользователей о продукте, чтобы принять осознанное решение о покупке.

1. **Варианты использования (Use Cases):**

Создайте **два варианта использования** для сайта. Описание должно включать основной сценарий (последовательность шагов, ведущих к достижению цели), а также альтернативные сценарии (что произойдет, если что-то пойдет не так).

**Пример варианта использования для поиска товаров:**

**Название:** Поиск товаров по ключевым словам на сайте.

**Акторы:** Покупатель (пользователь), Система (сайт).

**Описание:** Пользователь вводит ключевое слово в строку поиска на главной странице сайта для поиска товаров. Система анализирует введенный запрос, отображает результаты поиска или информирует пользователя о том, что по запросу ничего не найдено.

**Триггер:** Пользователь хочет найти товар и вводит запрос в строку поиска.

**Предусловие:** Пользователь открыл сайт и находится на главной странице. Строка поиска доступна для использования.

**Основной сценарий:**

1. Пользователь вводит запрос в строку поиска.
2. Система проверяет запрос на соответствие доступным товарам.
3. Система отображает список товаров, соответствующих запросу.
4. Пользователь выбирает нужный товар из списка.
5. Система перенаправляет пользователя на страницу выбранного товара.

**Альтернативный сценарий:**

Если система не находит товаров по запросу:

1. Система отображает сообщение "Товар не найден".
2. Система предлагает пользователю список похожих товаров (при наличии).
3. Пользователь может выбрать товар из предложенного списка или вернуться к повторному поиску.

**Результат:** Пользователь либо нашел товар, который его интересует, либо система предоставила ему информацию о том, что товар не найден, и предложила возможные альтернативы.

1. **Пользовательские сценарии:**

Для каждого из вариантов использования разработайте **пошаговый пользовательский сценарий**. Это детализированное описание каждого шага, который должен выполнить пользователь для достижения своей цели, с описанием реакции системы.

**Пример пользовательского сценария для поиска товаров:**

1. Пользователь открывает сайт и видит главную страницу.
2. Пользователь нажимает на строку поиска.
3. Пользователь вводит запрос "смартфон".
4. Система отображает список подходящих смартфонов.
5. Пользователь прокручивает список, выбирает интересующий товар и нажимает на его название.
6. Система перенаправляет пользователя на страницу товара с его описанием, ценой и возможностью добавить в корзину.

# Ожидаемый результат:

* Пользовательские требования (3 и более).
* Пользовательские истории (3 и более).
* Как минимум два варианта использования, включая основной и альтернативные сценарии.
* Как минимум два пользовательских сценария с детализированными пошаговыми инструкциями.

# Контрольные вопросы

1. Чем отличаются пользовательские истории от вариантов использования?
2. Как создание вариантов использования помогает улучшить процесс тестирования?
3. Почему важно учитывать пользовательские требования в тестировании веб-ресурсов?