

Часть 1. Идентификация проблем

- ⑩ Отсутствие четких приоритетов: CEO дает противоречивые указания, что приводит к рассеиванию усилий команды.
- ⑩ Блокирующие зависимости между ролями: Android-разработчик не может завершить авторизацию из-за задержки API от backend-разработчика, что приводит к простое и переключению на неприоритетные задачи вроде рефакторинга.
- ⑩ Отсутствие плана релиза и дедлайнов: Маркетолог запрашивает дату релиза для подготовки к выставке через 2 месяца, но нет общей видимости сроков, что усиливает неопределенность.
- ⑩ Проблемы с технической стабильностью: iOS-разработчик сообщает о падениях базы данных даже при малой нагрузке, что указывает на риски масштабируемости.
- ⑩ Неопределенность в требованиях и контенте: Дизайнер подготовил макеты с заглушками, но не знает деталей упражнений, что отражает отсутствие детализированных спецификаций для задач

Часть 2. Выбор методологии и первые шаги

1. Я бы предложил внедрить методологию Scrum. Это позволит установить четкие приоритеты через спринты и бэклог. Scrum внесет структуру в хаотичный процесс, да и это ИТ сфера, это классическая методология в этой сфере.
2. Повестка первой встречи с командой:
 - ⑩ Обзор текущего состояния: Обсудить ключевые проблемы и собрать обратную связь от каждого члена команды для выявления скрытых болей.
 - ⑩ Установление приоритетов: Определить топ-3 цели на ближайшие 2 недели(на время первого спринта).
 - ⑩ Планирование инструментов: Выбрать и настроить общий инструмент (например, Trello или Jira) для трекинга задач.
 - ⑩ Распределение ролей и следующий шаг: Назначить ответственных за блоки (например, backend для API) и запланировать ежедневные стендапы.
3. Бэклог из первых задач:
 - ⑩ Провести аудит зависимостей: Собрать список всех блокирующих проблем (например, API для авторизации) и назначить сроки их разрешения.
 - ⑩ Определить спецификации для контента: Собрать от команды детали упражнений и рецептов для обновления макетов в Figma.
 - ⑩ Исправить проблему с базой данных: Проанализировать и оптимизировать БД на тестовом стенде для поддержки хотя бы 50 пользователей.
 - ⑩ Установить дату релиза: Согласовать с маркетологом и CEO реалистичный дедлайн с учетом выставки через 2 месяца.
 - ⑩ Приоритизировать монетизацию: Разработать план внедрения оплаты подписки как ключевой фичи.

Часть 3. Коммуникация со стейкхолдером

Здравствуйте, ценю ваше время, поэтому отвечаю по пунктам.

-По поводу добавления оплаты в приложение, я организую встречу с командой завтра для распределения задач и составлю план с дедлайнами на неделю вперед, сразу после этого я с вами свяжусь для дальнейшего информирования.

Project Manager, "Фитнес-гуру"

Обоснование плана: План построен по классическому жизненному циклу разработки с акцентом на параллелизацию задач там, где это возможно (например, дизайн UI и разработка бэкенда), возможно выполнение или начало выполнения и других задач раньше, просто я брал просто стандартный вариант, чтобы сократить общее время. Основные риски, заложенные в оценку: возможные доработки API со стороны платежного провайдера и необходимость исправления критических багов во время тестирования.

Декомпозиция "Внедрение оплаты подписки"					Количество дней																						
Этап / Задача	Длительность (раб. дней)	Начало	Окончание	Зависит от	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1. Анализ и проектирование	5																										
1.1. Анализ требований и выбор платежного провайдера	3	День 1	День 3			1.1																					
1.2. Проектирование API для работы с подписками	2	День 4	День 5	1.1				1.2																			
2. Дизайн UI/UX	5																										
2.1. Создание макетов экранов оплаты и управления подпиской	4	День 1	День 4	1.1		2.1																					
2.2. Согласование макетов с CEO	1	День 5	День 5	2.1					2.2																		
3. Backend-разработка	10																										
3.1. Реализация API для подписок	5	День 6	День 10	1.2										3.1													
3.2. Интеграция с API платежного провайдера	5	День 6	День 10	1.1										3.2													
4. Frontend-разработка	10																										
4.1. Разработка UI экранов оплаты (iOS)	5	День 6	День 10	2.2										4.1													
4.2. Разработка UI экранов оплаты (Android)	5	День 6	День 10	2.2										4.2													
4.3. Интеграция с бэкенд-API (iOS/Android)	5	День 11	День 15	3.1, 4.1, 4.2											4.3												
5. Тестирование	7																										
5.1. Написание тест-кейсов	2	День 11	День 12	1.2, 2.2											5.1												
5.2. Функциональное и интеграционное тестирование	5	День 16	День 20	4.3, 3.2																5.2							
6. Релиз и документация	3																										
6.1. Подготовка к публикации (скриншоты, описание)	2	День 21	День 22	5.2																					6.1		
6.2. Публикация в App Store / Google Play	1	День 23	День 23	6.1																							6.2
Итого					~23 (это примерное количество дней)																						