

Задания

Часть 1. Идентификация проблем

Проанализируй "заметки" и выдели 3-5 ключевых проблем в текущем управлении

проектом. Сформулируй их в виде списка.

Пример формулировки проблемы:

1. "Отсутствие четких приоритетов: команда получает противоречивые указания от CEO (монетизация vs. иконка)."

ПРОБЛЕМЫ:

Продублирую приведённый в качестве примера пункт, как первостепенно значимый для эффективной работы:

1. Отсутствие четких приоритетов в работе над продуктов - необходима работа с бэклогом (при его отсутствии - составление), выбор приоритетных задач для целей бизнеса.

Составление конкретного плана по работе над продуктом для его завершения, без "прыжков" от одной фичи к другой.

2. Проблемы с коммуникацией и зависимостью между командами, отсутствие единого источника информации по работе команды (указаны - почта, Slack, Figma, Trello).

Несогласованность в выполнении задач и неполное понимание бизнес-требований в команде.

3. Блокировка задач и отсутствие прозрачности (возможности наблюдения статуса задачи) в их выполнении. Простой команды, выполнение задач с более низким приоритетом.

Нет конкретного описания статусов задач с соответствующими необходимыми атрибутами - название задачи, модуль, исполнитель, когда была взята в работу, дэдлайн по задаче. Наверняка аналогичная ситуация на всех этапах SDLC в описании работы над задачами, если таковые вообще ведутся в таск-трекере.

4. Нестабильное техническое состояние продукта на данный момент, технические риски для будущего релиза (падение БД на тестовом стенде уже сейчас при минимальной нагрузке).

5. Также на основании вводного части данного ТЗ необходимо отметить ПРОБЛЕМУ демотивированности команды, очевидно, ввиду чёткого понимания целей разработки, частой смены приоритетов, необходимости переделывать уже готовые задачи либо невозможности их завершения.

Часть 2. Выбор методологии и первые шаги

1. Какую методологию ты бы предложил внедрить этой команде в первую очередь?

Обоснуй свой выбор (2-3 предложения), опираясь на проблемы, которые ты выявил в Части 1.

На основании выявленных проблем, я бы решил использовать Agile фреймворк Scrum в силу его преимуществ, а именно гибкости для быстрой адаптации и реорганизации приоритетов, а также регулярных синхронизаций (в виде т.н.. "церемоний" Scrum - дэйли, ретро, ревью, демо) для улучшения коммуникации и прозрачности.

Таким образом, в нашем случае Scrum поможет структурировать работу и разбить проект на управляемые спринты с чёткими целями и ролями.

2. Ты решил провести первую встречу с командой, чтобы навести порядок. Составь краткую повестку этой встречи. Она должна включать 3-4 ключевых пункта, которые нужно обсудить.

В качестве основной цели первой встречи я выделю:

Внести понимание текущей ситуации работы над проектом - составление общей картины проекта, существующих проблем и поиск способов их разрешения.

Ключевые пункты для обсуждения:

- Обозначение и обсуждение текущих приоритетов и бизнес-целей - что самое важное сейчас в работе над продуктом, определить ключевой функционал. Остальные "фичи" - отложить во второй этап, чтобы команда увидела финишную черту первой версии.

- Обзор существующих проблем и препятствий - технические, коммуникационные
- По пункту из письма СЕО "Нужно срочно добавить в приложение оплату подписки!" выяснить у команды - есть ли хоть какие-то шаги в этом направлении.
- Согласование со всеми членами команды новых единых правила работы - введение подхода Scrum: роли, встречи, артефакты.
- Обсуждение плана по привязке задач к единому бэклогу и назначение ответственных. Ведение задач строго в выбранном таск-трекере.

3. На основе анализа ситуации создай в любом простом виде бэклог:

Backend - Определить и задокументировать API для авторизации.

Android - Интегрировать готовое API авторизации.

iOS - Провести нагружочное тестирование БД (QA) и подготовить план по устранению проблемы с производительностью (devops, iOS backend).

Дизайн/РМ - Собрать актуальный контент (упражнения) для раздела "Тренировки" и передать дизайнеру.

Часть 3. Коммуникация со стейкхолдером:

Добрый день, уважаемый "CEO"!

Спасибо за оперативное включение в проект и обозначение ключевых бизнес-целей - монетизация (оплата подписки) и выпуск приложения к выставке «Спорт-2025».

Я взял проект под свой контроль и уже провел аудит текущего состояния.

По статусу проекта:

Чтобы достичь цели по монетизации в срок, я сфокусировал команду на приоритетных задачах и ввожу прозрачную систему управления проектом по Scrum.

Вопрос с оплатой подписки и прочими важными фичами прорабатывается в первую очередь. Я подготовлю для вас детальный план-график реализации функции оплаты подписки, включая все этапы: анализ, дизайн, разработку, тестирование и релиз.

Запрос на редизайн иконки приложения и добавление новых разделов (питание/рецепты) я пока вынес во второй этап, чтобы не распылять ресурсы и гарантировать своевременный выход MVP с монетизацией к указанным срокам.

Мои первые шаги:

Сегодня провожу встречу с командой для синхронизации и устранения текущих блокировок - проблемы с API и нестабильность базы данных.

К концу недели представлю вам актуальный, реалистичный план по внедрению подписки с четкими сроками и ответственными.

С уважением, Алексей,

РМ проекта «Фитнес-гуру»

Часть 4 Диаграмма Ганта

Декомпозиция, длительность, зависимости:

1. Анализ требований и проработка UX/UI:

Сбор требований и создание UX/UI макетов формы оплаты.

