

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Звіт
про виконання лабораторної роботи № 2
«ОЗНАЙОМЛЕННЯ З ПІДХОДОМ ДО РОЗРОБКИ WATERFAL»

Виконав:
студент групи ФеП-41
Линва Віталій
Перевірив:
доц. Франів В. А.

Львів 2024

Мета: ознайомитися та вивчити основні принципи методології розробки Waterfall, оволодіти ключовими концепціями. Забезпечити себе навичками використання цієї методології для ефективного управління проектами та розробки продуктів в умовах фіксованих вимог і чітко визначених етапів виконання робіт.

Хід роботи

Проект: Веб-сервіс менеджер підписок за методологією Waterfall.

Розробка проекту за методологією Waterfall

1. Оцінка та Планування (Requirements)

1.1. Вимоги: Збір і деталізація вимог до програмного забезпечення

- **Функціональні вимоги:**

- Реєстрація та авторизація користувачів.
- Управління списками підписок (додавання, редагування, видалення).
- Інтеграція з сторонніми постачальниками послуг.
- Пошук та фільтрація підписок.
- Система сповіщень про нові підписки та оновлення.
- Звіти про використання.

- **Нефункціональні вимоги:**

- Масштабованість: система повинна бути здатною витримувати навантаження великої кількості користувачів.
- Безпека: система повинна бути захищена від несанкціонованого доступу та зберігати дані користувачів у безпеці.
- Надійність: система повинна бути доступною 24/7 з мінімальним простоєм.
- Використання: система повинна бути простою у використанні та зрозумілою для користувачів.
- Продуктивність: система повинна швидко та ефективно обробляти запити користувачів.

1.2. Планування: Визначення обсягу робіт, визначення ресурсів та часових рамок

- Створення проектного плану: визначити завдання, необхідні для реалізації проекту, та їхню послідовність.
- Оцінка ресурсів: визначити необхідні для реалізації проекту людські та матеріально-технічні ресурси.

- Створення календарного плану: визначити часові рамки для виконання кожного завдання.

2. Проектування (Design)

2.1. Архітектура: Розробка загальної архітектури системи

- **Вибір технологій:** визначити технології, які будуть використовуватися для реалізації проекту.
- **Проектування модулів:** розробити модульну архітектуру системи, де кожен модуль відповідає за певну функціональність.
- **Проектування бази даних:** розробити структуру бази даних для зберігання даних користувачів та їх підписок.

2.2. Дизайн інтерфейсу: Визначення вигляду та взаємодії користувача з сервісом

- **Розробка прототипів:** створити прототипи інтерфейсу користувача для візуалізації та тестування взаємодії користувача з веб-сервісом.
- **Створення макетів:** розробити макети інтерфейсу користувача, які описують візуальне оформлення та розташування елементів на екрані.

3. Реалізація (Implementation)

3.1. Вибір технологій: Вибір інструментів та технологій для реалізації проекту

- **Вибір мови програмування:** визначити мову програмування, яка буде використана для розробки веб-сервісу.
- **Вибір фреймворку:** визначити фреймворк, який буде використаний для розробки веб-сервісу.
- **Вибір інструментів тестування:** визначити інструменти тестування, які будуть використані для перевірки якості веб-сервісу.

3.2. Програмування: Створення коду та реалізація функціональності

- **Розробка модулів:** розробити код для кожного модуля веб-сервісу відповідно до проектної документації.
- **Інтеграція модулів:** інтегрувати модулі веб-сервісу один з одним.
- **Тестування модулів:** провести тестування кожного модуля веб-сервісу, щоб переконатися, що всі вони

функціонують належним чином.

4. Тестування (Testing)

4.1. Проведення різновидів тестування для перевірки якості та стабільності програмного забезпечення

- **Системне тестування:** тестування веб-сервісу в цілому, щоб переконатися, що він відповідає всім вимогам.
- **Навантажувальне тестування:** тестування веб-сервісу під навантаженням, щоб переконатися, що він може витримувати очікуване навантаження користувачів.
- **Тестування безпеки:** тестування веб-сервісу на наявність вразливостей та загроз безпеці.
- **Приймальне тестування:** тестування веб-сервісу з залученням кінцевих користувачів, щоб отримати їхні відгуки та переконатися, що веб-сервіс відповідає їхнім потребам.

5. Впровадження (Deployment)

5.1. Реліз та впровадження програмного продукту в робоче середовище

- **Розгортання веб-сервісу в тестовому середовищі:** встановити та протестувати веб-сервіс у тестовому середовищі, щоб переконатися, що він готовий до розгортання в робочому середовищі.
- **Розгортання веб-сервісу в робочому середовищі:** встановити та налаштувати веб-сервіс у робочому середовищі.
- **Навчання користувачів:** надати користувачам інструкції та необхідні матеріали для використання веб-сервісу.
- **Підтримка та моніторинг:** надати користувачам підтримку та моніторинг роботи веб-сервісу.

Висновок: метод Waterfall схожий до приготування страви за рецептом. Він непогано підходить для створення веб-сервісу менеджера підписок я гадаю, якщо все необхідне чітко визначено і чітко вказано з самого початку.

Сильні сторони Waterfall:

Чітка структура та послідовність: кожен етап чітко визначений і виконується послідовно, це полегшує планування та дає змогу краще контролювати проект.

Передбачувані результати: чітко визначені етапи та документація дозволяють передбачати, яким буде результат проекту, що в свою чергу може допомогти зменшити ризики.

Слабкі сторони Waterfall:

Відсутність гнучкості: внесення змін до проекту на пізніх стадіях може бути складним і дорогим, що інколи може бути великою проблемою.

Тривалий час розробки: оскільки усі етапи послідовні, загальний час розробки може бути довшим ніж у гнучких методологіях.

Високі ризики: є ризик виявлення помилок, особливо трагічно це може проявитися на пізніх стадія, це буде вкрай дорогим у виправленні.