



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ ИУК «Информатика и управление»**

**КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,**

**информационные технологии»**

## **Домашняя работа №2**

**«Модели: DevOps. Agile. MSF»**

**ДИСЦИПЛИНА: «Методология программной инженерии»**

Выполнил: студент гр. ИУК4-11М \_\_\_\_\_ ( \_\_\_\_\_ Сафронов Н.С. )  
(подпись) (Ф.И.О.)

Проверил: \_\_\_\_\_ ( \_\_\_\_\_ Белов Ю.С. )  
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024

**Цель работы:** формирование и закрепление навыков по применению моделей разработки, используя модели DevOps, Agile, MSF.

### **Постановка задачи**

Разработка книжного интернет-магазина. Модель: Kanban.

Реализация домашнего задания возможна как с помощью инструмента Miro, так и с помощью аналогов.

1. Создать список разработчиков программного продукта с указанием их роли (в т.ч. разделив разработчиков на front-end и back-end).

2. Создать график встреч в соответствии с методологией по варианту.

3. Создать таблицу для оценки сложности задачи.

4. Создать список задач бэклога для продукта по варианту (не менее 10-ти задач). Этап разработки (начало разработки, середина и т.д. выбирается самостоятельно).

5. Исходя из модели, указанной по варианту, реализовать следующие требования.

Создать Kanban-доску (количество столбцов и статусы выбираются самостоятельно). Выбрать часть задач для реализации, проставить приоритет, дать оценку сложности. Назначить выбранным задачам разработчиков.

6. Проанализировать полученный спринт/доску, нагрузку на разработчиков. Сформулировать риски разработки описанных задач.

7. Смоделировать ситуацию добавления задачи, исполнение которой отложить невозможно. Описать задачу, задать ей сложность, приоритет.

Смоделировать прохождение какого-либо срока разработки (не менее 1 недели). Перевести задачи в разные статусы. Проанализировать задачи, взятые в работу.

9. Провести анализ того, как задача, добавленная после начала разработки, повлияла на результаты работы.

10. Для всех моделей: проанализировать полученный спринт/доску с точки зрения возможности применения модели DevOps и MSF.

## Результаты выполнения работы

Команда

Участник команды	Роль	
Иван Иванов	Back-end разработчик	
Петр Петров	Back-end разработчик	
Степан Степанов	Front-end разработчик	
Егор Егоров	Front-end разработчик	
Алексей Алексеев	Тестировщик	
Александр Александров	Менеджер проекта	

**Рисунок 1 - Список разработчиков программного продукта**

График...

Название встречи	День недели, время	Цель встречи	
Митинг (Standup)	Ежедневно, 14:00	Краткий отчет команды о текущих задачах и статусах.	
Операционная встреча	Последняя среда месяца, 10:00	Обсуждение текущих операционных вопросов и процессов.	
Планирование поставки	По решению команды	Планирование сроков и приоритетов поставок задач.	
Наполнение бэклога	Пятница, раз в две недели, 15:00	Пополнение и приоритизация задач в бэклоге.	
Обзор сервиса	Последняя среда месяца, 16:00	Анализ производительности и статуса сервиса	
Анализ рисков	Последняя среда месяца, 17:00	Выявление и оценка рисков в процессе разработки.	
Обзор стратегии	Первый понедельник квартала, 10:00	Обсуждение стратегического видения и планов на квартал.	

**Рисунок 2 – График встреч**

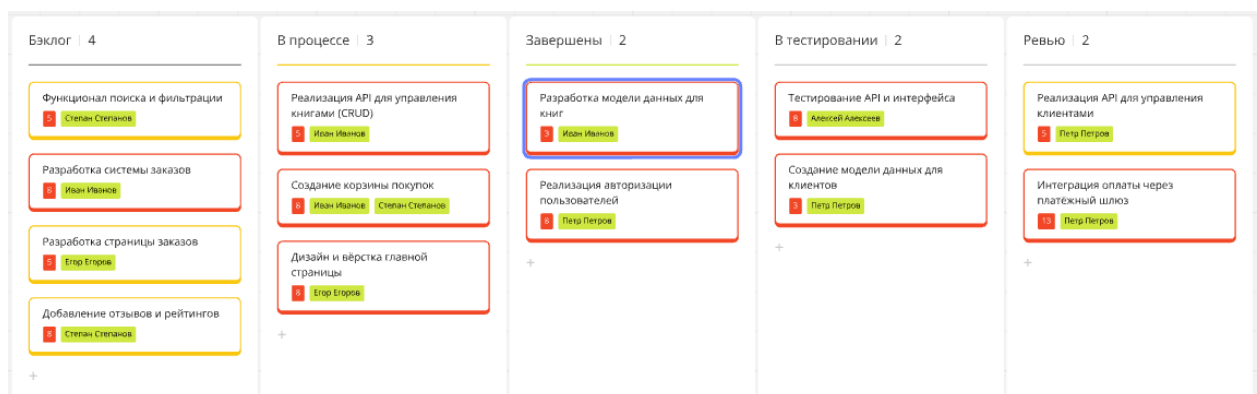
Сложн...

