# **Coursing**

веб-сайт для онлайн обучения

проект подготовлен

студентаками группы 22919/22

Лапшиной Виктории

и Дарьей Плотниковой

Проект направлен на разработку платформы онлайн-обучения, которая предоставляет курсы, викторины и форумы сообщества для пользователей для улучшения своих навыков в различных предметах.

**предметная область:**

Платформа для онлайн-обучения будет обслуживать студентов, профессионалов и учащихся на протяжении всей жизни. Будут добавлены роли пользователей и уровони доступа (отдельный и для создателей, и для потребителей контента), должности администраторов и модераторов. Также будет создана система безопасности для защиты конфиденциальности данных пользователей (платежная информация и системы доступа к курсам). Платформа ориентирована на одновременную активность 5000 пользователей и наличия 25000 аккаунтов на момент выпуска первой версии (рост количества аккаунтов до 100000).

**роли в команде разработчиков:**

1. Заказчик: заинтересованное лицо с неограниченным бюджетом
2. Руководитель проекта: Контролирует проект, координирует между командами.
3. Бизнес-аналитик: Собирает требования, определяет объем проекта.
4. UI/UX Designer: Разрабатывает пользовательские интерфейс
5. Frontend Developer: Реализует визуальные аспекты платформы.
6. Backend developer: Управляет логикой на стороне сервера и интеграцией баз данных.
7. Тест-менеджер: создает модели тестирования
8. Тестер: проводит тестирование
9. “Белый” хакер: проводит проверку безопасности, путем “взлома” программы.
10. DevOps engineer: обрабатывает развертывание и управление инфраструктурой, настройка серверов.
11. Data Scientist: использует данные, чтобы решить конкретную задачу бизнеса. Проводит глубокий анализ данных, строит модели, на которых затем проверяет различные гипотез.
12. Machine Learning engineer: создает программы, которые обучаются на заданном наборе информации, а затем могут самостоятельно анализировать данные и делать на их основе выводы и прогнозы
13. Fullstack-разработчик: универсальный разработчик

**условия проекта:**

*Роли:* команда разработчиков из тринадцати человек, команда технической поддержки из пятнадцати человек, медиа отдел из пяти работников. Заказчик нацеленный на мировой рынок, достаточно активно участвующий в разработке проекта.

*Сложность ПО:* умеренная сложность с разработкой пользовательских уровней доступа, сложности аналитики рынка, а вследствие создания ТЗ, а также возможная проблема с обработкой платежей из-за рубежа

*Сильные стороны:* высокий рыночный спрос на онлайн-образование, опыт у команды разработки, упор на медиа продвижение.

*Слабости:* Конкуренция со стороны уже популярных платформ, например: Coursera, SkillBox; необходимость надежных маркетинговых стратегий.

**выбор моделей жизненного цикла**

1. Agile манифест.
2. Waterfall модель: процесс разработки выглядит как поток, последовательно проходящий фазы анализа требований, проектирования, реализации, тестирования, интеграции и поддержки.
3. Спиральная модель: сочетает в себе итеративное развитие с оценкой рисков, подходит для проектов с меняющимися требованиями.

**анализ выбранной модели**

Спиральная модель

* : быстрые изменения по запросу клиента и пользователей,

ранее создание рабочих прототипов,

улучшенный анализ рисков

— : сложность в управлении,

потенциально высокие затраты из-за нескольких итераций,

требование специализированного анализа рынка.

Agile манифест

* : адаптивность к потребностям клиентов,

отсутствие необходимости строгой документации

— : сложность с дедлайнами из-за частых изменений,

отсутствие четкого плана.

7. Выбор видов деятельности из ГОСТ-Р ISO/IEC 12207-2010

**1. Первичные процессы жизненного цикла**

— участвуют в разработке и обслуживании программного обеспечения.

(6.2.4.3.3) Приобретение и обеспечение навыков: определение, планирование и управление приобретением программных систем.

(6.1.2.3.5) Поставка и поддержка продукта (услуги): управление и выполнение задач, необходимых для поставки программных продуктов и услуг.

(7.2.1.3.2) Проектирование и разработка: деятельность, связанная с разработкой программного обеспечения, кодированием и тестированием, в том числе:

1. Анализ требований
2. Проектирование системной архитектуры
3. Разработка программного обеспечения
4. Кодирование и тестирование
5. Интеграция и тестирование системы

(7.2.6.3.3) Технические ревизии: проводиться для оценки программных продуктов или услуг с позиции рассмотрения и представления свидетельств того, что они соответствуют принятым стандартам и спецификациям, их разработка, эксплуатация или сопровождение проводится в соответствии с планами.

**2. Вспомогательные Процессы**

— предоставляют помощь и ресурсы для основных видов деятельности.

(7.2.1.3.1) Реализация процесса: необходимо разработать, документально оформить, выполнить план, определить документы, которые производятся в течение жизненного цикла программного продукта.

(7.2.2.3.3) Управление конфигурацией: управление и контроль конфигураций программного обеспечения на протяжении всего жизненного цикла.

(6.2.5.3.1) Менеджмент качества: обеспечение качества процессов и продуктов посредством аудита, обзоров и соблюдения стандартов.

(6.4.1.3.2) Идентификация требований: обеспечение того, чтобы конечный программный продукт соответствовал своему предполагаемому использованию и потребностям пользователя.

(6.4.8.3.1) Поддержка приемки программных средств: разработчик должен поддерживать ревизии и тестирование программного продукта, поддерживать обновления и новые возможности проекта.

(6.4.9.3.1) Подготовка к функционированию: разработать план и определить эксплуатационные стандарты для выполнения действий и задач этого процесса. План должен быть документирован и реализован.

(7.2.7.3.2) Аудит программных средств: независимая оценка процессов и продуктов для обеспечения соответствия стандартам.

**3. Организационные Процессы Жизненного Цикла**

— являются всеобъемлющими и обеспечивают организационную инфраструктуру для поддержки первичного и вспомогательного процессов.

(6.3.2.3.2) Управление проектом: обеспечение того, чтобы каждое действие жизненного цикла соответствовало его указанным требованиям, а также выявление, отслеживание и решение проблем, связанных с программным обеспечением.

(6.2.2.3.2) Создание инфраструктуры: создание и обслуживание инструментов, методов и сред, необходимых для разработки программного обеспечения.

(6.2.1.3.3) Совершенствование процессов: мероприятия, направленные на улучшение программных процессов.

(6.2.4.3.4) Менеджмент знаний: обеспечение необходимого обучения и развития навыков для персонала, участвующего в деятельности по жизненному циклу программного обеспечения.

**этапы модели жизненного цикла:**

Agile и спиральная модели

1. Исследование и анализ будущего проекта

* определить объем проекта
* определить заинтересованных сторон

2. Планирование

* анализ требований
* планирование спринта

3. Дизайн

* дизайн UI/UX
* архитектурное проектирование

4. Разработка и развитие

* создание ПО
* итеративное развитие функций
* непрерывная интеграция

5. Тестирование

* тестирование во время спринта
* тестирование на принятие конечным пользователем

6. Релиз

* выпуск ПО для пользователя
* предварительная оценка затрат и анализ взаимозависимостей

7. Техническая поддержка

* Постоянная поддержка
* Внедрение обратной связи для будущих итераций

Этот структурированный подход обеспечивает гибкость, обеспечивая при этом всестороннее охват основных этапов развития. В конце проходит оценка результата для перехода к новому витку.