Длительность - 2 дня.

Зависимость - нет.

2. Дизайн UI оплаты подписки:

Разработка полноценного дизайна, визуализация платежных экранов.

Длительность - 2 дня.

Зависимость - этап 1.

3. Backend разработка:

Реализация серверной части - интеграция с платёжными системами, база данных, API.

Длительность - 7 дней.

Зависимость - этап 1.

4. Frontend разработка:

Кодирование компонентов UI для оплаты, интеграция с backend API.

4.1. iOS - разработка интерфейса и логики

4.2. Android - разработка интерфейса и логики

Длительность - 7 дней.

Зависимость - этап 1, этап 2.

5. Интеграция frontend и backend:

Объединение фронтенда и бэкенда, тестирование совместной работы.

Длительность - 3 дня.

Зависимость - этап 3, этап 4.

6. Тестирование:

Функциональное, интеграционное и нагрузочное тестирование.

Длительность - 4 дня.

Зависимость - этапы 1-5. Тестирование можно начать после первого этапа - тестирование документации, и так на каждом этапе разработки.

7. Исправление багов (багфиксинг), доработки и улучшения по результатам тестирования, ретестинг, регрессионное тестирование.

Длительность - 3 дня.

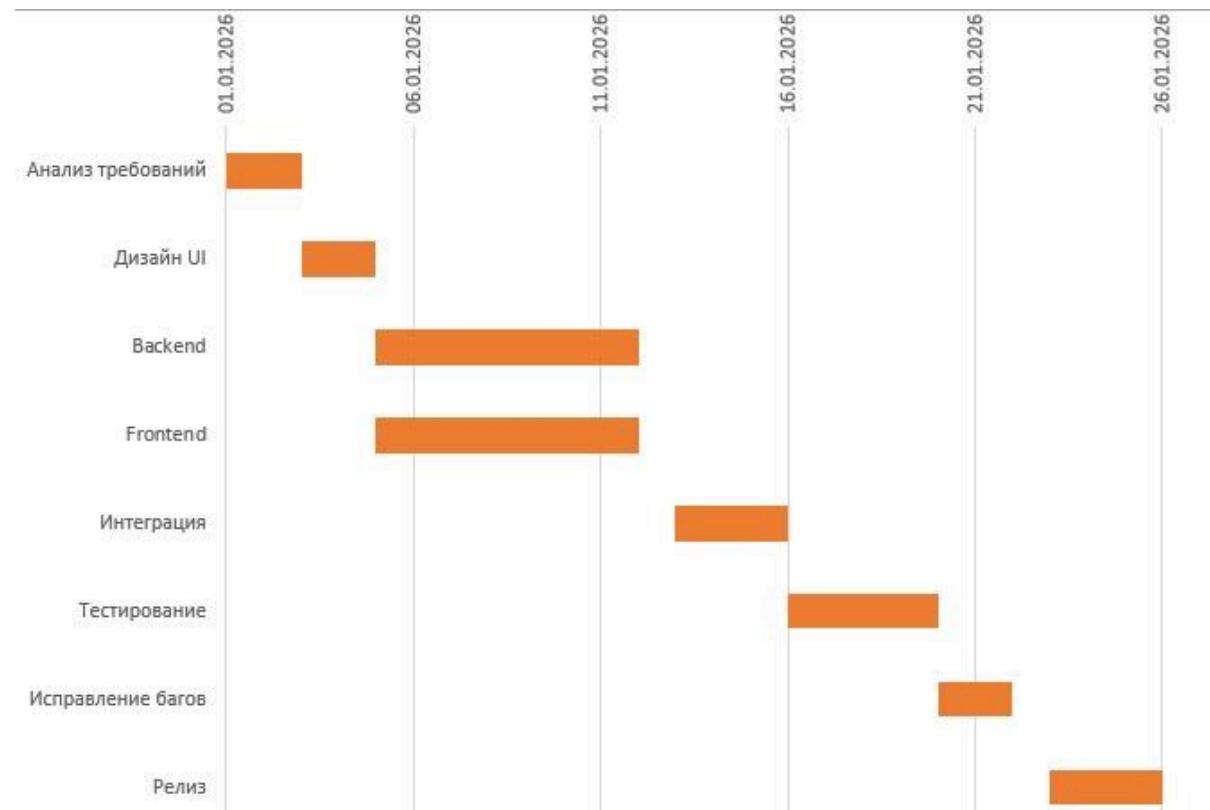
Зависимость - этап 6.

8. Подготовка к релизу и выпуск:

Приёмочное тестирование, подготовка документации, деплой на продакшен

Длительность - 3 дня.

Зависимость - этап 7.



Обоснование плана

Данный план основан на классическом поэтапном подходе (анализ -> дизайн -> разработка -> тестирование -> отладка -> релиз), что обеспечивает последовательность и управляемость проекта.

Основные риски и допущения:

- Нестабильность работы с базой данных может замедлить этап разработки и тестирования, так как все новые функции зависят от стабильности сервера. Этую проблему необходимо решить до начала разработки фичи платежей.
- Возможны непредвиденные задержки на этапах работы - технические и в работе команды (болезнь, увольнение сотрудников).
- Также из рисков можно было бы добавить следующие пункты - ошибки при интеграции с внешними сервисами платежей и последующие задержки в разработке.
- В целом закладывание ресурсов на риски увеличивает сроки и стоимость проекта, поэтому в условиях временных ограничений приходится использовать стратегию принятия риска.