

Лабораторная работа №4

• **Выбрать и обосновать модель ЖЦ ПО**

Для нашего программного обеспечения больше всего будет подходить каскадная модель жизненного цикла. В ходе анализа мы выявили следующие причины:

- **Итеративное уточнение требований:** По мере продвижения проекта, требования к ПО могут меняться и уточняться. Эта модель позволяет гибко реагировать на новые требования и адаптироваться под изменяющиеся условия.
- **Разбиение на небольшие части:** Проект можно разбить на небольшие, управляемые части, что значительно упрощает планирование и реализацию. Каждая часть может быть разработана и протестирована отдельно, снижая риски.
- **Ранний запуск проекта:** Благодаря созданию прототипов на ранних этапах, можно получить обратную связь от пользователей и внести необходимые изменения, прежде чем приступить к полной реализации. Это сокращает время до выхода на рынок и увеличивает шансы на успех.
- **Возможность многоступенчатой оценки качества:** На каждой стадии разработки можно проводить анализ продвижения проектирования продукта, что позволяет вовремя выявлять и устранять возможные проблемы. Это обеспечивает более стабильное и предсказуемое развитие проекта.
- **Многочисленные версии:** Проект строится в виде нескольких версий, каждая из которых проверяется и оценивается до окончательной реализации. Это позволяет постепенно наращивать функциональность и улучшать качество ПО.

Этапы выполнения проекта в соответствии с ЖЦ

Этап 1: Планирование и анализ требований

- **Сбор и анализ требований:** Изучение существующих решений и определение их сильных и слабых сторон, чтобы учесть их в новом проекте.
- **Создание пользовательских сценариев:** Разработка сценариев использования системы, описывающих, как различные пользователи будут взаимодействовать с приложением..
- **Формирование списка необходимых технологий и инструментов:** Выбор инструментов для управления проектом, контроля версий, автоматизации тестирования и других аспектов разработки и определение подходящих технологий для реализации проекта, включая языки программирования, фреймворки, базы данных и инфраструктурные решения.

Результат: Документ с техническим заданием, описывающий основные требования и функциональные возможности.

Этап 2: Проектирование архитектуры системы

- **Создание архитектуры системы:** Идентификация и описание ключевых компонентов системы, таких как серверная часть, клиентская часть, база данных и интерфейсы между ними.

- **Детальное проектирование модулей:** Определение структур данных и их взаимодействий в системе и описание функциональности каждого модуля, включая входные данные, выходные данные и основные операции.
- **Определение интерфейсов:** Описание интерфейсов прикладного программирования (API), которые будут использоваться для взаимодействия между клиентской и серверной частями системы.
- **Обеспечение безопасности системы:** Разработка методов защиты данных пользователей и контроля доступа и описание методов шифрования данных при передаче и хранении.

Результат: Документ с архитектурным и детальным проектированием системы, включая схемы, диаграммы и спецификации, готовые для реализации на этапе разработки.

Этап 3: Кодирование и разработка MVP (минимально жизнеспособного продукта):

- **Программирование модулей:** Создание пользовательского интерфейса (UI) с учетом удобства использования и функциональности и последующее проектирование и реализация структуры базы данных для эффективного хранения информации о прокате видеокассет и бронировании киносеансов.
- **Интеграция модулей:** Обеспечение корректного взаимодействия между пользовательским интерфейсом и сервером для обработки данных в реальном времени.
- **Тестирование на уровне модулей:** Проведение тестов для каждого отдельного модуля, чтобы проверить их функциональность и выявить возможные ошибки на ранних стадиях разработки.
- **Оптимизация кода:** Повышение быстродействия и снижения потребления ресурсов, а так же разработка решений, позволяющих системе работать эффективно при увеличении нагрузки и числа пользователей.

Результат: Рабочая версия приложения с интегрированными модулями, прошедшая начальное тестирование и готовая к системному тестированию..

Этап 4: Тестирование и отладка

- **Системное тестирование:** Проверка всех функциональных возможностей системы на соответствие требованиям, описанным в техническом задании и оценка производительности системы под различными условиями нагрузки для определения её пределов и устойчивости. Отдельная проверка защиты данных пользователей и устойчивости к внешним угрозам.
- **Тестирование пользовательского интерфейса:** Оценка удобства и интуитивности интерфейса для конечных пользователей и проверка соответствия интерфейса дизайнерским стандартам и ожиданиям пользователей..

Результат: Устранение выявленных ошибок, создание стабильной версии приложения.

Этап 5: Интеграция и комплексное тестирование

- **Запуск последней (финальной) версии:** Установка и конфигурация всех необходимых серверов, баз данных и прочих элементов инфраструктуры для обеспечения стабильной работы системы.
- **Регрессионное тестирование:** Повторное тестирование после внесения изменений, чтобы убедиться в отсутствии новых ошибок.
- **Подготовка релиза:** Сборка окончательной версии системы, готовой к внедрению.

Результат: Исправленная и стабильная версия приложения, прошедшая всестороннее тестирование и готовая к внедрению.

Этап 6: Внедрение и обучение пользователей

- **Подготовка инфраструктуры:** Установка и конфигурация всех необходимых серверов, баз данных и прочих элементов инфраструктуры для обеспечения стабильной работы системы.
- **Внедрение системы:** Установка и настройка приложения в рабочей среде и проверка корректности установки и готовности системы к использованию.
- **Поддержка в переходный период:** Обеспечение поддержки пользователей в процессе перехода на новую систему и решение возникающих вопросов.

Результат: Готовое приложение, готовое к использованию конечными пользователями.

Этап 7: Поддержка и доработка

- **Сбор обратной связи:** Мониторинг отзывов пользователей для выявления возможных улучшений.
- **Мониторинг системы:** Внедрение системы мониторинга для отслеживания работоспособности приложения в режиме реального времени, выявления и устранения проблем.
- **Техническая поддержка:** Обеспечение службы технической поддержки для пользователей, предоставление консультаций и помощь в решении возникающих проблем.

Результат: Внедренная система и обученные пользователи, готовые эффективно работать с новым программным обеспечением.