

12+

НАЧИНАЮЩЕМУ РУКОВОДИТЕЛЮ ПРОЕКТА В IT

ДМИТРИЙ БОЛЕСОВ



Дмитрий Болесов

**Начинающему
руководителю проекта в IT**

«Издательские решения»

Болесов Д.

Начинающему руководителю проекта в IT / Д. Болесов —
«Издательские решения»,

Пошаговые решения для начинающего РМ: как стартовать проект, договариваться с командой и заказчиком, оценивать сроки, реагировать на изменения и не выгорать. Примеры и шаблоны для веб- и мобильной разработки.

Содержание

Введение	6
Зачем эта книга	6
Что на самом деле делает РМ в IT	7
Мифы и страхи начинающего РМ	8
Как устроена книга	9
Глава 1. С чего начинается проект: первый разговор со стейкхолдером	10
Заказчик не знает, чего хочет, и это нормально	10
Вопросы, которые спасают проект до старта	11
Как зафиксировать ожидания	12
Пример: расшифровка расплывчатой задачи	13
После встречи РМ формулирует	14
Глава 2. Команда и роли: кто за что отвечает и как договариваться	15
Кто в команде	15
Как выстроить первые договорённости	16
Пример: сценарий встречи «знакомство с командой»	17
Культура обратной связи	18
Конец ознакомительного фрагмента.	19

Начинающему руководителю проекта в IT

Дмитрий Болесов

© Дмитрий Болесов, 2026

ISBN 978-5-0071-3510-8

Создано в интеллектуальной издательской системе Ridero

Введение

Зачем эта книга

Вы стали руководителем проекта. Возможно, вчера вы были разработчиком, тестировщиком, аналитиком или вообще пришли из смежной области. А сегодня вам поручили вести проект, и внутри крутится вопрос: «А я справлюсь?»

Эта книга не сделает вас идеальным РМ за один вечер. Но она даст набор практических инструментов, которые можно применять с первого дня: как понять, чего хочет заказчик, как спланировать веб- или мобильный проект, чтобы он не развалился к релизу, как говорить с командой и стейкхолдерами, как замечать риски до того, как они станут катастрофой.

Здесь нет абстрактных теорий менеджмента. Есть конкретные сценарии, шаблоны, чек-листы и разборы реальных ситуаций из веб- и мобильной разработки. Всё — без таблиц, простым языком, с примерами, которые можно сразу использовать.

Что на самом деле делает РМ в IT

Представьте себе мост. С одной стороны — бизнес, которому нужен результат: новый сайт, мобильное приложение, доработка существующего продукта. С другой — команда, которая этот результат создаёт: разработчики, дизайнеры, тестировщики, DevOps. РМ — это и есть мост. Без него бизнес не понимает, почему «простая кнопка» занимает две недели, а команда не понимает, зачем вообще нужна эта кнопка.

РМ не пишет код, не рисует дизайн, не тестирует баги. РМ делает так, чтобы все остальные могли делать свою работу эффективно. Это значит:

- **Понимать цель проекта** и уметь объяснить её команде так, чтобы все тянули в одну сторону.

- **Согласовывать ожидания** между заказчиком и командой, чтобы в конце не оказалось, что все понимали по-разному.

- **Управлять сроками и ресурсами**, не обещая невозможного и не впадая в микроменеджмент.

- **Замечать риски** раньше, чем они станут проблемами.

- **Фасилитировать коммуникацию**, чтобы важное не терялось в переписках и встречах.

Мифы и страхи начинающего РМ

«Я недостаточно техничен» — самый частый страх. Вам не нужно уметь писать код. Вам нужно понимать, как устроен процесс разработки, уметь задавать правильные вопросы и доверять команде в технических деталях. В веб-проектах достаточно понимать разницу между фронтендом и бэкендом, знать, что такое API и зачем нужен деплой. В мобильных — понимать, что iOS и Android требуют разных подходов, и почему релиз в App Store занимает больше времени, чем в Google Play.

«Меня не будут слушать, потому что я не разработчик» — вас будут слушать, если вы прозрачны, честны и защищаете команду от хаоса. Разработчики ценят РМ, который не обещает заказчику невыполнимого и не заставляет делать бессмысленную работу.

«Всё сломается в последний день» — и это случится. Но если вы планировали риски, у вас будет план Б. Если нет — будет паника. Эта книга как раз про то, чтобы план Б был.

