
title: "База DevRel" subtitle: "Руководство для начинающих DevRel специалистов" author: "Рафаэль Миниахметов" date: "Октябрь 2025" version: "1.0"

БАЗА DEVREL

Практическое руководство для начинающих специалистов

Developer Relations и бренд работодателя в IT

Автор: Рафаэль Миниахметов

Соавтор: Ксения Романова

Октябрь 2025

Содержание

Введение

Часть 1. Основы DevRel

1. Что такое DevRel
2. Компетенции DevRel
3. Карьерный путь в DevRel

Часть 2. Бренд работодателя

4. Что такое Employer Brand
5. Построение бренда работодателя
6. Каналы продвижения Employer Brand

Часть 3. Сообщество и контент

7. Community Building
8. Контент для разработчиков
9. Developer Advocacy: голос разработчиков

Часть 4. Мероприятия

10. Типы DevRel мероприятий
11. Организация митапов
12. Геймификация и активации

Часть 5. Инструменты и метрики

13. Инструменты DevRel
14. AI-инструменты в DevRel
15. Метрики эффективности

Часть 6. Практика и кейсы

16. Кейсы российских компаний
17. Международный опыт DevRel
18. Частые ошибки и как их избежать

Заключение

Приложения

- Приложение А. Полезные ресурсы для DevRel
 - Приложение Б. Шаблоны и чек-листы
 - Приложение В. Глоссарий терминов
-

Введение

Для кого эта книга

Вы держите в руках практическое руководство по Developer Relations и построению бренда работодателя в IT. Эта книга для вас, если вы:

- **Начинающий DevRel-специалист** — хотите понять как работает профессия и с чего начать
- **Разработчик, думающий о DevRel** — рассматриваете карьерный переход
- **HR или рекрутер в IT** — хотите строить сильный employer brand
- **Руководитель IT-компании** — думаете нужен ли вам DevRel и как это работает
- **Маркетолог в tech** — хотите понять специфику работы с developer audience

Особенный фокус: Компании с командой **200+ разработчиков** в России и СНГ.

Чем эта книга отличается

Практичность.

Это не академический труд. Это сборник практических инструментов, чек-листов, кейсов из реального опыта российского DevRel-сообщества.

Актуальность.

Материалы собраны из активных DevRel-сообществ 2024-2025 года. Это то, что работает сейчас.

Фокус на РФ/СНГ.

Учтена специфика российского рынка: Telegram вместо Discord, Habr вместо Medium, онлайн-культура встреч.

Связка DevRel + Employer Brand.

Большинство книг про DevRel фокусируются на B2D компаниях (продукты для разработчиков). Мы рассматриваем DevRel как инструмент построения бренда работодателя — более широкая и актуальная для большинства IT-компаний задача.

Как пользоваться этой книгой

Если вы новичок в DevRel:

Читайте последовательно. Главы 1-3 дадут foundation, дальше углубление в каждый аспект.

Если вы уже DevRel:

Можете читать выборочно. Каждая глава независима. Ищите главы по вашим challenges (метрики? community? геймификация?).

Если вы руководитель:

Начните с Глав 1, 4, 15 (что такое DevRel, employer brand, метрики). Это даст понимание стоит ли инвестировать.

Если вы HR:

Фокус на Главы 4-6, 11-12 (employer brand, каналы, мероприятия).

Структура книги

Часть 1: Основы DevRel (Главы 1-3)

Что такое DevRel, какие навыки нужны, как развивать карьеру.

Часть 2: Бренд работодателя (Главы 4-6)

Как строить привлекательный бренд для IT-талантов.

Часть 3: Сообщество (Главы 7-9)

Community building, контент, developer advocacy.

Часть 4: Мероприятия (Главы 10-12)

Типы событий, организация митапов, геймификация.

Часть 5: Инструменты (Главы 13-15)

Инструменты, AI, метрики эффективности.

Часть 6: Кейсы и практика (Главы 16-18)

Российские компании, международный опыт, частые ошибки.

Приложения:

Ресурсы, шаблоны, глоссарий.

