|  |  |
| --- | --- |
| http://www.pl130.ru/doc/index/%D0%9B%D0%BE%D0%B3%D0%BE%D1%82%D0%B8%D0%BF22.jpg | Санкт-Петербургское государственное бюджетное  профессиональное образовательное учреждение  "Колледж электроники и приборостроения" |

|  |
| --- |
| Фащевская Надежда Владимировна |
| ФИО |

|  |
| --- |
| 41ИП |
| Группа |

Практическая работа

Тема: «Разработка технической документации с использованием современных инструментов»

Санкт-Петербург – 2024

**Цель:** 1. Закрепить знания о типах технической документации.

2. Научиться использовать инструменты для разработки и управления технической документацией.

3. Освоить основные принципы автоматизации процесса создания документации.

**Ход работы:**

1. Название проекта: Мобильное приложение для отслеживания расходов.

Структура документации

**Техническое задание1**

Функциональные требования2

Нефункциональные требования3

Функциональные возможности

**Руководство пользователя4**

**План** **тестирования5**

1. <https://disk.yandex.ru/i/kzMyI68E5U2OVQ>

**Контрольные вопросы**

1. Общие вопросы:

1.1. Что такое техническая документация, и какова её роль в разработке программного обеспечения?

Техническая документация – набор документов, используемых для автоматизации и систематизации процесса разработки и использования программного продукта

Роль т.д. – упрощение передачи знаний, обеспечение возможности поддержки и масштабирования системы, снижение рисков, связанных с уходом сотрудников или сменой команды

1.2. Какие виды технической документации вы знаете? Приведите примеры.

1. Документы для разработчиков: архитектурные решения, спецификации API, технические задания.

2. Документы для тестировщиков: тест-кейсы, отчёты об ошибках, тестовые планы.

3. Документы для пользователей: руководства, инструкции, справочные материалы.

1.3. Какие цели преследует разработка технической документации?

Основные цели:

- Упрощение передачи знаний.

- Обеспечение возможности поддержки и масштабирования системы.

- Снижение рисков, связанных с уходом сотрудников или сменой команды.

2. Инструменты разработки документации:

2.1. Назовите основные категории инструментов для разработки технической документации.

* Редакторы текста и Markdown редакторы
* Системы управления контентом (CMS)
* Инструменты для генерации документации
* Инструменты для совместной работы
* Платформы для хостинга и публикации
* Инструменты для внесения изменений и отслеживания версий
* Инструменты для визуализации

2.2. Чем отличаются MS Word и LaTeX при создании документации?

MS Word используется для простых и средних документов без типографики, LaTeX же используется для сложных документов с высоким уровнем типографики

2.3. Какие инструменты используются для автоматической генерации документации из исходного кода?

Doxygen и Swagger

2.4. Как Git может быть полезен в управлении технической документацией?

При помощи него можно хранить и управлять изменениями документов, отслеживать версии документов и разработки

3. Технологии разработки документов:

3.1. В чём заключается подход "Документация как код" (Docs as Code)?

При нём документация хранится в репозиториях и обрабатывается так же, как и программный код

3.2. Что такое семантическая разметка, и почему она важна в документации?

Подход к разметке, который опирается не на содержание сайта, а на смысловое предназначение каждого блока, она важна, так как помогает поддерживать читаемость и автоматическую обработку документации

3.3. Какие технологии используются для управления версиями документации?

Git и Pandoc

3.4. Объясните, что такое динамическая документация и где она применяется.

Динамическая документация – это динамическое обновление на основе изменений в коде или данных. Она используется при разработке больших проектов между раздельно расположенной командой, когда поддержание документации актуальной важная часть разработки

4. Практические аспекты:

4.1. Какие преимущества предоставляет использование шаблонов при создании документации?

Преимуществами можно считать автоматизацию процесса создания документации, так как оформление и структура больше не занимает столько же времени

4.2. Как CI/CD может помочь в поддержании актуальности технической документации?

