

Agile-документация проекта PollFlow

Версия 1.18 | Последнее обновление: 11.12.2025

1. Общая информация о проекте

Название продукта	PollFlow – «скорая помощь» для сбора мнений
Видение	Массовый сервис для оперативных решений с фокусом на скорость, простоту и доверие
Дата начала проекта	19 сентября 2025
Дата сдачи/защиты	12 декабря 2025
Методология	Гибкая гибридная (Scrum/Kanban)
Статус	MVP реализован

2. Команда и распределение ролей

Роль	Исполнитель	Ключевые обязанности
Product Owner / Владелец продукта	Михаил Сорокин (внешний заказчик)	Формирование Видения, приоритизация Бэклога, приемка результатов
Project Manager / Scrum Master	Браганец Александра	Фасилитация процессов, организация встреч, устранение блокеров, отчетность
Product Manager	Зверев Макар	Анализ рынка, работа с требованиями, позиционирование продукта

Business Analyst	Сердюк Александр	Детализация требований, анализ метрик, коммуникация с заказчиком
Team Lead / Архитектор	Дубровин Александр	Техническое руководство, проектирование архитектуры, разработка ML-модуля
Backend Developer	Ямщиков Никита	Разработка серверной логики, API, интеграция ML-модуля
Frontend Developer	Татаринцева Екатерина	Разработка пользовательского интерфейса (React)
DevOps Engineer	Пронин Михаил	Настройка инфраструктуры, CI/CD, развертывание, мониторинг
QA Engineer	Ефремочкин Андрей	Тестирование функционала, составление тест-кейсов, обеспечение качества

3. Рабочий процесс и инструменты

3.1. Цикл разработки (Спринт)

- Длина спринта:** 2 недели.
- Планирование спринта:** Проводится в **воскресенье** в начале каждого спринта.
- Ретроспектива/Синхронизация:** Проводится в **четверг** каждой недели для обсуждения прогресса и блокеров.

3.2. Инструменты

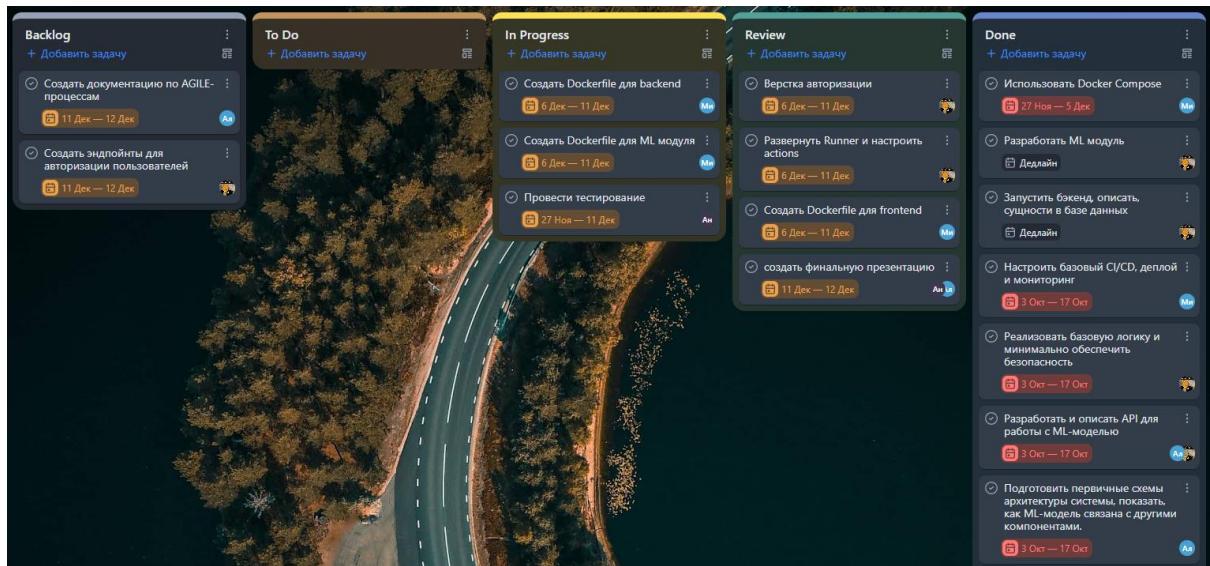
Категория	Инструмент	Назначение
Управление задачами	YouGile	Основная канбан-доска, роудмап

Управление кодом	GitHub	Хостинг репозиториев, контроль версий
Коммуникация	Telegram	Ежедневное оперативное общение, создание групповых чатов по задачам
Встречи	Яндекс Телемост	Проведение планирований, ретроспектив, демо-сессий
Документация	Telegram (ветка) / Git	Хранение рабочих файлов, финальная документация в репозитории
Демо	Живые встречи	Презентация инкремента заказчику и преподавателям

