|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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.

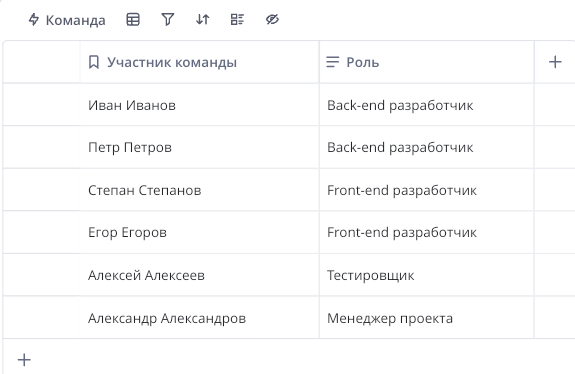
7. Смоделировать ситуацию добавления задачи, исполнение которой отложить невозможно. Описать задачу, задать ей сложность, приоритет.

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

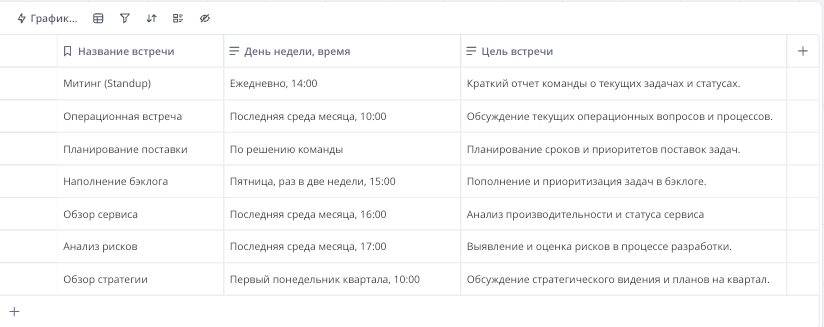
9. Провести анализ того, как задача, добавленная после начала разработки, повлияла на результаты работы.

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

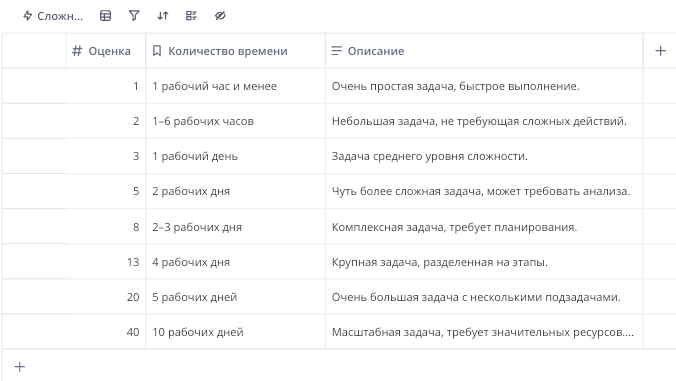
**Результаты выполнения работы**

****

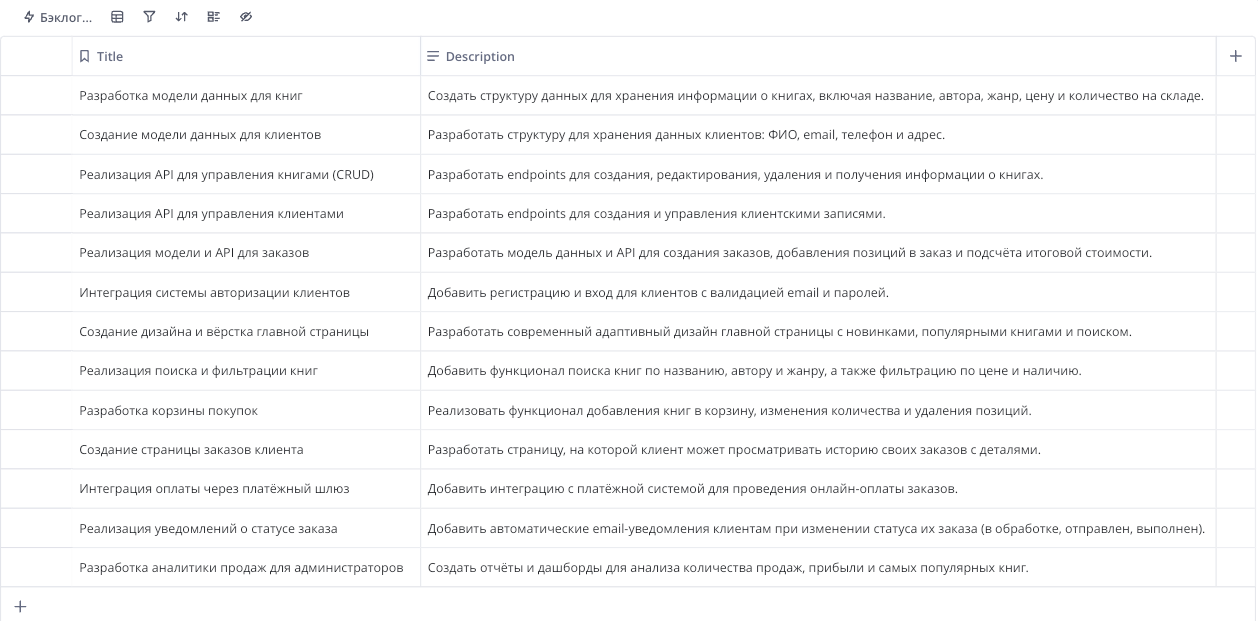
**Рисунок 1 -** Список разработчиков программного продукта



