МИНИСТЕРСТВО НАУКИ и высшего образования

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

Институт среднего профессионального образования

**Отчёт по практической работе**

**по учебной дисциплине «МДК.04.01 Обеспечение качества функционирования КС»**

**Тема: «Выбор модели жизненного цикла»**

Выполнила студентка

Код специальности 09.02.07

Информационные системы и программирование

22919/21

Колодина О. Я.

Иванова Д.В.

Санкт-Петербург,

2025

**Тема:**

Выбрана тема – Книжный онлайн магазин.

**Предметная область:**  
Должно быть 4 уровня доступа:   
1) пользователь (неавторизированный);  
2) пользователь (авторизированный);  
3) админ.

Сайт, на котором есть каталог книг доступных для заказа (наличие на складе), корзина, обратная связь, отзывы, отслеживание заказа.

*Объём нагрузки:*  
Одновременно может выдерживать нагрузку в 10 тыс. пользователей.  
Количество личных кабинетов: до 100 тыс. (с возможностью обновления и развития).

*Конфиденциальность:*  
Защита личного кабинета пользователя, защита данных пользователя, создание защищенного соединения с банком.

*Характеристика заказчика:*  
Заказчику нужен сайт для покупки книг онлайн и их доставки. Сроки выполнения проекта – 6 месяцев.  
Заказчик требовательный, контактный, точное представление интерфейса проекта, его функционала.  
Заказчик имеет высшее гуманитарное управленческое образование.

*Аналог:*

Литрес.

**Роли для группы разработки:**

1. Frontend-разработчик;
2. Архитектор программного обеспечения;
3. Технический писатель;
4. Бизнес-аналитик;
5. Менеджер проекта;
6. Инженер по тестированию;
7. Менеджер по тестированию;
8. Заказчик;
9. Системный администратор;
10. Инженер по требованиям;
11. UX/UI-дизайнер;
12. Руководитель команды разработки;
13. Специалист по кибербезопасности;
14. Backend-разработчик;
15. Лингвист-локализатор.

**Условия проекта:**

*Количество и качество ролей:*   
15 высококвалифицированных сотрудников, имеют опыт совместной работы. Имеют высшее специальное образование.

*Функционал:*   
Средней сложности разработки. Нет четких требований к ПО. Нужна работа без сбоев.

*Местоположение:*   
Доставка организованна из России по таким странам, как: Казахстан, Беларусь.

*Особенности:*   
Доставка по другим странам с единой дополнительной наценкой.

Сильные стороны: (для разработки)

Есть чёткие требования к интерфейсу, есть связь с заказчиком, в команде высококвалифицированные работники, работавшие ранее вместе.

Слабые стороны:

Нет чётких требований к ПО, мало времени на разработку, заказчик не имеет технического образования.

**Модель ЖЦ**

Подобраны три МЖЦ, наиболее подходящие для данного проекта:  
1) каскадная   
2) прототипирование  
3) scrum

КАСКАДНАЯ

*Преимущества:*   
Разработка идет строго по установленному плану (требования -> разработка -> тест -> заказчик), можно вернуться к любому виду деятельности и изменить ошибку, если таковая есть.

*Недостатки:*  
У заказчика есть чёткие требования, а в этой модели ЖЦ план не всегда соответствует реальному проекту, изначально есть чёткие требования к интерфейсу, а в этой модели ЖЦ их нет, они появляются только в будущем.  
  
ПРОТОТИПИРОВАНИЕ

*Преимущества:*  
Есть прототип проекта, заказчику будет удобнее рассмотреть сначала его, просмотреть свои требования.

*Недостатки:*  
Должны быть чёткие требования к ПО, однако в данном проекте их нет.   
  
SCRUM

*Преимущества:*  
Встречи раз в 24ч (или спринты 2-4 недели), что позволяет команде успешнее организовывать совместную работу. Получение быстрых результатов, что важно, т.к. наш срок сдачи проекта достаточно небольшой.

*Недостатки:*  
Имеются ежедневные сборы, не всем сотрудникам это может подойти. Требуется время на адаптацию к планированию.  
  
Выбрана модель ЖЦ – SCRUM.  
  