Благодарности

Эта книга не появилась бы без:

- **DevRel-сообщества России** — ваши обсуждения, опыт и советы стали основой
- **Ксении Романовой** — за соавторство и экспертизу
- **Всех DevRel-практиков** — кто делится опытом в каналах и на конференциях

Особая благодарность каналу “Говорите громче!” — откуда пришло множество практических инсайтов про спикерство и DevRel-работу.

Начнем?

Developer Relations — молодая, динамичная, и невероятно интересная профессия. Вы будете общаться с умными людьми, влиять на индустрию, помогать разработчикам, и строить сообщества.

Это не легко. Но оно того стоит.

Давайте разберемся как это работает.

Рафаэль Миниахметов

Октябрь 2025

ЧАСТЬ 1. ОСНОВЫ DEVREL

Глава 1. Что такое Developer Relations

Зачем эта глава: Developer Relations — относительно молодая профессия в России и СНГ, и даже среди IT-специалистов не все понимают, чем конкретно занимается DevRel. В этой главе мы разберем суть профессии, ее историю, чем DevRel отличается от смежных ролей, и когда компании действительно нужен DevRel-специалист.

1.1. Определение: что такое DevRel простыми словами

Developer Relations (DevRel) — это профессия и функция в IT-компании, которая отвечает за **построение и поддержание отношений с разработчиками**: как внутри компании, так и во внешнем сообществе.

Простыми словами:

Если ваша компания:

- Создает инструменты/API/платформу для разработчиков
- Хочет привлекать и удерживать талантливых инженеров
- Стремится быть частью технологического сообщества
- Нуждается в technical thought leadership

...то вам нужен DevRel.

DevRel-специалист — это человек, который:

- Помогает разработчикам использовать ваш продукт (если B2D)
- Представляет компанию на конференциях и митапах
- Пишет технические статьи и делает туториалы
- Строит сообщества вокруг технологий компании
- Собирает обратную связь от разработчиков и доносит ее до продуктовой команды
- Делает компанию привлекательной для IT-талантов

Кратк◆ ◆я история Developer Relations

Ранние годы (1980-1990-е):

Первые DevRel-программы появились в компаниях, которые создавали платформы и инструменты для разработчиков: Apple, Microsoft, Sun Microsystems. Их задача была проста: помочь разработчикам создавать приложения для их платформ.

Рост (2000-2010-е):

С появлением облачных платформ (AWS, Google Cloud, Azure) и развитием API-экономики DevRel стал критичной функцией. Компании поняли: **чем больше разработчиков используют твою платформу, тем сильнее твой бизнес.**

Современность (2015-настоящее):

DevRel вышел за рамки B2D (business-to-developer) компаний. Теперь это:

- **Employer Branding** в IT (привлечение талантов)
- **Community Building** вокруг open-source проектов
- **Technical Thought Leadership** (стать экспертами в глазах индустрии)
- **Developer Advocacy** (представлять интересы разработчиков внутри компании)

В России:

DevRel как профессия начал активно развиваться с 2015-2017 года. Первыми были крупные компании: Яндекс, VK, JetBrains, Kaspersky. С 2020 года DevRel появляется и в средних компаниях (200-500 человек).

Типичные заблуждения о DevRel

Миф 1: "DevRel = Marketing для разработчиков"

Реальность:

DevRel — это не реклама. Разработчики ненавидят продающие презентации и пушливых промоутеров. DevRel создает **реальную ценность** для сообщества: образовательный контент, инструменты, помочь в решении проблем.

Разница:

- Marketing: "Купи наш продукт, он лучший!"
- DevRel: "Вот как решить твою проблему. Кстати, наш инструмент может помочь, если захочешь."

Миф 2: "DevRel = HR/Рекрутинг"

Реальность:

DevRel работает на бренд и awareness, что косвенно помогает найму. Но **прямой хантинг — не задача DevRel.**

Что делает DevRel:

- Создает visibility компании
- Показывает экспертизу команды
- Формирует позитивное восприятие

Что делает HR/ТА:

- Поиск кандидатов
- Screening и интервью
- Офферы и онбординг

Связь: DevRel создает warm leads (люди уже знают и любят компанию), HR конвертирует их в найм.

Миф 3: "DevRel нужен только B2D компаниям"

Реальность:

Изначально DevRel появился в B2D (продукты для разработчиков: AWS, Stripe, Twilio). Но сегодня DevRel нужен **любой IT-компании, которая хочет привлекать таланты**.

