# Учебный материал по программной инженерии

## 1. Введение в программную инженерию

Программная инженерия — это область знаний, занимающаяся разработкой, проектированием, тестированием и сопровождением программного обеспечения с использованием систематических подходов. Основная цель — обеспечение создания качественного, надёжного и эффективного программного обеспечения (ПО) при оптимальных затратах.

## 2. Определение и основные задачи программной инженерии

\*\*Определение:\*\* Программная инженерия — это дисциплина, связанная с применением инженерных подходов к разработке программного обеспечения.

\*\*Основные задачи:\*\*
- Определение требований к ПО.
- Разработка архитектуры и проектирование системы.
- Написание и отладка программного кода.
- Тестирование и проверка соответствия требованиям.
- Сопровождение и обновление ПО.

## 3. Различия между программной инженерией и программированием

1. \*\*Программирование\*\* — это процесс написания программного кода на определённом языке программирования.
2. \*\*Программная инженерия\*\* включает программирование, но также охватывает управление проектами, тестирование, проектирование архитектуры и другие аспекты.

| \*\*Аспект\*\* | \*\*Программирование\*\* | \*\*Программная инженерия\*\* |
|-------------------------|---------------------------------------|------------------------------------|
| Объём задач | Написание кода | Весь жизненный цикл ПО |
| Основная цель | Создание работающей программы | Создание качественного ПО |
| Уровень сложности | Часто локальные задачи | Комплексные проекты |

## 4. Жизненный цикл разработки программного обеспечения

Жизненный цикл ПО — это процесс создания программного обеспечения, включающий несколько этапов:
1. \*\*Сбор и анализ требований.\*\*
2. \*\*Проектирование.\*\* Разработка архитектуры системы.
3. \*\*Разработка.\*\* Написание программного кода.
4. \*\*Тестирование.\*\* Проверка качества ПО.
5. \*\*Внедрение и сопровождение.\*\* Обновление и устранение ошибок.

## 5. Модели жизненного цикла разработки ПО

Для организации процесса разработки используются различные модели жизненного цикла:

- \*\*Водопадная модель (Waterfall)\*\*
- \*\*Спиральная модель (Spiral)\*\*
- \*\*Гибкая методология (Agile)\*\*
- \*\*DevOps и CI/CD\*\*

## 6. Водопадная модель (Waterfall)

\*\*Суть:\*\* Линейная последовательность этапов, где каждый этап начинается после завершения предыдущего.

\*\*Преимущества:\*\*
- Простота управления.
- Чёткая структура.

\*\*Недостатки:\*\*
- Трудно вносить изменения.
- Высокие риски в крупных проектах.

## 7. Спиральная модель (Spiral)

\*\*Суть:\*\* Разработка происходит в циклической последовательности с фокусом на анализ рисков.

\*\*Преимущества:\*\*
- Подходит для крупных проектов.
- Гибкость в требованиях.

\*\*Недостатки:\*\*
- Высокие затраты.
- Требуется опытная команда.

## 8. Гибкая методология (Agile)

\*\*Суть:\*\* Разработка ведётся небольшими итерациями, что позволяет оперативно вносить изменения.

\*\*Преимущества:\*\*
- Быстрая адаптация к изменению требований.
- Частая доставка рабочих версий ПО.

\*\*Недостатки:\*\*
- Требуется тесное взаимодействие команды.
- Менеджмент сложнее, чем в традиционных моделях.

## 9. DevOps и CI/CD

\*\*DevOps:\*\* Методология, объединяющая процессы разработки (Dev) и эксплуатации (Ops).

\*\*CI/CD:\*\*
- \*\*Continuous Integration (CI):\*\* Постоянное объединение кода и автоматическое тестирование.
- \*\*Continuous Deployment (CD):\*\* Автоматизированное развёртывание ПО.

\*\*Преимущества:\*\*
- Ускорение выпуска обновлений.
- Повышение надёжности ПО.

## 10. Сбор и анализ требований

\*\*Цель:\*\* Определение функциональных и нефункциональных характеристик будущей системы.

## 11. Виды требований: функциональные и нефункциональные

1. \*\*Функциональные требования\*\* описывают, что должна делать система (например, регистрация пользователей).
2. \*\*Нефункциональные требования\*\* описывают качество работы системы (например, скорость, безопасность).

## 12. Методы сбора требований

- Интервью с заказчиками.
- Анкетирование.
- Анализ документации.
- Проведение семинаров и встреч.

## 13. Проблемы и ошибки на этапе анализа требований

- Неполнота требований.
- Противоречивость требований.
- Неправильное понимание заказчиком задач системы.

## 14. Проектирование программного обеспечения

\*\*Цель:\*\* Разработка архитектуры, структуры и логики ПО.

## 15. Понятие архитектуры ПО

Архитектура программного обеспечения — это высокоуровневое представление системы, описывающее её основные компоненты, связи между ними и принципы взаимодействия.

\*\*Примеры архитектурных стилей:\*\*
- Клиент-сервер.
- Микросервисы.
- Слоёная архитектура.