#	Оценка	Количество времени	Описание	
	1	1 рабочий час и менее	Очень простая задача, быстрое выполнение.	
	2	1-6 рабочих часов	Небольшая задача, не требующая сложных действий.	
	3	1 рабочий день	Задача среднего уровня сложности.	
	5	2 рабочих дня	Чуть более сложная задача, может требовать анализа.	
	8	2-3 рабочих дня	Комплексная задача, требует планирования.	
	13	4 рабочих дня	Крупная задача, разделенная на этапы.	
	20	5 рабочих дней	Очень большая задача с несколькими подзадачами.	
	40	10 рабочих дней	Масштабная задача, требует значительных ресурсов....	

**Рисунок 3 – Оценка сложности задач**

	Title	Description	
	Разработка модели данных для книг	Создать структуру данных для хранения информации о книгах, включая название, автора, жанр, цену и количество на складе.	
	Создание модели данных для клиентов	Разработать структуру для хранения данных клиентов: ФИО, email, телефон и адрес.	
	Реализация API для управления книгами (CRUD)	Разработать endpoints для создания, редактирования, удаления и получения информации о книгах.	
	Реализация API для управления клиентами	Разработать endpoints для создания и управления клиентскими записями.	
	Реализация модели и API для заказов	Разработать модель данных и API для создания заказов, добавления позиций в заказ и подсчёта итоговой стоимости.	
	Интеграция системы авторизации клиентов	Добавить регистрацию и вход для клиентов с валидацией email и паролей.	
	Создание дизайна и верстка главной страницы	Разработать современный адаптивный дизайн главной страницы с новинками, популярными книгами и поиском.	
	Реализация поиска и фильтрации книг	Добавить функционал поиска книг по названию, автору и жанру, а также фильтрацию по цене и наличию.	
	Разработка корзины покупок	Реализовать функционал добавления книг в корзину, изменения количества и удаления позиций.	
	Создание страницы заказов клиента	Разработать страницу, на которой клиент может просматривать историю своих заказов с деталями.	
	Интеграция оплаты через платёжный шлюз	Добавить интеграцию с платёжной системой для проведения онлайн-оплаты заказов.	
	Реализация уведомлений о статусе заказа	Добавить автоматические email-уведомления клиентам при изменении статуса их заказа (в обработке, отправлен, выполнен).	
	Разработка аналитики продаж для администраторов	Создать отчёты и дашборды для анализа количества продаж, прибыли и самых популярных книг.	

**Рисунок 4 – Бэклог задач**



**Рисунок 5 – Kanban-доска**

Нагрузка на разработчиков:

1. Иван Иванов (Back-end):

- В процессе: Реализация API для книг (5)
- Завершённая задача: Модель данных для книг (3)

2. Петр Петров (Back-end):

- Ревью: API для клиентов (5) и Интеграция оплаты через платёжный шлюз (13)
- Завершённая задача: Авторизация пользователей (8)

3. Степан Степанов (Front-end):

- В процессе: Создание корзины покупок (8)

- Backlog: Добавление отзывов и рейтингов (8)

4. Егор Егоров (Front-end):

- В процессе: Дизайн и вёрстка главной страницы (8)
- Backlog: Страница заказов (5)

5. Алексей Алексеев (Тестировщик):

- В тестировании: Тестирование API и интерфейса (8) и Модель данных клиентов (3)

6. Александр Александров (Менеджер проекта):

- Управляет распределением задач и приоритетами.

Общее состояние:

- Back-end разработчики сильно загружены задачами высокого приоритета.
- Front-end разработчики также работают над крупными задачами.
- Тестировщик близится к перегрузке, так как задачи на этапе тестирования растут.
- Backlog пополняется, но баланс между завершёнными и новыми задачами соблюдается.

Риски:

1. Возможная задержка в тестировании из-за высокой загрузки Алексея Алексеева.
2. Узкое место на этапе Ревью, что может затормозить завершение задач Петра Петрова.
3. Потенциальное перенаправление задач для выравнивания нагрузки.



