

Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №2
по дисциплине «Экономика программной инженерии»

Вариант:
<https://supabase.com>
Kanban

Преподаватель:
Блохина Елена Николаевна

Выполнил:
Карташев Владимир
Группа: Р3415

Санкт-Петербург, 2025 г.

Оглавление

Оглавление.....	2
Текст задания.....	3
Анализ текущей ситуации.....	4
Перечень незавершенных функций и их приоритизация.....	5
План действий.....	6
Варианты выхода из ситуации.....	7
Заключение по выходу из сложившейся ситуации.....	9
Выводы по работе.....	10

Текст задания

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

Анализ текущей ситуации

Выполнено: $2857 \times 0.5 = 1429$ ч

Осталось: $2857 \times 0.5 = 1429$ ч

Оставшееся время: 1/4 от общего срока.

Для выполнения всего объема работ в оставшееся время потребовалось бы удвоить производительность, что на практике недостижимо без кардинальных изменений.

Перечень незавершенных функций и их приоритизация

Критичные функции

№	EPIC	Ключевые задачи	Трудоемкость (час)
1	Базовая инфраструктура	Docker Compose, .env, логирование	60
2	PostgreSQL Database	Развертывание, схемы, RLS	50
3	API Gateway (Kong)	Развертывание, базовая маршрутизация, JWT plugin	70
4	Authentication (GoTrue)	Регистрация/логин, JWT, Refresh-токены	100
5	REST API (PostgREST)	Генерация endpoints, JWT-интеграция, query-params	90
6	Row Level Security	Включение RLS, базовые политики для ролей	70
7	JavaScript SDK	createClient, базовые методы Auth и Database	180
8	Тестирование	Unit и интеграционные тесты для core-компонентов	120
9	Документация	Getting Started guide, API Reference	70
10	Релиз	Базовое production-развертывание	150
Итого по критичным функциям:			960

Можно отложить

№	EPIC	Ключевые задачи	Трудоемкость (час)
1	Storage API & SDK	Не является core-функцией BaaS. Система работоспособна без управления файлами.	350
2	Realtime Server & SDK	Не является core-функцией BaaS. Система работоспособна без управления файлами.	500
3	Расширенная аутентификация	Magic Link, OAuth 2.0 (Google/GitHub)	80
4	Расширенные возможности SDK	Retry Logic, продвинутая TypeScript-типовизация, все фильтры	100
5	Инфраструктура (продвинутая)	High Availability, сложный мониторинг, A/B-тестирование	200
Итого отложенных функций			1230

План действий

Первым шагом является немедленная и прозрачная коммуникация с заказчиком/стейххолдерами, в рамках которой представить текущее положение дел, анализ причин отставания и предложить согласованный план выхода из кризиса.

Шаг 1. Подготовка к встрече с заказчиком

- Подготовить наглядный отчёт: какие EPIC уже реализованы, какие в работе.
- Сформировать два списка: «MVP для релиза» (ядро продукта) и «Функции v1.1».
- Рассчитать сценарии завершения проекта с разными вариантами.

Шаг 2. Проведение встречи и согласование дальнейших действий

На встрече необходимо сообщить о проблеме, предложить варианты решений и совместно принять решение.

Варианты выхода из ситуации

Вариант 1: Релиз с урезанным функционалом (MVP)

Позволяет уложиться в исходные сроки без увеличения бюджета, но с уменьшением объема функционала:

- Фокусируемся на критических функциях (960 чч), которые составляют ядро продукта: аутентификация, база данных, автоматический REST API, базовый SDK и развертывание. Выпустить работоспособную первую версию, которая уже будет полезна разработчикам.
- Откладывается Storage, Realtime, расширенная аутентификация и другие улучшения
- Заказчик должен будет предоставить подтверждение на изменение объема работ (Scope Change Request)

Оставшееся время - $\frac{1}{4}$ срока = 714 часов = 5 недель

Исходная команда - 4 человека (оптимальный размер по PERT)

Рабочих часов в неделю на команду - 4 чел. * 25 часов = 100 часов в неделю

Критичные функции = 960 часов

Требуемое время = 960 / 100 = 10 недель

Итого: придется фокусироваться только на критических функциях и следить за работой подчиненных (т.к. была взята успеваемость в неделю 25 часов на человека)

Вариант 2: Сохранения полного функционала за счет увеличения команды

Позволяет реализовать все запланированные функции, но требует дополнительных инвестиций и сдвига сроков:

- Увеличиваем команду на 2 разработчика (например, Backend Developer и Frontend Developer);
- Заказчик должен будет согласовать увеличение бюджета до конца дедлайна

Оставшееся время - $\frac{1}{4}$ срока = 714 часов = 5 недель

Новая команда - 6 человек

Производительность новой команды будет ниже из-за затрат на онбординг (-20%)

Фактическая производительность = 6 - 20% = 4.8 "эффективных" разработчика

Рабочих часов в неделю на команду - 4.8 чел. * 25 часов = 120 часов в неделю

Требуемое время = 1429 / 120 = 12 недель

Итого: потребуется продление сроков, но функции будут реализованы

Вариант 3: Сохранения полного функционала за счет увеличения команды

Баланс между первым и втором вариантом:

- Увеличиваем команду на 1 человека (фуллстек)
- Частично отложить наименее критичные функции (например, Realtime & SDK)
- Заказчик должен будет увеличить бюджет и утвердить список откладываемых функций

Оставшееся время - $\frac{1}{4}$ срока = 714 часов = 5 недель

Новая команда - 5 человек

Требуемое время = $930 / (5 * 25\text{ч}) = 7$ недель

Итого: задержка всего на 2 недели

Заключение по выходу из сложившейся ситуации

Приоритет: сфокусироваться на реализации ядра системы (EPIC 1-7 в урезанном виде).

Команда: Увеличить на одного фулстек-разработчика для усиления команды.

Отложить: Реализацию Realtime-модуля (EPIC-10, EPIC-11) до версии v1.1. Storage (EPIC-8, EPIC-9) является следующим кандидатом на откладывание в случае возникновения новых рисков.

Процесс: После согласования с заказчиком:

1. Зафиксировать изменения в дополнении к договору.
2. Начать срочный рекрутинг 1 разработчика.
3. Ввести ежедневные миты для контроля прогресса и еженедельные демо для заказчика.

Выводы по работе

В ходе анализа кризисной ситуации проекта Supabase было установлено, что для выхода из состояния "50% готовности за 75% времени" необходимы срочные и решительные меры.

Наиболее реалистичным и сбалансированным представляется Гибридный подход (Вариант 3), который предполагает незначительное увеличение команды и откладывание наиболее трудоемкого, но не критичного для MVP функционала (Realtime). Это позволит с минимальными потерями выпустить на рынок работоспособную и полезную версию продукта в срок, близкий к первоначальному, сохранив доверие заказчика и команды.

Проделанная работа подтвердила важность гибкого планирования, постоянного мониторинга прогресса и открытой коммуникации для успешного управления ИТ-проектами.