Практическая работа 4

**Выработка требований к программному обеспечению и программному модулю**

Задание 1:

**Жизненный цикл ПО** – процесс, описывающий все этапы создания и поддержки программного продукта, от идеи до вывода из эксплуатации.

**Программный продукт** – сложный и многогранный объект, который включает в себя не только код, но и множество других элементов, необходимых для его нормальной работы и поддержки.

**Критерии качества ПП** – это набор характеристик и атрибутов, которые используются для оценки того, насколько хорошо программный продукт соответствует потребностям пользователей и заданным требованиям.

**Критерии ПП:**

* Функциональность
* Надежность
* Удобство использования
* Эффективность
* Тестируемость

**Виды ПО:**

* Системное
* Прикладное
* Инструментальное

**Стадии разработки программного продукта(ПП):**

1. Планирование, анализ требований

*Определение целей и задач*

1. Проектирование

*Определение общей структуры программного продукта*

1. Реализация

*Написание кода, интеграция модулей, управление версиями*

1. Тестирование

*Приемочное тестирование*

1. Развертывание

*Подготовка инфраструктуры для развертывания ПП*

1. Эксплуатация

*Исправление ошибок, адаптация ПП*

Задание 2:

Разработка требований – это процесс выявления анализа, документирования и согласования потребностей и ожиданий заинтересованных сторон относительно функциональности , производительности, характеристик и ограничений будущего ПП.

Виды работ:

* Выявление требований
* Анализ требований
* Документирований требований

Задание 3:

**Пользовательские требования** – описание того, что пользователи смогут делать с программным продуктом.

**Системные требования –** это более детальное и техническое описание функциональности характеристик и ограничений программного продукта

**Проектная системная спецификация –** это документ, который детально описывает архитектуру и внутреннюю структуру программного продукта

Задание 4:

**Виды требований к ПП:**

* Функциональные требования
* Пользовательские требования
* Бизнес-требования
* Системные требования
* Нефункциональные требования

Задание 5:

**Функциональные требования –** требования, которые описывают что система должна делать

**Нефункциональные требования** – требования, которые описывают как система должна работать, то есть её характеристики и ограничения, не связанные с конкретными функциями.

Задание 6:

**Сравнение моделей разработки**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | Модель разработки | Особенности | **+** | **\_** |
| 1 | Водопадная | Линейная последовательность этапов. Каждый этап должен быть завершен до начала следующего | Простота реализации, подходит для проектов с четкими требованиями | Сложно вносить изменения после завершения этапа |
| 2 | Каскадная | Похожа на водопадную, но допускает возврат на предыдущий этап при обнаружении проблем | Позволяет вносить изменения на ранних этапах | Все еще сложно вносить изменения |
| 3 | Спиральная | Итеративная модель, в которой проект развивается по спирали, с каждой итерацией добавляется новая функциональность | Высокая гибкость, возможность управления рисками на каждой интеграции | Сложность в управлении, требует высокой квалификации |
| 4 | Итеративная | Проект разбивается на небольшие итерации, в каждой из которых рабочий прототип | Быстрый выход первой версии | Требует чёткого планирования и управления итерациями |

Вывод: Лучшей моделью для меня является каскадная. Она хороша для небольших проектов и позволяет возвращаться на предыдущие этапы.

Задание 7:

SADT-модели – это методология графического моделирования для анализа и проектирования систем.

Примеры: Моделирования бизнес-процессов, разработка информационных систем

Задание 8:

Case-средства – это программные инструменты, которые автоматизируют различные этапы разработки программного обеспечения, от анализа требований до генерации кода и тестирования.

Примеры: Rational Rose, Microsoft Visio.

Задание 9 :

Объективно-ориентированный анализ и проектирование – это подход к анализу и проектированию программного обеспечения, основанный на концепциях объектов, классов, наследования и инкапсуляции.

Примеры: UML-редакторы, Visual Studio.

Задание 10:

Языки UML – это стандартизованный графический язык моделирования, используемый для визуализации, спецификации, конструирования и документирования артефактов ПО.

Примеры: Диаграммы прецедентов, диаграммы последовательностей.

Задание 11:

Группа разработчиков – это команда людей, совместно разрабатывающих программное обеспечение.