DevRel для B2D: Помощь external разработчикам использовать ваш API/платформу

DevRel для B2C/B2B: Построение employer brand, привлечение талантов, technical thought leadership

Миф 4: "DevRel — это только про конференции и митапы"

Реальность:

Публичные выступления — видимая часть, но это **лишь один инструмент** из многих.

Что еще делает DevRel:

- Технический контент (статьи, туториалы, видео)
- Open Source программы
- Внутренние сообщества (knowledge sharing)
- Developer Advocacy (донос feedback от разработчиков до продукта)
- Community Management (модерация, вовлечение)
- Аналитика и метрики
- Стратегия и планирование

1.2. DevRel vs смежные роли: в чем разница?

DevRel часто путают со смежными ролями. Разберем ключевые отличия.

DevRel vs Developer Marketing

Критерий	DevRel	Developer Marketing
Цель	Построение отношений и trust	Продвижение продукта и конверсия
Метрики	Engagement, retention, sentiment	Leads, conversions, revenue

Критерий	DevRel	Developer Marketing
Контент	Образовательный, honest	Продуктовый, selling
Подход	"Помочь разработчику"	"Продать продукт разработчику"
Горизонт	Долгосрочный (годы)	Краткосрочный (кварталы)

Пример:

Developer Marketing:

"Наш API в 10x быстрее конкурентов! Зарегистрируйся сейчас и получи \$100 кредитов!"

DevRel:

"Как оптимизировать работу с API: 5 best practices. Вот примеры кода и бенчмарки."

(В статье упоминается ваш API, но фокус на решении проблемы)

Важно:

Обе функции нужны! DevRel создает trust и awareness, Marketing конвертирует в sales. Они работают вместе, но с разными KPI.

DevRel vs Community Manager

Критерий	DevRel	Community Manager
Фокус	Технологии + сообщество	Сообщество + коммуникации
Контент	Технический (код, архитектура)	Любой (анонсы, ивенты, general)
Аудитория	Преимущественно разработчики	Широкая (пользователи, фанаты бренда)
Навыки	Технические + коммуникативные	Коммуникативные + организаторские

Пример:

Community Manager (для игры):

- Модерирует Discord сервер игроков
- Анонсирует обновления
- Собирает баг-репорты
- Организует турниры

DevRel (для геймdev платформы):

- Пишет туториалы "Как создать игру на нашем движке"
- Выступает на геймdev конференциях

- Помогает разработчикам игр решать технические проблемы
- Собирает feedback от геймдев-сообщества для улучшения движка

В IT-компаниях:

Часто DevRel = Community Manager + Developer Advocate в одном лице (особенно в небольших командах).

DevRel vs Developer Advocate

Developer Advocate — это одна из ролей внутри DevRel.

DevRel (широкая функция):

- Developer Advocates (публичное лицо)
- Community Managers (работа с сообществом)
- Technical Writers (документация, туториалы)
- Developer Experience (улучшение DX)
- Program Managers (стратегия и координация)

Developer Advocate (конкретная роль):

- Публичные выступления
- Технический контент
- Представление продукта в сообществе
- Сбор feedback от разработчиков

Аналогия:

DevRel — это отдел, Developer Advocate — должность в этом отделе.

DevRel vs Technical Evangelist

Исторически:

Technical Evangelist — старое название (из эпохи Microsoft, Apple).

Сейчас чаще используют Developer Advocate.

Тонкое отличие:

- **Evangelist** — “проповедник”, продвигает технологию (иногда агрессивно)
- **Advocate** — “адвокат”, защищает интересы разработчиков

Современный тренд:

Переход от “евангелизации” к “адвокатированию” — от одностороннего продвижения к двустороннему диалогу.

1.3. Роль DevRel в структуре IT-компании

Где находится DevRel в оргструктуре? Кому подчиняется? С кем работает?

Варианты размещения DevRel

Вариант 1: Под СТО (Engineering)

Плюсы:

- Близость к технологиям и инженерам
- Легче найти амбассадоров и спикеров
- Проще понимать технический контекст
- Доверие от инженерной команды

Минусы:

- Может быть оторван от маркетинговой стратегии
- СТО может не понимать ценность DevRel
- Конфликт приоритетов (продуктовая разработка vs бренд)

Когда подходит:

Компании с B2D продуктом (API, SDK, платформа) или с фокусом на employer branding.