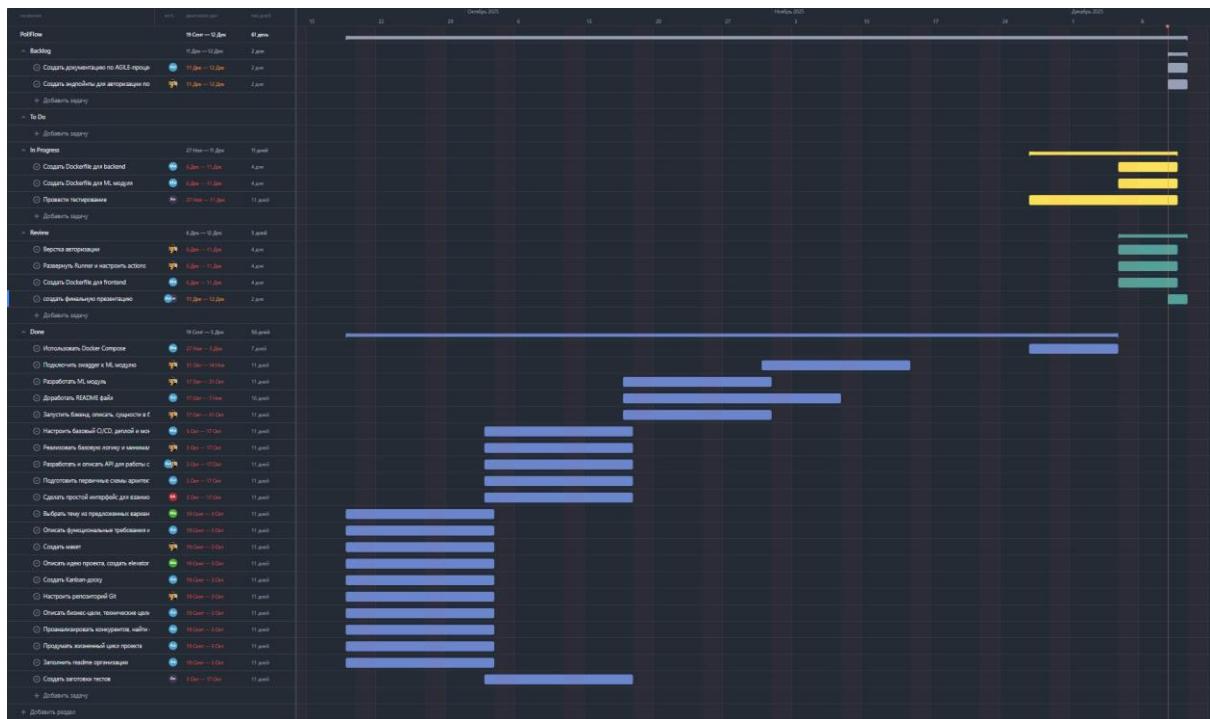
3.3. Workflow (Процесс работы над задачей)

1. **Инициация:** Задача создается в YouGile на основе требований из Product Backlog.
2. **Приоритизация:** Product Manager расставляют приоритеты.
3. **Взятие в работу:** На планировании спринта команда выбирает задачи из бэклога в колонку «**To Do**».
4. **Выполнение:** Разработчик перемещает задачу в «**In Progress**» и приступает к работе.
 - a. **Особенность процесса:** В связи с узкой специализацией (1 бэкенд, 1 фронтенд) практика обязательных **Pull Request** и **Code Review** не применяется. Валидация кода происходит через парное программирование на этапе интеграции и последующее тестирование QA-инженером.
5. **Тестирование:** Готовый функционал передается QA-инженеру, задача перемещается в «**Review**».
6. **Завершение:** После успешного тестирования задача перемещается в «**Done**».

Канбан-доски в YouGile:



Роудмап в YouGile:



4. Ритуалы и встречи

Встреча	Частота	Участники	Длительность	Цель
Сprint Планирование	Каждые 2 недели (Вс)	Вся команда + РО	1.5 часа	Определить цели и задачи на спринт, сформировать Sprint Backlog
Еженедельный Синк	Каждую неделю (Чт)	Вся команда	1 час	Обсуждение прогресса, выявление блокеров, корректировка планов
Демо-сессия	По окончанию спрингта	Вся команда + РО + стейкхолдеры	1 час	Демонстрация рабочего инкремента продукта, сбор обратной связи
Ретроспектива	Каждые 2 недели (после Демо)	Вся команда (без РО)	45 мин	Анализ процесса работы, выявление улучшений (Что хорошо?, Что улучшить?)
Ежедневный стендап	Асинхронно, ежедневно	Вся команда	15 мин (в чат)	Краткий отчет в Telegram: 1) Что сделал, 2) Что буду делать, 3) Блокеры

5. Артефакты

- **Product Backlog:** Единый перечень всех требований к продукту, хранится в YouGile.
- **Инкремент продукта:** Потенциально готовый к релизу кусок функционала, полученный по итогам спринта.
- **Definition of Done (DoD) – Критерии готовности задачи:**
 - Код написан и соответствует принятым стандартам.
 - Функционал протестирован разработчиком
 - Функционал протестирован QA-инженером
 - Код замержен в основную ветку (main/development).
 - Документация (комментарии, README) обновлена при необходимости.
 - Задача перемещена в колонку «**Done**» в YouGile.

6. Анализ рисков и принятые меры

Риск	Вероятность	Влияние	Принятые меры (Mitigation)
Проблемы с реализацией live-статистики	Высокая	Высокое	Упрощение архитектуры live-обновлений, использование готовых решений (WebSockets, Server-Sent Events)
Недостаток вычислительных мощностей для ML-модуля	Средняя	Высокое	Оптимизация модели, использование более легких моделей (например, TinyLLM), кеширование результатов
Задачи зависают в работе (In Progress)	Средняя	Среднее	Внедрение WIP-лимитов на колонку «In Progress» в YouGile, более детальное

			декомпозирование задач на этапе планирования
Узкая специализация (1 back, 1 front)	Высокая	Высокое	Активное использование парного программирования на стыке фронтенда и бэкенда, смещение акцента на интеграционное тестирование

Документ составлен командой PollFlow. Актуальность информации подтверждена на 11.12.2025.