Как устроена книга

Каждая глава — это самостоятельный блок, который можно читать отдельно, но лучше по порядку. Внутри глав вы найдёте:

- **Примеры** — разборы реальных ситуаций из веб- и мобильной разработки.
 - **Сценарии разговоров** — что сказать и как, чтобы не обидеть и добиться результата.
 - **Чек-листы** — что проверить перед важными этапами.
 - **Шаги** — конкретные действия, которые можно сделать сегодня.
- Начнём.

Глава 1. С чего начинается проект: первый разговор со стейкхолдером

Заказчик не знает, чего хочет, и это нормально

Веб- и мобильные проекты редко начинаются с чёткого технического задания. Чаще звучит что-то вроде:

- «Нам нужен новый сайт, современный, чтобы продавал».
- «Хотим приложение, как у конкурентов, только лучше».
- «Нужно переделать админку, текущая — кошмар».

Это не плохо. Это нормально. Задача РМ на старте — не получить идеальное ТЗ, а помочь заказчику превратить расплывчатое желание в конкретные, измеримые цели. Для этого нужен разговор.

Вопросы, которые спасают проект до старта

На первой встрече со стейкхолдером задайте эти вопросы. Запишите ответы. Отправьте их заказчику после встречи с просьбой подтвердить — это ваша первая страховка от «я имел в виду другое».

О цели:

— Какую бизнес-задачу должен решить проект? (Не «нужен сайт», а «увеличить количество заявок с сайта на 30% за полгода».)

— Что произойдёт, если проект не запустится? (Помогает понять реальную значимость и приоритет.)

— Как вы поймёте, что проект успешен? Какие метрики?

О границах:

— Что обязательно должно войти в первую версию, а что можно отложить?

— Что точно НЕ должно быть в проекте? (Иногда это важнее, чем список «хотелок».)

— Есть ли существующие системы, с которыми нужно интегрироваться? (CRM, платёжные шлюзы, аналитика — частый источник сюрпризов в веб-проектах.)

О пользователях:

— Кто будет пользоваться сайтом или приложением? Возраст, устройство, контекст.

— Для мобильных проектов: какие платформы нужны — iOS, Android? Какие версии? Только телефон или планшет тоже?

— Для веб-проектов: какие браузеры и устройства нужно поддерживать? (Если заказчик скажет «все» — это красный флаг, нужно уточнять.)

Об ограничениях:

— Какой бюджет? Хотя бы вилкой — «до X» или «от X до Y».

— Какие сроки жёсткие, а какие гибкие? Есть ли внешняя дата (конференция, запуск сезона, договор)?

— Кто будет принимать результат и подписывать этапы?

О рисках и зависимостях:

— От чего зависит запуск, что вне вашего контроля? (Релиз в App Store, доступы к серверам, согласование дизайна у руководства.)

— Были ли раньше попытки сделать этот проект? Если да — что не получилось и почему?

Как зафиксировать ожидания

После первой встречи отправьте заказчику короткое резюме — одну страницу, не больше. Структура:

- **Цель проекта** — одним предложением.
- **Что входит в первую версию** — нумерованный список, 5—10 пунктов.
- **Что НЕ входит** — отдельный список, чтобы избежать иллюзий.
- **Ключевые ограничения** — сроки, бюджет, платформы.
- **Следующие шаги** — что и когда делаете вы, что нужно от заказчика.

Попросите подтвердить письменно. Это не формальность — это ваш щит на будущее, когда через два месяца заказчик скажет: «А мы думали, сюда ещё и личный кабинет войдёт».

Пример: расшифровка расплывчатой задачи

Заказчик говорит: «Нам нужно мобильное приложение для доставки еды. Как у X, но удобнее».

РМ уточняет:

- Что значит «удобнее»? Что именно в приложении конкурента вам не нравится?
- Какие платформы нужны? iOS, Android или обе?
- Доставка только по городу или по всей стране?
- Оплата внутри приложения или при получении?
- Нужна ли система лояльности, промокоды, push-уведомления?
- Есть ли готовый бэкенд или его тоже нужно делать?
- Кто будет наполнять каталог блюд? Нужна ли админка?
- Какой бюджет и к какому сроку нужен MVP?

После встречи РМ формулирует

- Проект: мобильное приложение для доставки еды, MVP.
 - Платформы: iOS и Android (Flutter/React Native — техническое решение команды).
 - В первой версии: каталог блюд, корзина, оплата картой внутри приложения, отслеживание статуса заказа, push-уведомления.
 - НЕ входит в первую версию: система лояльности, отзывы, мультиязычность, веб-версия заказа.
 - Срок MVP: 3 месяца.
 - Ограничения: нужен готовый бэкенд или его разработка; нет дизайн-макетов — нужен дизайнер.
- Разница между «как у X, но удобнее» и этим списком — это разница между провалом и управляемым проектом.