Вариант 2: Под СМО (Marketing)

Плюсы:

- Согласованность с общей маркетинговой стратегией
- Доступ к маркетинговому бюджету
- Синергия с PR, SMM, Content Marketing
- Четкие метрики и KPI

Минусы:

- Инженеры могут не доверять ("это маркетологи, не свои")
- Давление на "продажи" вместо education
- Может не быть технической экспертизы

Когда подходит:

B2D компании с сильным маркетингом, где DevRel = канал привлечения пользователей.

Вариант 3: Под СРО (Product)

Плюсы:

- Прямая связь с продуктом
- DevRel feedback сразу влияет на roadmap
- Фокус на Developer Experience
- Метрики: adoption, retention, satisfaction

Минусы:

- Может быть узкий фокус только на продукт
- Меньше внимания employer branding

Когда подходит:

B2D SaaS/PaaS компании (Stripe, Twilio модель).

Вариант 4: Под CEO (отдельная функция)

Плюсы:

- Высокий приоритет в компании
- Доступ ко всем ресурсам
- Стратегическая роль
- Независимость от отделов

Минусы:

- Высокие ожидания
- Политика и конфликты с другими отделами

Когда подходит:

Крупные компании (1000+ человек), где DevRel = стратегический приоритет.

Практическая рекомендация

Для компаний 200-500 человек:

Начните с DevRel под СТО, если фокус на employer branding.

Под СМО, если фокус на продуктовый DevRel (B2D).

Для компаний 500-1000 человек:

Отдельная DevRel команда с прямым репортом СТО или CEO.

Для компаний 1000+ человек:

Полноценный DevRel отдел с подфункциями:

- Developer Advocacy
- Community Management
- Technical Content

- Developer Experience
- Events & Programs

С кем взаимодействует DevRel

DevRel — это **кроссфункциональная** роль. Вы работаете со всеми:

С Engineering:

- Ищете амбассадоров и спикеров
- Собираете технический контент
- Доносите feedback от сообщества

С HR/ТА:

- Помогаете в employer branding
- Создаете warm pipeline кандидатов
- Организуете внутренние сообщества

С Marketing:

- Координируете messaging
- Создаете технический контент
- Участвуете в кампаниях

С Product:

- Собираете feedback от разработчиков
- Представляете интересы пользователей
- Помогаете с adoption новых фич

С Leadership (СТО/CEO):

- Репортите метрики
- Выстраиваете стратегию
- Получаете бюджет и поддержку

Важно:

Основной стейкholder работы DevRel **чаще всего СТО**, даже если формально вы подчиняетесь кому-то другому. При организационных изменениях или сомнениях в ценности DevRel — идите к СТО, обсуждайте приоритеты и выравнивайте ожидания.

1.4. Когда компании нужен DevRel?

Не каждой компании и не на каждой стадии нужен DevRel. Разберем, когда это действительно имеет смысл.

Критерии “Нужен DevRel”

1. Размер инженерной команды

< 50 разработчиков:

Фуллтайм DevRel **скорее всего не нужен**.

Почему:

- Контента на 10 докладов и 15 статей в год хватит на 6-12 месяцев
- Потом нужно год кодить новые проекты, чтобы было о чем рассказать
- Запал команды закончится быстро
- Фуллтайм DevRel будет простоявать

Альтернатива:

- Внешний консультант (разовые проекты)
- Агентство под конкретные задачи
- Кто-то из команды part-time (20-30%)

50-200 разработчиков:

Можно начинать строить DevRel, но **осторожно**.

Подход:

- Начните с 0.5-1 FTE DevRel
- Фокус на 2-3 инструмента максимум
- Проверьте гипотезу: работает ли это для вашей компании?

200+ разработчиков:

Вам **нужен DevRel**. Вопрос не “нужен ли?”, а “какого масштаба?”.

