УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Дисциплина «Экономика программной инженерии»

# Лабораторная работа 3

## Студенты

### Новиков Е.С.

Преподаватель *Гаврилов А. В.*

Санкт-Петербург, 2024 г.

**Задание:**

Предложить план действий в ситуации, когда прошло 3/4 срока, запланированного на реализацию проекта, а фактически выполнена только половина задач:

1. Определить, какие функции на данный момент еще не завершены и оценить, реализацию каких из них можно отложить для того, чтобы не сдвигать срок выпуска устраивающего заказчика работоспособного продукта с максимально сохраненной функциональностью.
2. Оценить возможность увеличения команды разработчиков для соблюдения сроков проекта, либо попытаться оптимизировать план работ

**Выполнение:**

В предложенных условиях, когда выполнена только половина задач за 3/4 срока проекта, можно рассмотреть следующий подход по разделению функций:

1. **Задача выполнена**: те задачи, которые завершены, и не требуют дальнейшего вмешательства.
2. **Время выполнения сокращено**: ключевые функциональности, которые еще не завершены, но которые являются важными для работоспособного продукта. Здесь стоит рассмотреть возможности сокращения объема работ по этим задачам, упрощения реализации, чтобы ускорить процесс.
3. **Отказ от задачи**: задачи не являющиеся критичными для основной функциональности продукта. Эти задачи можно отложить на будущее или совсем исключить, чтобы не увеличивать сроки разработки.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | Функционал | Optimistic (h-h) | Pessimistic (h-h) | Optimal (h-h) |
| 1.1 | Выбор технологий | 10 | 20 | 15 |
| 1.2 | Написание ТЗ | 20 | 40 | 30 |
| 1.3 | Дизайн и прототипирование | 60 | 110 | 85 |
| 2.1 | Каталог с фильтрацией | 70 | 100 | 85 |
| 2.2 | Корзина и оформление заказа | 40 | 80 | 60 |
| 2.3 | Интеграция платежной системы | 40 | 80 | 60 |
| 2.4 | Авторизация | 20 | 40 | 30 |
| 2.5 | Личный кабинет | 50 | 80 | 65 |
| 2.6 | Модуль записи на консультацию | 40 | 70 | 55 |
| 2.7 | Модуль записи на примерку дома | 40 | 70 | 55 |
| 2.8 | API для взаимодействия с фронтендом | 40 | 70 | 55 |
| 2.9 | База данных (структура, миграции) | 50 | 90 | 70 |
| 3.1 | Дизайн интерфейса (UI/UX) | 80 | 100 | 90 |
| 3.2 | Адаптивная верстка | 100 | 150 | 125 |
| 3.3 | Каталог и фильтрация | 80 | 120 | 100 |
| 3.4 | Корзина покупок | 50 | 80 | 65 |
| 3.5 | Личный кабинет | 50 | 80 | 65 |
| 3.6 | Поиск по сайту | 30 | 50 | 40 |
| 3.7 | Модуль записи на консультацию | 40 | 60 | 50 |
| 3.8 | Модуль записи на примерку дома | 40 | 60 | 50 |
| 3.8 | Страницы с юрид. документами (оферта, политика конфиденциальности) | 10 | 20 | 15 |
| 4.1 | Модульное тестирование | 40 | 80 | 60 |
| 4.2 | Интеграционное тестирование | 20 | 40 | 30 |
| 4.3 | Функциональное тестирование | 40 | 80 | 60 |
| 5.1 | Альфа и бета тестирование | 50 | 70 | 60 |
| 5.2 | Настройка сервера и деплой | 30 | 50 | 40 |
| 5.3 | Поддержка и исправление багов | 30 | 70 | 50 |

**Распределение задач по категориям**

**Желтые задачи: Важные, но не критичные для первого релиза**

1. **Личный кабинет пользователя (Backend и Frontend)** (2.5 и 3.5)  
   Личный кабинет важен для долгосрочного использования приложения, однако на начальном этапе можно ограничиться только базовым функционалом:
   * Отображение профиля пользователя (имя, email).
   * Возможность изменить минимальные настройки, например пароль.
   * Упрощение сложных функций, таких как история покупок или интеграция с внешними сервисами.

**Причина выбора:** Базовая версия личного кабинета позволит пользователям работать с системой, а полное развитие функционала можно перенести на последующие релизы.

1. **Интеграционное тестирование** (4.2)  
   Это тестирование важно для выявления ошибок взаимодействия модулей, однако на данном этапе достаточно сосредоточиться на критически важных сценариях, таких как:
   * Проверка взаимодействия API с фронтендом.
   * Основные сценарии работы корзины и оформления заказа.

**Причина выбора:** Полное покрытие тестами может быть отложено, а фокус на ключевых модулях ускорит выпуск продукта, сохраняя стабильность.

**Красные задачи: Отказ, не критичные для запуска продукта в срок**

1. **Подарочные сертификаты и промокоды (Backend и Frontend)** (2.8 и 3.8)  
   Данная функция добавляет удобство и гибкость пользователям, однако она не обязательна для минимально жизнеспособного продукта (MVP).

**Причина отказа:** Отсутствие этой функции не мешает основной работе приложения. Ее можно внедрить в следующем релизе.

1. **Модуль многоязычной поддержки** (3.9)  
   Поддержка нескольких языков важна для международной аудитории, но для первого релиза достаточно одной основной локализации.

**Причина отказа:** Этот модуль не критичен для базового функционала и не влияет на основные бизнес-процессы.

1. **Поиск по сайту** (3.10)  
   Эта функция делает работу с каталогом удобнее, но даже без нее пользователи смогут находить товары через категорийное деление.