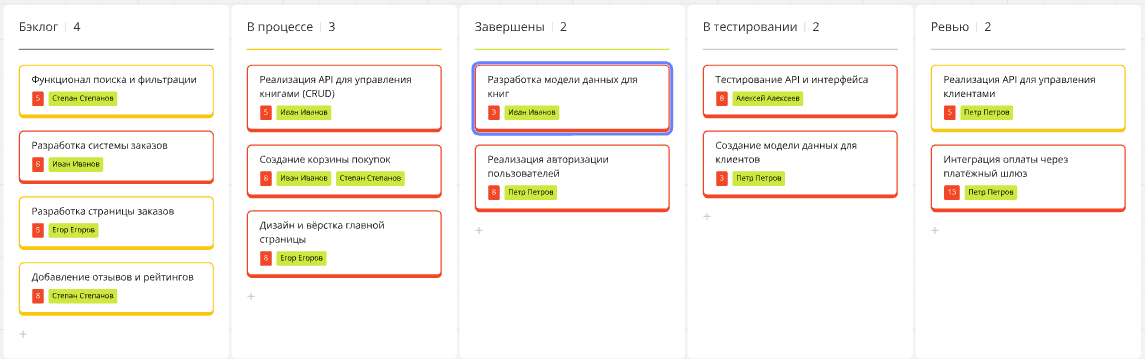
**Рисунок 2 –** График встреч



**Рисунок 3 –** Оценка сложности задач



**Рисунок 4 –** Бэклог задач



**Рисунок 5 –** Kanban-доска

Нагрузка на разработчиков:

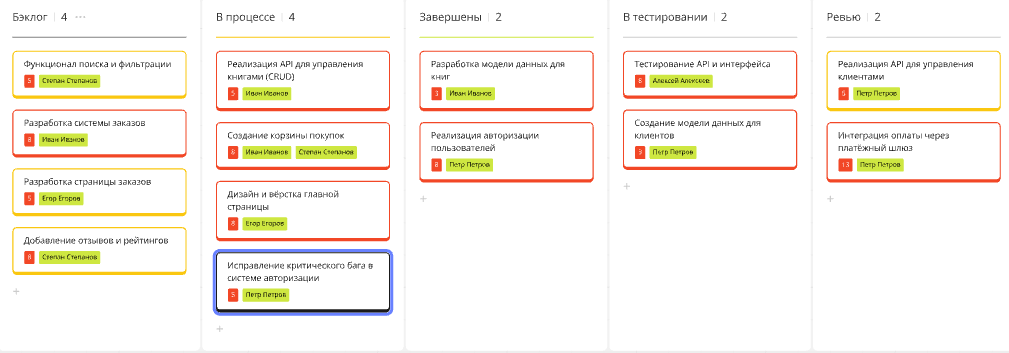
1. Иван Иванов (Back-end):
   * В процессе: Реализация API для книг (5)
   * Завершённая задача: Модель данных для книг (3)
2. Петр Петров (Back-end):
   * Ревью: API для клиентов (5) и Интеграция оплаты через платёжный шлюз (13)
   * Завершённая задача: Авторизация пользователей (8)
3. Степан Степанов (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:

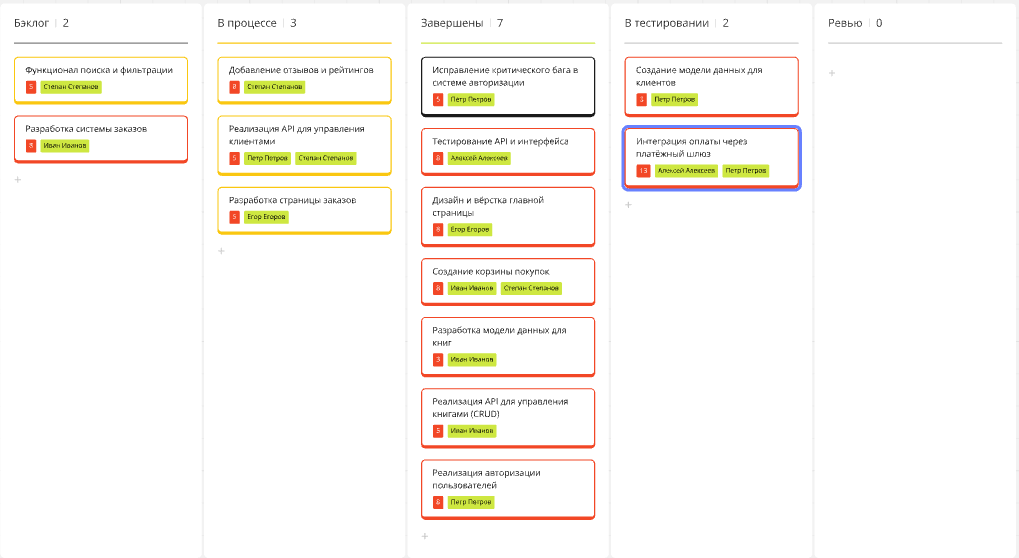
* Название: "Исправление критического бага в системе авторизации"
* Приоритет: Критический
* Сложность: 5 (2 рабочих дня)
* Ответственный: Петр Петров

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 навыков по применению моделей разработки, используя модели DevOps, Agile, MSF.