Лабораторная работа №4

Процессы жизненного цикла программных средств

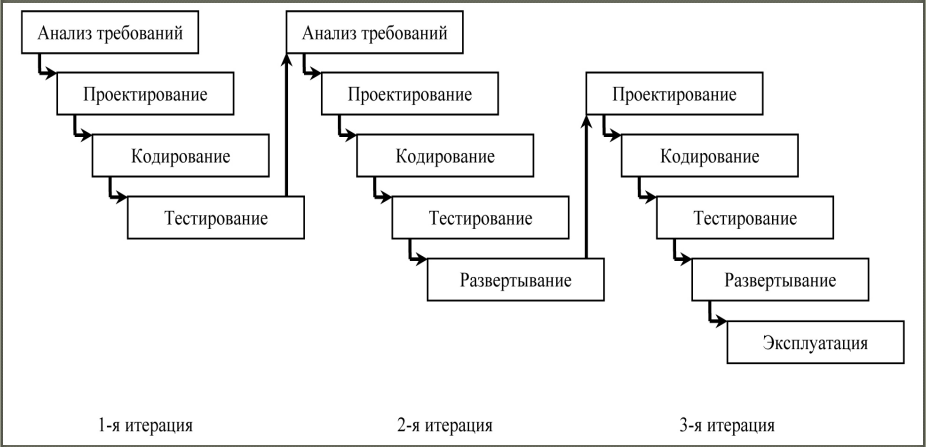
1. Обосновать модель ЖЦ, наиболее подходящую для вашего проекта.

Для данного проекта наиболее подходящей моделью жизненного цикла будет **итеративная модель.**

Эта модель позволяет постепенно развивать и улучшать систему, начиная с базовой версии и постепенно добавляя новые функции. В контексте нашего проекта, начальная версия может включать простую отправку поздравлений, а затем, по мере тестирования, можно добавлять более сложные функции, такие как персонализация поздравлений, интеграция с API ВКонтакте и улучшение пользовательского интерфейса.

Преимущества итеративного подхода:

1. **Гибкость и адаптация**: позволяет адаптировать проект под изменяющиеся требования, что важно для социальной сети, где предпочтения пользователей могут меняться.
2. **Легкость в тестировании**: на каждом этапе можно выявить проблемы и устранить их, минимизируя потенциальные сбои в проекте.
3. **Быстрое получение результата**: первая версия системы будет работать достаточно быстро, что позволит получить обратную связь от пользователей и улучшать функциональность на основе реальных данных.



Почему не другие?

**Каскадная модель.** В данном проекте требования могут меняться, и тестирование продукта может выявить необходимость в доработках, которые данная модель не позволяет сделать из-за жесткой последовательности этапов разработки.

**Спиральная модель.** Проект не слишком сложен. Данная модель будет избыточной, так как требует дополнительных усилий на анализ рисков и проектирование.

**С промежуточным контролем.** Модель требует частого контроля и не эффективна при меняющихся требованиях.

**V-образная модель.** Требует жесткую последовательность этапов и неэффективна при изменениях на пройденных этапах.

**Модель фазы.** Требует излишне много документации.

2. Выделить этапы выполнения проекта в соответствии с ЖЦ.

**Итерация 1:**

**Цель:** разработать базовую архитектуру системы и основные функции.

**Задачи:**

* 1. Сбор начальных требований и создание концептуальной модели.
  2. Разработка первой версии функционала: генерация простых поздравлений.
  3. Интеграция с API ВКонтакте для отправки сообщений.
  4. Первичное тестирование базового функционала.

**Результат:** Рабочий прототип, который генерирует и отправляет базовые поздравления.

3. Провести кодирование и комплексную отладку ПО.

В процессе

### ****Итерация 2:****

**Цель**: расширить функциональность и улучшить пользовательский интерфейс.

**Задачи**:

* 1. Добавление персонализации поздравлений (например, с учетом интересов или данных получателей).
  2. Разработка улучшенного интерфейса для ввода данных и управления поздравлениями.
  3. Интеграция новых функций (например, возможность выбора разных типов поздравлений).
  4. Проведение тестирования новых функций и исправление ошибок.
  5. Сбор обратной связи от пользователей.

**Результат**: Расширенная версия программы.

### ****Итерация 3:****

**Цель**: завершить разработку, оптимизировать систему и подготовить к запуску.

**Задачи**:

* 1. Оптимизация производительности системы, улучшение алгоритмов генерации поздравлений.
  2. Завершение тестирования всех функций и исправление выявленных ошибок.
  3. Подготовка документации
  4. Нагрузочное тестирование и обеспечение стабильности работы системы.
  5. Подготовка к релизу.

**Результат**: Готовая к запуску версия системы с полным функционалом и стабильной работой.

Вопросы

Какие виды жизненного цикла программного обеспечения вы знаете?

1. Каскадная модель
2. Итеративная модель
3. Модель с промежуточным контролем
4. V-образная модель
5. Модель фазы

Понятие процесса. Виды процессов в соответствии с принятыми стандартами

Процесс разработки ПО — это совокупность взаимосвязанных действий, направленных на создание, разработку, тестирование, внедрение и сопровождение программных продуктов. Процесс включает в себя планирование, проектирование, реализацию, тестирование, документирование и другие этапы, которые организуются в определённой последовательности для достижения качественного результата.

### Стандарты, регулирующие процессы в разработке ПО:

1. **ISO/IEC 12207** – Международный стандарт, который определяет процесс жизненного цикла программного обеспечения, включая все стадии: планирование, анализ, проектирование, разработка, тестирование, внедрение, эксплуатация и обслуживание.
2. **CMMI (Capability Maturity Model Integration)** – Модель зрелости процессов, которая описывает процессы для разработки и совершенствования процессов в организациях. CMMI имеет несколько уровней зрелости, от начального (неформализованного) до оптимизированного.
3. **SPICE (Software Process Improvement and Capability dEtermination)** – Стандарт для оценки и улучшения процессов разработки ПО, ориентирован на зрелость процессов и их повышение.
4. **ISO/IEC 15504** – Модель, основанная на SPICE, для оценки и улучшения процесса разработки ПО и управления качеством.

ГОСТ Р ИСО/МЭК 12207-2010) устанавливает 11 процессов:

a) определение требований правообладателей;

b) анализ системных требований;

c) проектирование архитектуры системы;

d) процесс реализации;

e) процесс комплексирования системы;

f) процесс квалификационного тестирования системы;

g) процесс инсталляции программных средств;

h) процесс поддержки приемки программных средств;

i) процесс функционирования программных средств;

j) процесс сопровождения программных средств;

k) процесс изъятия из обращения программных средств

ГОСТ 19.102-77 основные этапы разработки ПО:

• постановка задачи (стадия «Техническое задание»);

• анализ требований и разработка спецификаций (стадия «Эскизный проект»);

• проектирование (стадия «Технический проект»);

• реализация (стадия «Рабочий проект»).

Основные отличия между **ГОСТ Р ИСО/МЭК 12207-2010** и **ГОСТ 19.102-77** заключаются в следующем:

* **ГОСТ Р ИСО/МЭК 12207-2010** фокусируется на **процессах жизненного цикла ПО**, их стандартизации и обеспечении качества на всех этапах разработки.
* **ГОСТ 19.102-77** ориентирован на **документирование процесса проектирования ПО**, включая создание технической документации.