CI/CD помогает уменьшить ручной труд, минимизировать ошибки и улучшить качество документации, обеспечивая ее актуальность и точность посредством интеграции с репозиториями, автоматизацией обновлений, проверкой на соответствие, упрощением совместной работы и т.д.

4.3. Почему важно учитывать целевую аудиторию при создании документации?

Чтобы документы были прочтены и учтены клиентом, их нужно оформлять в связи с языком, используемым целевой аудиторией, иначе могут возникать недопонимания и дополнительные исправления, занимающие время

5. Современные подходы и инструменты:

5.1. Какие инструменты используются для интерактивной документации?

Doxygen, JSDoc

5.2. Как Confluence может помочь при коллективной работе над документами?

* Совместное редактирование
* Комментарии и обсуждения
* Управление версиями
* Структурирование информации
* Шаблоны
* Интеграция с другими инструментами
* Настройка прав доступа
* Поиск и фильтрация

5.3. Для чего предназначены инструменты Swagger и OpenAPI?

Для тестирования и описания REST API

5.4. Приведите примеры облачных сервисов, применяемых для совместной работы над документацией.

Google Docs, Dropbox Paper и Microsoft OneDrive

6. Критическое осмысление:

6.1. Какие сложности могут возникнуть при автоматической генерации документации?

Сложность персонализации документации, отсутствия частей из-за их нечитаемости в коде, искажение информации и т.д.

6.2. Как можно минимизировать риски устаревания документации в крупном проекте?

Обновлять информацию раз в определённый промежуток времени

6.3. Какие ошибки часто встречаются при создании технической документации? Как их избежать?

Использование неправильных терминов при оформлении документов, специализированного языка без опоры на клиента. Упущение важных деталей продукта, таких как второстепенные свойства или функции

Избежать их можно проверкой документации повторно лицом не участвующим в создании документации, а также разделением документов на отделы и части при разработке для уделения достаточного внимания каждому из них

7. Творческое задание:

7.1. Представьте, что вам нужно разработать техническую документацию для нового API. Опишите шаги, инструменты и технологии, которые вы будете использовать.

7.2. Разработайте структуру документации для веб-приложения, предназначенного для управления задачами.

1. Введение

- 1.1. Обзор приложения

- 1.2. Цели и задачи

- 1.3. Основные функции

2. Установка и настройка

- 2.1. Требования к системе

- 2.2. Установка приложения

- 2.3. Конфигурационные файлы

3. Архитектура приложения

- 3.1. Общая архитектура

- 3.2. Используемые технологии

- 3.3. Структура базы данных

4. Пользовательский интерфейс

- 4.1. Основные элементы интерфейса

- 4.2. Навигация по приложению

- 4.3. Основные страницы

5. Функциональные возможности

- 5.1. Управление задачами

- 5.1.1. Создание новой задачи

- 5.1.2. Редактирование задачи

- 5.1.3. Удаление задачи

- 5.1.4. Состояния задач

- 5.2. Управление проектами

- 5.2.1. Создание проекта

- 5.2.2. Назначение задач на проект

- 5.3. Командная работа

- 5.3.1. Приглашение участников

- 5.3.2. Комментарии к задачам

- 5.4. Отчеты и аналитика

6. Пользовательская документация

- 6.1. Регистрация и вход в систему

- 6.2. Установка прав доступа

- 6.3. Часто задаваемые вопросы (FAQ)

7. Администраторская документация

- 7.1. Управление пользователями

- 7.2. Обслуживание системы

- 7.2.1. Резервное копирование данных

- 7.2.2. Обновление приложения

- 7.3. Решение проблем

8. Разработка и поддержка

- 8.1. Установка и использование среды разработки

- 8.2. Архитектурные решения

- 8.3. Конструкция кода

- 8.4. Направления для дальнейшей разработки

9. Лицензия

- 9.1. Условия использования

- 9.2. Авторы и юрисдикция

10. Полезные ссылки

- 10.1. Сообщество пользователей

- 10.2. Внешние библиотеки или API

- 10.3. Инструкции по интеграции с другими сервисами

Заключение

- 11.1. Обратная связь и контактная информация