ПРАКТИЧЕСКАЯ РАБОТА 4

**1. ЖЦ ПП, критерии качества ПП, виды ПО, стадии разработки ПП**

Жизненный цикл программного продукта (ЖЦ ПП) - это период времени от начала разработки до окончания его использования, включающий этапы: анализ требований, проектирование, разработка, тестирование, внедрение, сопровождение и вывод из эксплуатации.

Критерии качества ПП:

* Функциональная соответствие
* Надежность
* Удобство использования
* Эффективность
* Сопровождаемость
* Переносимость

Виды ПО:

* Системное ПО
* Прикладное ПО
* Инструментальное ПО

Стадии разработки ПП:

* Формирование требований
* Проектирование
* Реализация
* Тестирование
* Внедрение
* Сопровождение

**2. Разработка требований (определение, виды работ)**

Разработка требований - процесс выявления, документирования и проверки требований к программному обеспечению.

Виды работ:

* Сбор требований
* Анализ требований
* Спецификация требований
* Верификация требований
* Управление изменениями требований

**3. Пользовательские требования, системные требования, проектная системная спецификация**

Пользовательские требования - требования, описывающие, что система должна делать с точки зрения конечного пользователя.

Системные требования - детализированные технические требования, определяющие функции и ограничения системы.

Проектная системная спецификация - документ, содержащий детальное описание архитектуры, компонентов и интерфейсов системы.

**4. Виды требований к ПП**

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

**5. Функциональные требования, нефункциональные требования**

Функциональные требования - определяют, что система должна делать, какие функции выполнять.

Нефункциональные требования - определяют, как система должна работать (производительность, безопасность, надежность и т.д.).

**6. Таблица «Сравнение моделей разработки»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Модель разработки** | **Особенности** | **+** | **-** |
| **1** | Водопадная (Cascade) | Последовательное, поэтапное выполнение фаз. Переход к следующему этапу только после полного завершения предыдущего. | Простота понимания, строгий контроль, понятная документация. | Низкая гибкость, позднее тестирование, сложность внесения изменений. |
| **2** | Итерационная (Interative) | Поэтапная разработка, где система создается постепенно, по частям (итерациям). | Быстрое получение прототипа, легче вносить изменения, снижение рисков. | Сложность управления, может потребовать больше ресурсов в целом. |
| **3** | Спиральная (Spiral) | Циклическая модель, фокусирующаяся на оценке и минимизации рисков на каждой итерации. | Высокая ориентированность на риски, подходит для крупных и рискованных проектов. | Сложность в управлении и оценке стоимости, требует высокой квалификации. |
| **4** | Гибкая (Agile) | Итеративная и инкрементальная разработка, тесное взаимодействие с заказчиком, быстрая реакция на изменения. | Высокая гибкость, удовлетворенность заказчика, ранняя и частая поставка ПО. | Требует высокой дисциплины команды, размытые сроки итоговой поставки. |

**7. SADT-модели (определение, примеры)**

SADT (Structured Analysis and Design Technique) - методология структурного анализа и проектирования систем. Пример: функциональные диаграммы, диаграммы потоков данных.

**8. CASE-средства (определение, примеры)**

CASE-средства - инструменты для автоматизации процессов разработки ПО. Примеры: IBM Rational Rose, Enterprise Architect, Visual Paradigm.

**9. Объектно-ориентированный анализ и проектирование (OOAD)**

OOAD - методология анализа и проектирования систем на основе объектов. Инструменты: UML, паттерны проектирования, средства рефакторинга.

**10. UML, языки UML**

UML (Unified Modeling Language) - унифицированный язык моделирования. Примеры диаграмм: Use Case, Class, Sequence, Activity, State Machine.

**11. Группа разработчиков**

Типовая структура группы разработчиков:

* Руководитель проекта
* Аналитики
* Архитектор
* Разработчики (backend, frontend)
* Тестировщики
* Технические писатели

Вывод:

Вывод.