**Жизненный цикл ПО**

**Тестирование –** не изолированный процесс. Это часть модели жизненного цикла программного обеспечения (Software Development Life Cycle, SDLC). Именно поэтому выбор средств и методик тестирования будет напрямую зависеть от выбранной модели разработки. В этом разделе мы рассмотрим наиболее часто применяемые подходы к разработке программного обеспечения, а также популярные сегодня методологии и практики, такие как Agile и Scrum.

**Жизненный цикл программного обеспечения (также называемый циклом разработки) –** это условная схема, включающая отдельные этапы, которые представляют стадии процесса создания ПО. При этом, на каждом этапе выполняются разные действия.

Цикл разработки предлагает шаблон, использование которого облегчает проектирование, создание и выпуск качественного программного обеспечения. Это методология, определяющая процессы и средства, необходимые для успешного завершения проекта.

Хотя реализация принципов построения модели жизненного цикла для разных компаний может существенно отличаться, существуют стандарты, такие как ISO/IEC 12207, определяющие принятые практики разработки и сопровождения программного обеспечения.

Цель использования модели жизненного цикла – создать эффективный, экономически выгодный и качественный программный продукт.

**Анализ требований**

Цель этой стадии – определение детальных требований к системе. Кроме этого, необходимо убедиться в том, что все участники правильно поняли поставленные задачи и то, как именно каждое требование будет реализовано на практике.

Этот этап предполагает сбор требований к разрабатываемому программному обеспечению, их систематизацию, документирование, анализ, а также выявление и разрешение противоречий.

**Проектирование**

На стадии проектирования (называемой также стадией дизайна и архитектуры) программисты и системные архитекторы, руководствуясь требованиями, разрабатывают высокоуровневый дизайн системы.

Дизайн, как правило, закрепляется отдельным документом – дизайн-спецификацией (Design Specification Document, DSD).

На этом этапе, для упрощения визуализации процесса проектирования, используются так называемые нотации– схематическое выражение характеристик системы.

Основные используемые нотации:

* Блок-схемы.
* ER-диаграммы.
* UML-диаграммы.
* Макеты – например, нарисованный в фотошопе прототип сайта.

**Разработка и программирование**

На этом этапе начинается написание программистами кода программы в соответствии с ранее определенными требованиями.

Программирование предполагает четыре основных стадии:

* Разработка алгоритмов – создание логики работы программы.
* Написание исходного кода.
* Компиляция – преобразование в машинный код.
* Тестирование и отладка – юнит-тестирование.

**Документация**

Существует четыре уровня документации:

* Архитектурная (проектная) – например, дизайн-спецификация. Это документы, описывающие модели, методологии, инструменты и средства разработки, выбранные для данного проекта.
* Техническая – вся сопровождающая разработку документация (различные документы, поясняющие работу системы на уровне отдельных модулей).
* Пользовательская – включает справочные и поясняющие материалы, необходимые конечному пользователю для работы с системой. Это, к примеру, Readme и User guide, раздел справки по программе.
* Маркетинговая – включает рекламные материалы, сопровождающие выпуск продукта.

**Тестирование**

Процесс тестирования состоит из таких этапов:

* Планирование и управление - планирование тестирования включает действия, направленные на определение основных целей тестирования и задач, выполнение которых необходимо для достижения этих целей; составление тест-стратегии, тест-плана.
* Анализ и проектирование - это процесс написания тестовых сценариев и условий на основе общих целей тестирования.
* Внедрение и реализация - написание тест-кейсов, на основе написанных ранее тестовых сценариев, собирается необходимая для проведения тестов информация, подготавливается тестовое окружение и запускаются тесты.
* Оценка критериев выхода и написание отчетов - необходимо проверить было ли проведено достаточное количество тестов, достигнута ли нужная степень обеспечения качества системы.

Действия по завершению тестирования - собираем, систематизируем и анализируем информацию о его результатах.

**Основные цели этого этапа**

* убедиться, что вся запланированная функциональность действительно была реализована;
* проверить, что все отчеты об ошибках, поданные ранее, были, так или иначе, закрыты;
* завершение работы тестового обеспечения, тестового окружения и инфраструктуры;
* оценить общие результаты тестирования и проанализировать опыт, полученный в его процессе.

**Внедрение и сопровождение**

Когда программа протестирована и в ней больше не осталось серьезных дефектов, приходит время релиза и передачи ее конечным пользователям.

В случае обнаружения пользователями тех или иных пост-релизных багов, информация о них передается в виде отчетов об ошибках команде разработки, которая, в зависимости от серьезности проблемы, либо немедленно выпускает исправление (hot-fix), либо откладывает его до следующей версии программы.