Эта модель ЖЦ позволяет держать эффективный совместный рабочий темп (благодаря совместным совещаниям и созвонам), так же за счет того, что у нас есть возможность общаться с заказчиком, он может так же присутствовать на таких собраниях, если его что то не устроит или он захочет что то обсудить.  
Эта модель ЖЦ позволяет получать быстрые результаты, для того чтобы уложиться в выделенные сроки.

**Виды деятельности** по ГОСТ Р ИСО/МЭК 12207-2010 «Процессы жизненного цикла программных средств»:

*Frontend-разработчик:*

- разработка пользовательского интерфейса (ui): с. 4.4.5 – производство программного обеспечения

- реализация взаимодействия с пользователем (ux): с. 6.2.1 – разработка и реализация

- оптимизация производительности клиентской части веб-приложения: с. 6.5.4 – тестирование

- тестирование интерфейса на различных устройствах и браузерах: с. 6.5.3 – тестирование

*Архитектор программного обеспечения:*

- проектирование архитектуры системы: с. 4.4.2 – определение архитектуры

- выбор технологий и инструментов для разработки: с. 5.1.2 – определение и оценка решений

- анализ и оценка рисков архитектуры: с. 5.3.4 – управление рисками

- создание документации по архитектуре системы: с. 6.1.3 – документация

*Технический писатель:*

- подготовка документации по пользовательскому интерфейсу и api: с. 6.1.1 – документация для пользователей

- создание справочной и эксплуатационной документации: с. 6.3.1 – поддержка и документация

- обеспечение актуальности документации по мере изменений в проекте: с. 6.1.4 – обновления документации

*Бизнес-аналитик:*

- сбор и анализ требований заказчика: с. 5.3.3 – определение требований

- описание и документирование бизнес-процессов: с. 6.2.4 – процесс разработки

- проведение анализа возможностей и ограничений системы: с. 5.3.5 – оценка возможностей

- подготовка спецификаций требований: с. 6.1.2 – спецификация требований

*Менеджер проекта:*

- планирование и распределение ресурсов: с. 5.1.1 – управление проектом

- управление сроками и бюджетом проекта: с. 4.4.5 – оценка и контроль

- контроль выполнения задач и качество результата: с. 5.1.4 – оценка производительности

- взаимодействие с заинтересованными сторонами: с. 5.3.2 – общение и информирование

*Инженер по тестированию:*

- разработка и выполнение тест-кейсов: с. 6.5.1 – тестирование

- проведение функционального и регрессионного тестирования: с. 6.5.2 – утверждение тестирования

- документирование найденных дефектов: с. 6.5.3 – регистрация дефектов

- оценка успешности тестирования: с. 6.5.5 – анализ результатов тестирования

*Менеджер по тестированию:*

- управление командой тестировщиков: с. 5.1.3 – управление ресурсами

- определение стратегий тестирования: с. 6.5.1 – стратегия тестирования

- анализ и отчетность по результатам тестирования: с. 6.5.4 – отчет о тестировании

- обеспечение качества тестовой документации: с. 6.1.3 – поддержка документации

*Заказчик:*

- формулирование целей и задач проекта: с. 5.3.1 – определение потребностей

- участие в определении приоритетов и функциональных требований: с. 5.3.3 – определение требований

- участие в выборе решения и средств разработки: с. 5.1.2 – оценка и выбор решений

- оценка итогового результата и соответствия требованиям: с. 6.5.4 – оценка качества продукта

*Системный администратор:*

- настройка и поддержка серверной инфраструктуры: с. 4.7.3 – обеспечение системы

- управление системной безопасностью: с. 5.3.6 – безопасность

- мониторинг работы приложений и серверов: с. 6.4.1 – поддержка и обслуживание

- резервное копирование и восстановление данных: с. 6.4.2 – управление данными

*Инженер по требованиям:*

- анализ и обоснование требований к системе: с. 5.3.3 – определение требований

- моделирование и валидация требований: с. 5.3.5 – валидация возможностей

- подготовка технических спецификаций на основе требований: с. 6.1.2 – спецификация требований

- прослеживаемость требований на всех этапах разработки: с. 6.1.5 – отслеживание требований