Почему:

- Достаточно экспертов для контента ($10\% \text{ от } 200 = 20$ человек)
- Можно поддерживать постоянный поток активностей
- Компания уже заметна — нужно управлять репутацией
- Конкуренция за таланты высокая

Масштаб DevRel для 200+:

- 1-2 FTE DevRel на 200-500 разработчиков

- 3-5 FTE DevRel на 500-1000 разработчиков
- 5-10+ FTE DevRel на 1000+ разработчиков (п полноценный отдел)

2. Тип продукта/бизнеса

DevRel критичен для:

- **B2D компании** (API, SDK, платформы, инструменты для разработчиков)
- **Open Source проекты** (нужно community)
- **Технологические лидеры** (хотят быть thought leaders)
- **Быстрорастущие компании** (нужно много нанимать)

DevRel полезен для:

- **B2B SaaS** (если есть API/интеграции)
- **B2C с сильным tech брендом** (Яндекс, VK)
- **Консалтинговые компании** (нужна экспертиза)

DevRel скорее всего не нужен:

- Небольшие аутсорс компании (< 50 человек)
- Продуктовые компании без технического фокуса
- Компании, где tech — не конкурентное преимущество

3. Стадия компании

Стартап (0-50 человек):

DevRel **обычно рано**. Фокусируйтесь на product-market fit.

Исключение:

Если ваш продукт = инструмент для разработчиков (B2D), тогда DevRel нужен с первых дней (пример: Vercel, Supabase).

Scale-up (50-500 человек):

Самое время начинать DevRel, особенно если:

- Активно нанимаете (10+ открытых вакансий)
- Конкурируете за таланты с крупняком
- Есть что рассказать (интересные проекты, технологии)

Enterprise (500+ человек):

DevRel **обязателен** для поддержания репутации и найма.

4. Стратегические цели

DevRel нужен, если:

- 🎯 Хотите быть technical thought leaders в индустрии
- 🎯 Страйте платформу и нужны external разработчики
- 🎯 Активно нанимаете и конкурируете за таланты
- 🎯 Развиваете Open Source и нужно community
- 🎯 Выходите на новые рынки и нужна visibility

DevRel не приоритет, если:

- ✗ Основная проблема — retention (сначала фиксите HR)
- ✗ Нет четкой стратегии компании (DevRel не поможет)
- ✗ Нет поддержки от руководства (будет блокировка)

Самопроверка: нужен ли нам DevRel?

Ответьте на вопросы:

1. У нас в компании 200+ разработчиков? **Да / Нет**
2. Мы активно нанимаем (10+ вакансий)? **Да / Нет**
3. У нас есть что рассказать (интересные проекты, технологии)? **Да / Нет**
4. Мы хотим быть известными в tech-сообществе? **Да / Нет**
5. HR функция работает реально (не номинально)? **Да / Нет**
6. Руководство поддерживает идею DevRel? **Да / Нет**
7. Есть бюджет на это (минимум 2-3 млн в год)? **Да / Нет**

Результат:

- 6-7 “Да” → DevRel вам нужен, начинайте!
- 4-5 “Да” → Можно начинать, но осторожно (пилот на квартал)
- < 4 “Да” → Рано. Сначала решите другие проблемы.

1.5. Что конкретно делает DevRel: обзор активностей

Теперь конкретнее: **чем DevRel занимается каждый день?**

Основные направления деятельности

1. Контент (40-50% времени)

Что:

- Технические статьи и туториалы

- Видео и демо
- Документация
- Кейсы и best practices
- Подкасты и интервью

Подготовка контента:

- Работа со спикерами и авторами (помощь в подготовке)
- Редактура и вычитка
- Публикация и дистрибуция
- Анализ метрик (что зашло, что нет)

Среднее время:

- Статья (с автором): 20 часов эксперта + 6 часов DevRel
- Доклад (с начинающим спикером): 40 часов эксперта + 8 часов DevRel
- Видео-тutorиал: 10-15 часов

2. События (20-30% времени)

Что:

- Организация митапов (собственных)
- Участие в конференциях (спикерство, стенды)
- Воркшопы и хакатоны
- Внутренние tech talks

Задачи:

- Поиск площадки и согласование деталей
- Поиск и подготовка спикеров
- Продвижение события
- Проведение
- Follow-up и анализ

Среднее время:

- Митап (50 человек): 40-60 часов DevRel + помощь команды
- Участие в конференции со стендом: 80-120 часов подготовки

3. Community Management (15-20% времени)

Что:

- Модерация Telegram чатов / Discord
- Ответы на вопросы сообщества
- Вовлечение участников
- Работа с амбассадорами и активистами
- Разрешение конфликтов

Реальность:

Построение крепкого сообщества — это **годы работы**. Много сил. Разбор конфликтов.

