Методологии разработки программного обеспечения

Тема: Методологии разработки программного обеспечения.

Цель: исследовать методологии разработки программного обеспечения; провести анализ программного продукта для выбора подходящей методологии.

Задание

1. Классификация методологий разработки программного обеспечения.

По степени гибкости

1. Гибкие методологии (Agile):

* Scrum
* Kanban
* Extreme Programming (XP)

1. Жесткие методологии (Waterfall):

* Каскадная модель
* V-Model

По подходу к проектированию

1. Проектно-ориентированные:

* Waterfall
* PRINCE2

1. Инкрементные и итерационные:

* Agile
* RUP (Rational Unified Process)

По степени вовлечения заказчика

1. Клиенто-ориентированные:

* Agile
* DSDM (Dynamic Systems Development Method)

1. Проектно-ориентированные:

* Waterfall
* Spiral

По подходу к тестированию

1. Тестирование на основе требований:

* Waterfall
* V-Model

1. Тестирование, управляемое дизайном (DDT):

* Design Driven Testing

1. Автоматизированное тестирование:

* TDD (Test Driven Development)
* BDD (Behavior Driven Development)

1. Исследование методологии «Design Driver Testing».

Исследование методологии «Design Driver Testing» представлена в таблице 1.

Таблица 1 – Исследование методологии «Design Driver Testing»

|  |  |
| --- | --- |
| Характеристика | Описание |
| Полное название | Design Driven Testing |
| Авторы | Неизвестны, развитие шло в рамках практики разработчиков |
| Год появления | 2010-е годы |
| Основные принципы и критерии для выбора | Акцент на тестировании интерфейсов и пользовательского опыта |
| К какой стратегии разработки ПО относится | Agile, Lean |
| Состав команды | Дизайнеры, тестировщики, разработчики |
| Формулировка требований | Основана на пользовательских сценариях |
| Специализированное ПО (при наличии) | Инструменты для прототипирования (Figma, Adobe XD), инструменты тестирования |
| Примеры успешных проектов | Приложение для прогноза погоды, приложения с акцентом на UX |
| Достоинства методологии | Упрощение взаимодействия, высокое качество пользовательского интерфейса |
| Предполагаемые недостатки методологии | Высокая зависимость от качества прототипов и дизайн-документации |

Влияние методологии на проект.

1. Положительное влияние на создание проекта

* Обеспечивает улучшение пользовательского опыта благодаря тщательному тестированию интерфейсов.
* Позволяет выявить проблемы на ранних стадиях за счёт тесной интеграции дизайнеров и тестировщиков.

1. Риски для проекта

* Возможны задержки из-за необходимости создания подробных прототипов.
* Высокие требования к квалификации команды.

1. Предполагаемые роли в проекте

* Дизайнеры: Разработка прототипов и дизайн-макетов.
* Тестировщики: Создание тестов на основе дизайн-документации.
* Разработчики: Реализация функций согласно тестам и макетам.

Дополнительное задание

Проект «Приложение для просмотра и изменения расписания».

Преимущества и недостатки различных методологий представлены в таблице 2.

Таблица 2 – Сравнение различных методологий

|  |  |  |
| --- | --- | --- |
| Методология | Преимущества | Недостатки |
| Design Driven Testing | Высокое качество UX, раннее выявление ошибок | Зависимость от прототипов, задержки |
| Scrum | Гибкость, быстрая адаптация | Возможна нехватка документации |
| Waterfall | Чёткая структура, подробная документация | Не подходит для гибких проектов |
| TDD | Высокая надёжность кода, автоматизация тестирования | Требует высокой квалификации команды |

Наиболее подходящая методология для проекта «Приложение для прогноза погоды» является «Design Driven Testing».

1. Положительное влияние на создание проекта

* Упрощение взаимодействия пользователя с приложением.
* Быстрое выявление ошибок в пользовательских сценариях.

1. Риски для проекта

* Возможные задержки из-за длительного создания прототипов.
* Высокие требования к квалификации дизайнеров и тестировщиков.

1. Предполагаемые роли в проекте

* Дизайнеры: Создание UX/UI макетов.
* Тестировщики: Написание и выполнение тестов.
* Разработчики: Реализация функций в соответствии с макетами и тестами.