**Управление качеством программных систем. Супрунов Матвей, ПИ21-3**

**ЛАБОРАТОРНАЯ РАБОТА №2 (8).**

Тема лабораторной работы: работа с классификацией видов тестирования.

**Документация к веб-приложению с курсами повышения квалификации**

**Цель работы:**

Целью данной работы является проведение анализа ранее разработанного проекта системы микробатчевой обработки данных для ресторана. В рамках анализа будут рассмотрены различные уровни, типы, методы и виды тестирования, применимые к данному проекту, а также приведены примеры характерных дефектов и сделаны выводы по результатам работы.

**Описание программного проекта:**

Проект представляет собой систему микробатчевой обработки данных в рамках микросервисной архитектуры с использованием следующих компонентов:

1. **Контейнеризация**: Docker для контейнеризации микросервисов.
2. **Оркестрация**: Kubernetes для управления контейнерами и развертывания микросервисов.
3. **Брокер сообщений**: Apache Kafka для передачи данных между сервисами.
4. **Key-Value хранилище**: Redis для хранения справочной информации.
5. **База данных**: PostgreSQL для хранения обработанных данных.
6. **ETL-процессы**: Реализация микробатчевой обработки данных.
7. **DWH**: Структура хранения данных с использованием слоев STG, DDS и CDM.

**По каждому из элементов классификации, описанному в разделе методических рекомендаций, привести список видов, типов, методов, уровней тестирования, применимых для выбранного проекта:**

**Уровни тестирования**

1. **Компонентное (модульное) тестирование**:
   * **Цель**: Тестирование отдельных модулей, функций или классов для выявления локализованных ошибок.
   * **Применимо**: Для микросервисов, написанных на Python, каждую функцию или класс необходимо тестировать отдельно.
   * **Примеры дефектов**: Ошибки реализации алгоритмов, неправильная обработка исключений.
2. **Интеграционное тестирование**:
   * **Цель**: Проверка взаимодействия между компонентами системы.
   * **Применимо**: Тестирование взаимодействия микросервисов через Kafka и их взаимодействие с Redis и PostgreSQL.
   * **Примеры дефектов**: Ошибки передачи данных между микросервисами, неправильная интеграция с Redis.
3. **Системное тестирование**:
   * **Цель**: Проверка функциональных и нефункциональных требований к системе в целом.
   * **Применимо**: Тестирование всей системы, включая контейнеризацию и оркестрацию.
   * **Примеры дефектов**: Неверное использование ресурсов, не предусмотренные сценарии использования.

**Типы тестирования**

1. **Функциональное тестирование**:
   * **Цель**: Проверка функциональности системы согласно требованиям.
   * **Применимо**: Тестирование микросервисов на соответствие функциональным требованиям, проверка правильности обработки заказов.
   * **Примеры дефектов**: Неправильная логика обработки заказов, ошибки в бизнес-логике.
2. **Нефункциональное тестирование**:
   * **Цель**: Проверка нефункциональных аспектов системы, таких как производительность и надежность.
   * **Применимо**: Нагрузочное тестирование Kafka, стресс-тестирование PostgreSQL, тестирование удобства использования системы.
   * **Примеры дефектов**: Низкая производительность при высокой нагрузке, неудобный интерфейс пользователя.

**Методы тестирования**

1. **Статические методы**:
   * **Цель**: Проверка программного кода и документации без выполнения кода.
   * **Применимо**: Рецензирование кода микросервисов, использование статических анализаторов (например, SonarQube).
   * **Примеры дефектов**: Упущения в требованиях, ошибки в коде, которые могут быть обнаружены без запуска программы.
2. **Динамические методы**:
   * **Цель**: Проверка системы при ее выполнении.
   * **Применимо**: Тестирование функциональности микросервисов, интеграционных процессов, нагрузочное тестирование.
   * **Примеры дефектов**: Ошибки при взаимодействии компонентов, сбои при высокой нагрузке.

**Виды тестирования**

1. **По объекту**:
   * **Применимо**: Модульное тестирование отдельных микросервисов, интеграционное тестирование взаимодействия через Kafka, системное тестирование всей системы.
2. **По субъекту**:
   * **Применимо**: Тестирование может проводиться как разработчиками (модульное тестирование), так и тестировщиками (интеграционное и системное тестирование).
3. **По степени автоматизации**:
   * **Применимо**: Использование автоматизированных тестов для модульного и интеграционного тестирования, использование CI/CD для автоматизации развертывания и тестирования.
4. **По признаку позитивности**:
   * **Применимо**: Позитивное тестирование для проверки правильности работы системы в нормальных условиях, негативное тестирование для проверки системы в условиях ошибок и исключений.
5. **По времени проведения**:
   * **Применимо**: Регрессионное тестирование после изменений в коде, тестирование на стадии разработки и перед выпуском в продуктивную среду.

**Привести примеры дефектов, характерных для каждого вида тестирования:**

1. **Компонентное тестирование**:
   * Ошибки в алгоритмах обработки данных.
   * Неправильная обработка исключений в функциях.
2. **Интеграционное тестирование**:
   * Ошибки при передаче данных между микросервисами.
   * Неправильное взаимодействие с Redis.
3. **Системное тестирование**:
   * Непредусмотренные сценарии использования, приводящие к сбоям.
   * Неверная обработка комбинаций данных на пользовательском уровне.

**Выводы по работе**

Проведенный анализ показывает, что для успешной реализации проекта необходимо учитывать все уровни, типы, методы и виды тестирования. Важно разработать и внедрить автоматизированные тесты для всех уровней тестирования, чтобы обеспечить высокое качество системы и ее стабильную работу. Необходимо также проводить регулярное рецензирование кода и использовать статические анализаторы для выявления потенциальных проблем на ранних этапах разработки.

**Список использованных источников**

1. Официальная документация Redis (Дата обращения: 22.05.2024)

https://master--redis-doc.netlify.app/docs/

1. Статья “ Лидерство в тестировании — документация” (Дата обращения: 22.05.2024)

https://habr.com/ru/articles/772242/

1. Статья “ Тестирование и анализ пользовательской документации: подходы, инструменты и советы ”(Дата обращения: 22.05.2024)

https://documenterra.ru/testirovanie-i-analyz-polzovatelskoy-documentacii-podhody-instrumenty-i-sovety/