**Рисунок 6** – Kanban-доска с добавленной неотложной задачей

#### Описание задачи:

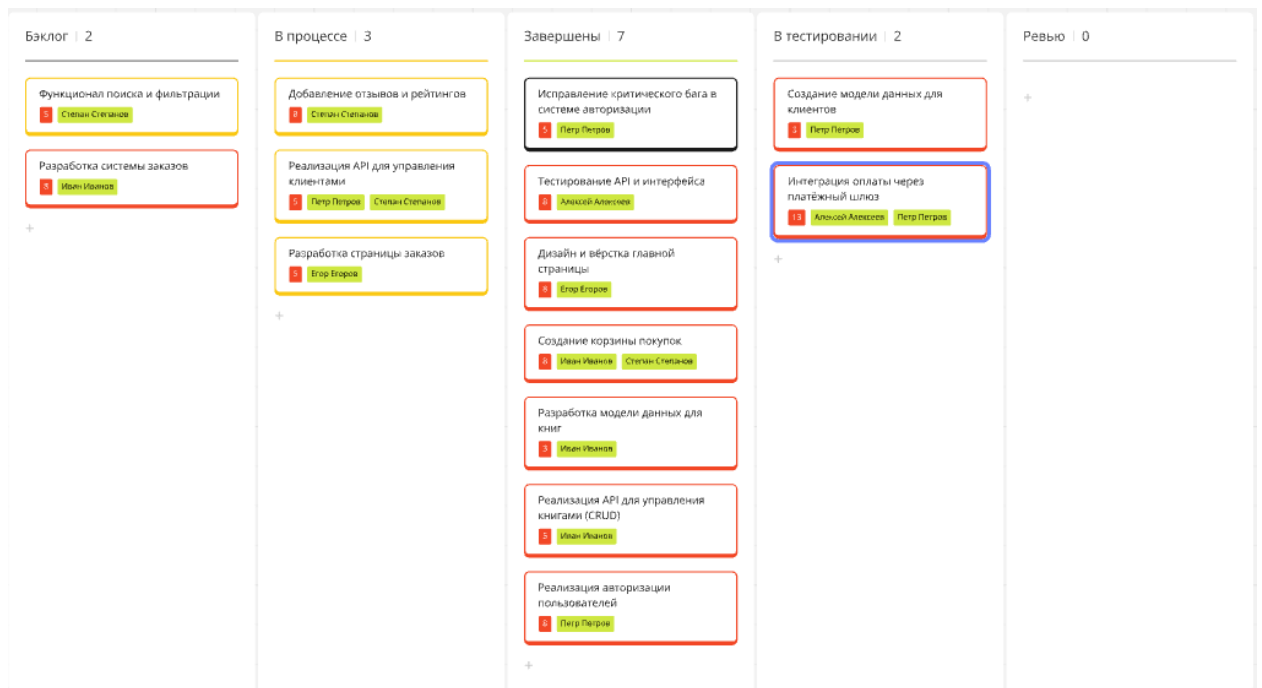
- Название: "Исправление критического бага в системе авторизации"
- Приоритет: Критический
- Сложность: 5 (2 рабочих дня)
- Ответственный: Петр Петров

#### Обоснование:

Баг влияет на авторизацию пользователей, что критично для работы магазина. Исправление необходимо срочно, даже несмотря на текущую загрузку разработчиков.

#### Результат:

- Петр Петров временно приостанавливает проверку задач на *Ревью*, чтобы исправить баг.
- Влияние: небольшая задержка задач в *Ревью* и временный рост нагрузки на Петра.



**Рисунок 7** – Kanban-доска после прохождения недели

Последствия добавления задачи:

- **Плюсы:** Критический баг был оперативно исправлен, что позволило восстановить функциональность авторизации и минимизировать риски для пользователей.
- **Минусы:**
  - Задержка в задачах *Ревью* из-за временного отвлечения Петра Петрова.
  - Увеличение времени на завершение других задач Петра.

Решение:

Менеджер проекта перераспределил задачи: Иван Иванов помог Петр Петрову на *Ревью*, что сократило задержку.

### Анализ Kanban-доски с точки зрения DevOps и MSF:

DevOps:

1. Автоматизация тестирования:
  - Внедрение CI/CD для автоматического тестирования новых изменений позволит сократить нагрузку на тестировщика.
2. Мониторинг и логирование:

- Добавление мониторинга ошибок поможет оперативно выявлять баги, как в случае с критическим багом авторизации.

### 3. Непрерывная интеграция и доставка (CI/CD):

- Постоянное внедрение новых функций без ручного вмешательства.

### MSF (Microsoft Solutions Framework):

#### 1. Проактивное управление рисками:

- Добавление задачи по исправлению бага показало способность команды оперативно реагировать на риски.

#### 2. Управление приоритетами:

- Использование Kanban позволяет гибко реагировать на изменения, включая срочные задачи.

#### 3. Контроль качества:

- Централизация тестирования и код-ревью обеспечивает высокое качество продукта.

**Вывод:** в ходе выполнения работы были сформированы и закреплены навыки по применению моделей разработки, используя модели DevOps, Agile, MSF.