Правила, про которые все спорят. Желание все бросить и уйти в лес.

Совет:

Прежде чем создавать "yet another community", посмотрите: может оно уже существует?

Можно присоединиться к существующему сообществу и предложить помочь организаторам.

Так вы окажетесь в ядре крупного сообщества с готовыми охватами.

4. Developer Advocacy (10-15% времени)

Что:

- Сбор feedback от разработчиков (external или internal)
- Донос этого feedback до продуктовой/инженерной команды
- Представление интересов разработчиков
- Участие в продуктовых решениях

Пример:

Разработчики в сообществе жалуются, что ваш API неудобный. DevRel собирает конкретные кейсы, приоритизирует, доносит до команды, помогает изменить API, потом коммуницирует улучшения обратно.

5. Стратегия и аналитика (10-15% времени)

Что:

- Планирование активностей (квартал, год)
- Трекинг метрик
- Анализ эффективности
- Отчеты для руководства
- Координация с другими отделами

Ключевые выводы главы

1. **DevRel — это про отношения, не про продажи.** Фокус на создании ценности для сообщества, а не на прямой конверсии.
 2. **DevRel ≠ Marketing ≠ HR ≠ Community Manager.** Это отдельная функция с уникальными компетенциями и метриками, хотя и пересекается со смежными ролями.
 3. **Роль DevRel в компании зависит от целей:**
 - B2D компании → под Product/CTO
 - Employer branding → под CTO/HR
 - Маркетинг платформы → под СМО
 4. **DevRel нужен не всем.** Компаниям < 50 человек обычно рано. Компаниям 200+ — пора, если еще не начали.
 5. **Основа работы DevRel:**
 - Контент (40-50%)
 - События (20-30%)
 - Community (15-20%)
 - Advocacy (10-15%)
 - Стратегия (10-15%)
 6. **DevRel — это марафон.** Результаты видны через 6-12 месяцев, не сразу. Нужна поддержка руководства и терпение.
-

Чек-лист для старта DevRel в компании

Перед наймом DevRel:

- [] Инженерная команда 50+ человек (лучше 200+)
- [] HR работает реально (есть HRBP для IT)
- [] Внутренний NPS > 20 (не критичный negative)
- [] Есть что рассказать (проекты, технологии, кейсы)
- [] Руководство поддерживает и понимает ценность
- [] Есть бюджет (минимум 2-3 млн/год)
- [] Четкие ожидания от роли (какие метрики, какие цели)

После найма DevRel (первые 3 месяца):

- [] Провести аудит (чек-лист из раздела 5.1 предыдущей главы)
- [] Сформулировать “Саморепрезентацию 1+1+1”

- [] Создать EVP
- [] Найти первых 3-5 бренд-амбассадоров
- [] Составить коммуникационную стратегию на год
- [] Запустить 1 pilotную активность (митап или статья)
- [] Настроить трекинг метрик

Первый год:

- [] 10-15 докладов на конференциях
 - [] 15-20 статей в блог
 - [] 2-4 митапа собственных
 - [] Telegram канал/чат с активностью
 - [] Команда из 10-15 амбассадоров
 - [] Измеримое улучшение метрик (awareness, eNPS, hiring)
-

Что дальше?

В следующей главе мы детально разберем **компетенции DevRel-специалиста**: какие навыки нужны, как их развивать, и какой путь проходит человек от начинающего до опытного DevRel.

А пока — если вы:

Компания, думающая о DevRel:

1. Пройдите чек-лист “Нужен ли DevRel”
2. Если да — начните с аудита текущего состояния
3. Определите: какой тип DevRel вам нужен (product vs employer brand)
4. Составьте job description с четкими ожиданиями

Человек, рассматривающий карьеру в DevRel:

1. Поймите разницу между DevRel и смежными ролями
2. Определите: какое направление вам интереснее (advocacy, community, content?)
3. Прочтайте Главу 2 про компетенции
4. Начните с малого: напишите статью, сделайте доклад

Текущий DevRel:

1. Убедитесь, что вы и руководство aligned по ожиданиям
2. Проверьте: решаете ли вы задачи правильных стейкхолдеров?