Глава 2. Команда и роли: кто за что отвечает и как договариваться

Кто в команде

Веб- и мобильные проекты обычно собираются из следующих ролей. В небольших командах один человек может совмещать несколько, но функции остаются теми же.

Продуктовый дизайнер (UI/UX) — отвечает за то, как продукт выглядит и как им пользоваться. В веб-проектах готовит макеты экранов, прототипы, адаптации под мобильные устройства. В мобильных — учитывает нативные гайдлайны (Human Interface Guidelines для iOS, Material Design для Android).

Фронтенд-разработчик — превращает макеты в работающие веб-страницы. Веб: HTML, CSS, JavaScript, фреймворки (React, Vue и т. д.). В мобильных проектах фронтендом часто называют клиентскую часть приложения.

Бэкенд-разработчик — пишет серверную логику, API, работает с базой данных. Отвечает за то, что данные приходят и уходят правильно.

Мобильный разработчик — пишет нативное (Swift/Kotlin) или кроссплатформенное (Flutter, React Native) приложение. Часто два разработчика — по одному на платформу, или один кроссплатформенный.

Тестировщик (QA) — проверяет, что всё работает. В веб-проектах тестирует разные браузеры и устройства. В мобильных — разные версии OS, размеры экранов, сценарии прерываний (звонок во время оформления заказа).

DevOps-инженер — настраивает инфраструктуру: серверы, CI/CD, деплой, мониторинг. В небольших проектах эту роль часто берёт на себя бэкенд-разработчик.

Аналитик — собирает и формализует требования, описывает пользовательские сценарии, составляет документацию. В небольших командах эту работу часто делает PM.

Как выстроить первые договорённости

На первой встрече с командой не нужно сразу загружать задачами. Нужно договориться о правилах игры. Вот минимум:

Каналы коммуникации:

— Где обсуждаются задачи? (Таск-трекер, чат, встречи.)

— Где принимаются решения? (Важно: «мы обсудили в курилке» — это не решение. Решение зафиксировано там, где все видят.)

— Как сообщать о проблемах? (Разработчик застрял на задаче два дня — это норма или повод для эскалации?)

Встречи:

— Какие встречи нужны и как часто? (Дейли, планирование, ретроспектива.)

— Кто готовит повестку? Кто фасилитирует?

— Что делать, если встреча не нужна? (Отменить. Без сожалений.)

Эскалация:

— Кто принимает решения, если РМ и команда не согласны?

— Как и когда поднимать проблему к руководству заказчика?

— Что считать «красным флагом», требующим немедленного внимания?

Правило обратной связи:

— О проблемах сообщают сразу, а не в день дедлайна.

— РМ не наказывает за честность — наоборот, благодарит за ранний сигнал.

— Обратная связь по работе даётся команде регулярно, а не только на ретроспективе.

Пример: сценарий встречи «знакомство с командой»

РМ открывает встречу:

«Привет! Я — ваш РМ на проекте доставки еды. Моя задача — чтобы вы могли работать без хаоса и перегруза, а заказчик получал то, что ожидает. Расскажите, как вам удобнее: какие инструменты, какие встречи, как сообщать о проблемах. Я не буду микроменеджить, но мне важно знать статус, чтобы защищать вас перед заказчиком».

Вопросы команде:

— С какими инструментами вам комфортно работать? (Jira, Trello, GitHub Projects — что угодно, главное, чтобы все согласились.)

— Как часто нужен дейли? Каждый день или через день?

— Есть ли что-то, что вас раздражало в прошлых проектах? (РМ, который обещал заказчику без согласования с командой — частый ответ.)

— Кто будет оценивать задачи — вы сами, или мне нужно организовать отдельную встречу с оценкой?

РМ фиксирует договорённости и отправляет в общий чат:

«Итак, договорились: задачи в Jira, дейли в 11:00, оценки дает разработчик, я согласовываю сроки с заказчиком только после вашей оценки. Проблемы пишем в чат сразу. Если что-то блокирует — тегаеете меня».

Культура обратной связи

Обратная связь — это не критика, это калибровка. Правило простое: говорите о работе, а не о человеке. Не «ты медленный», а «задача, которую мы оценили в два дня, идёт четвёртый. Что мешает? Как помочь?».

Три правила, которые работают:

— **Регулярность важнее формы.** Лучше пять минут каждый день, чем час раз в месяц.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «Литрес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на Литрес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.