**Причина отказа:** Функционал можно добавить позже, когда база данных будет расширена и оптимизирована.

1. **Раздел FAQ** (3.12)  
   Страницы с часто задаваемыми вопросами полезны для пользователей, но их можно представить в виде статичного документа или ссылки.

**Причина отказа:** Это не влияет на основной функционал и может быть добавлено в последующих версиях.

1. **Функциональное тестирование** (4.3)  
   Полное функциональное тестирование занимает значительное время, однако на этапе MVP достаточно протестировать ключевые модули.

**Причина отказа:** Полное покрытие тестами можно перенести на более поздний этап, сохранив фокус на основных сценариях.

**Зеленые задачи: Критичные, обязательные для первого релиза**

1. **Каталог с фильтрацией** (2.1)  
   Каталог обеспечивает доступ к товарам, что является ключевой функцией приложения. Однако фильтры можно упростить, например:
   * Только фильтрация по категориям или цене.
   * Исключение сложных условий (например, несколько критериев одновременно).

**Причина критичности:** Это основной функционал для поиска товаров, без которого приложение теряет смысл.

1. **Корзина и оформление заказа** (2.2)  
   Ключевая функция для онлайн-магазина. Необходимо реализовать:
   * Добавление и удаление товаров из корзины.
   * Базовые шаги оформления заказа.

**Причина критичности:** Без этого пользователи не смогут завершить процесс покупки.

1. **Интеграция платежной системы** (2.3)  
   Необходимо реализовать хотя бы один метод оплаты, например онлайн-оплату картой.

**Причина критичности:** Оплата является неотъемлемой частью процесса покупки.

1. **Авторизация** (2.4)  
   Реализация базовой регистрации и входа в систему. Дополнительные функции, такие как восстановление пароля, можно отложить.

**Причина критичности:** Авторизация позволяет пользователям взаимодействовать с системой.

1. **API для взаимодействия с фронтендом** (2.8)  
   Должен быть реализован минимальный набор методов для взаимодействия с ключевыми функциями:
   * Работа с каталогом и корзиной.
   * Авторизация.

**Причина критичности:** API обеспечивает взаимодействие между фронтендом и бекендом.

1. **База данных (структура, миграции)** (2.9)  
   Необходимо создать минимально жизнеспособную структуру базы данных, чтобы поддерживать ключевые функции.

**Причина критичности:** База данных — основа приложения.

1. **Альфа и бета тестирование** (5.1)  
   Этот этап необходим для проверки работоспособности ключевых функций перед выпуском.

**Причина критичности:** Тестирование позволяет выявить критические ошибки перед релизом.

1. **Настройка сервера и деплой** (5.2)  
   Базовая настройка и развертывание приложения на сервере.

**Причина критичности:** Без этого продукт не сможет быть запущен.

**Рассмотрим два подхода для соблюдения сроков проекта: увеличение команды и оптимизация плана работ.**

**Подход 1: Увеличение команды разработчиков**

**Преимущества:**

1. **Ускорение выполнения задач:** Дополнительные разработчики смогут взять на себя часть задач, что позволит завершить больше функций из желтой группы и реализовать их в полном объеме.
2. **Повышение качества:** Привлечение дополнительных ресурсов позволяет сохранить высокий уровень качества реализации, избегая упрощения функционала.

**Недостатки:**

1. **Дополнительные затраты:** Увеличение команды потребует дополнительных финансовых вложений, что может превысить запланированный бюджет проекта.
2. **Время на адаптацию:** Новым разработчикам потребуется время, чтобы вникнуть в проект, что на начальном этапе может даже замедлить работу команды.
3. **Сложность координации:** Управление увеличенной командой усложняется, особенно на поздних стадиях разработки, где синхронизация работы важна для соблюдения сроков.

**Подход 2: Оптимизация плана работ**

**Возможности для оптимизации:**

1. **Сокращение объема задач:** Реализация упрощенных версий функций, позволяя сосредоточиться на критичных элементах.
2. **Перераспределение ресурсов:** Акцент на завершение задач, необходимых для MVP, исключение менее приоритетных задач.
3. **Гибкость управления:** Возможность быстрее адаптироваться к изменениям требований и сроков.

**Преимущества:**

1. **Экономия ресурсов:** Не требует увеличения команды, что снижает финансовые затраты и минимизирует управленческие сложности.
2. **Фокус на ключевых функциях:** Оптимизация позволяет сконцентрироваться на основных модулях и задачах.
3. **Скорость адаптации:** Текущая команда, уже знакомая с проектом, быстрее и эффективнее завершит работу над MVP.

**Недостатки:**

1. **Риск неполного функционала:** Если заказчик ожидает большего, чем MVP, это может вызвать недовольство.
2. **Потенциальное снижение качества:** Упрощение задач и тестирования может привести к дефектам, выявленным после релиза.

**Выбор подхода**

Оптимизация плана работ представляется лучшим решением в данном случае, так как проект выполнен только на 50% при завершении 3/4 срока. Увеличение команды на таком этапе может создать больше проблем, чем принести пользы, в то время как оптимизация плана позволяет:

* Сохранить текущий бюджет.
* Уложиться в сроки.
* Выпустить MVP, который удовлетворит заказчика.

Таким образом, **оптимизация плана работ** более предпочтительна для успешного завершения проекта в срок. Этот подход позволит сэкономить ресурсы, избежать управленческих сложностей и выполнить ключевые функции в пределах оставшегося времени.