*UX/UI-дизайнер:*

- разработка графического оформления интерфейса (ui-дизайна): с. 6.2.4 – концепция проектирования – создание прототипов и макетов пользовательского интерфейса: с. 6.2.3 – дизайн интерфейсов

- проведение пользовательского тестирования на этапе дизайна: с. 6.5.1 – участие пользователей в тестировании

- взаимодействие с фронтэнд разработчиками для реализации дизайна: с. 4.4.5 – проектирование и реализация

*Руководитель команды разработки:*

- координация работы команды разработчиков: с. 5.1.3 – управление ресурсами

- проведение ежедневных стендапов и планерок: с. 5.1.4 – оценка и управление

- наставничество и развитие членов команды: с. 5.3.2 – обучение и развитие

- решение внутренних конфликтов и проблем: с. 5.3.1 – управление взаимодействиями

*Специалист по кибербезопасности:*

- проведение аудитов безопасности программного обеспечения: с. 6.5.6 – безопасность программного обеспечения

- определение угроз и уязвимостей системы: с. 5.3.6 – управление рисками безопасности

- разработка политики безопасности и защиты данных: с. 5.3.7 – политика безопасности

- обучение команды методам обеспечения безопасности: с. 6.3.2 – обучение по безопасности

*Backend-разработчик:*

- разработка серверной логики и api: с. 6.2.4 – процесс разработки

- интеграция с базами данных и внешними сервисами: с. 6.4.1 – подключение и интеграция

- оптимизация работы серверной части приложения: с. 6.5.3 – оптимизация производительности

- участие в код-ревью и тестировании серверной части: с. 6.5.1 – совместная работа над кодом

*Лингвист-локализатор:*

- перевод интерфейса и пользовательской документации: с. 6.3.1 – создание документации

- учет культурных особенностей в процессе локализации: с. 6.3.4 – локализация

- проверка и тестирование локализованных материалов: с. 6.5.4 – проверка качества

- сбор обратной связи от пользователей по локализованному контенту: с. 6.4.4 – оценка удовлетворенности пользователей

Данное распределение видов деятельности поможет четко определить обязанности каждого члена команды на всех этапах жизненного цикла программных средств, обеспечивая успешное выполнение проекта.

**Объединение видов деятельности в этапы и построение МЖЦ (Scrum):**

Для проекта «Книжный онлайн-магазин» выбран гибкий подход Scrum, который разбивает разработку на итерации (спринты) с регулярной обратной связью от заказчика. Виды деятельности, описанные в документе, группируются в следующие этапы:

*1. Подготовительный этап (Pre-Sprint)*

Цель: Определение требований, планирование.

Виды деятельности:

Бизнес-аналитик: Сбор и документирование требований.

Инженер по требованиям: Анализ и валидация требований.

Архитектор ПО: Проектирование архитектуры системы.

Менеджер проекта: Создание дорожной карты, распределение ролей.

Заказчик: Утверждение целей и приоритетов.

*2. Этап спринтов (Sprint 1..N)*

2.1. Планирование спринта (Sprint Planning)

Команда разработки (Frontend, Backend, DevOps) + Менеджер проекта выбирают задачи.

Тестировщики готовят тест-кейсы.

Технический писатель начинает работу над документацией.

2.2. Разработка (Development)

Frontend/Backend-разработчики: Реализация функционала (каталог, корзина, авторизация).

Дизайнеры: Уточнение UI/UX.

Специалист по безопасности: Проверка уязвимостей.

2.3. Тестирование (QA)

Инженеры по тестированию: Проводят тесты.

Менеджер по тестированию: Анализирует результаты, формирует отчёт.

2.4. Демонстрация (Sprint Review)

Презентация готового функционала заказчику.

Фиксация изменений в требованиях.

2.5. Ретроспектива (Retrospective)

Анализ ошибок и улучшение процессов.

*3. Финальный этап (Release & Deployment)*

Цель: Подготовка к релизу и масштабирование.

Виды деятельности:

Системный администратор: Настройка серверов.

Лингвист-локализатор: Перевод интерфейса.

Специалист по кибербезопасности: Финальный аудит.

Технический писатель: Подготовка руководства пользователя.