3. Трекаете ли ключевые метрики?
4. Есть ли у вас стратегия или вы в режиме "тушения пожаров"?

Помните: DevRel — это не волшебная палочка. Это системная работа по построению репутации и отношений. Но при правильном подходе эффект огромен.

Статус главы: Черновик v1.0

Объем: ~2,400 слов (~10 страниц)

Следующая редакция: Добавить больше примеров из российских компаний, статистику по DevRel в РФ, ссылки на исследования DevRel-сообщества

Глава 2. Компетенции DevRel-специалиста

Зачем эта глава: DevRel — уникальная профессия на стыке технологий, коммуникаций и маркетинга. Какие навыки действительно нужны? Что критично, а что nice-to-have? Можно ли стать DevRel без опыта разработки? В этой главе — детальный разбор компетенций с практическими советами по развитию.

2.1. Технические навыки: нужно ли уметь кодить?

Один из частых вопросов: **“Обязательно ли DevRel должен быть разработчиком?”**

Короткий ответ:

Зависит от типа DevRel.

Для B2D компаний (API, SDK, платформы):

Да, технический бэкграунд **критичен**. Вы должны понимать код, архитектуру, уметь читать и писать примеры кода, разбираться в технических проблемах разработчиков.

Для Employer Branding DevRel:

Технический бэкграунд **желателен**, но не обязателен. Важнее понимать как работают разработчики, их боли, мотивацию, культуру.

Уровни технической экспертизы

Уровень 1: Технический энтузиаст (минимум для EB DevRel)

Что нужно:

- Понимать основы программирования (хотя бы 1 язык)
- Читать код (не обязательно писать с нуля)
- Понимать терминологию (API, микросервисы, CI/CD, etc.)
- Разбираться в типовых стеках (frontend, backend, devops)

Как прокачать:

- Пройти курс “CS50” или аналог (базы программирования)
- Научиться читать код на 1-2 популярных языках (Python, JavaScript)
- Следить за tech news (Habr, Hacker News)
- Общаться с разработчиками, задавать вопросы

Достаточно для:

- Организации митапов и конференций
- Работы с амбассадорами-спикерами
- Понимания о чем говорят инженеры
- Редактуры технических текстов (с помощью экспертов)

Уровень 2: Junior/Middle Developer (для большинства DevRel)

Что нужно:

- Реальный опыт разработки (1-3 года)
- Умение писать работающий код
- Понимание архитектуры и best practices
- Опыт работы с Git, CI/CD, основными инструментами

Как прокачать:

- Поработать разработчиком 1-2 года
- Сделать несколько pet-projects
- Поконтрибьютиль в Open Source
- Решать задачи на LeetCode/HackerRank (для понимания, не обязательно мастерство)

Достаточно для:

- Написания технических туториалов
- Демонстрации продукта/API
- Помощь разработчикам в troubleshooting
- Представления технических решений публично

Уровень 3: Senior Developer (для technical Developer Advocate)

Что нужно:

- Опыт разработки 5+ лет
- Глубокая экспертиза в 1-2 доменах
- Понимание архитектуры распределенных систем
- Опыт решения сложных технических задач

Как прокачать:

- Стать Senior разработчиком
- Поработать в разных компаниях и проектах

- Глубоко изучить 1-2 технологии
- Написать/поддерживать open-source проект

Достаточно для:

- Выступлений на крупных конференциях
- Написания глубоких технических статей
- Technical thought leadership
- Консультирования разработчиков по сложным вопросам

Важный нюанс: “DevRel без coding”

Существует дискуссия: **можно ли делать DevRel, не являясь разработчиком?**

Практика показывает:

Можно, но с ограничениями:

- Community Management (модерация, организация)
- Event Management (митапы, конференции)
- Content Coordination (работа с авторами, но не написание самому)
- Program Management (стратегия, координация DevRel программ)

Сложно или невозможно:

- Developer Advocacy (нужна technical credibility)
- Technical Content Creation (тutorиалы, примеры кода)
- Technical Troubleshooting
- Deep technical discussions с разработчиками

Вывод:

В DevRel есть роли для людей без coding skills, но для полноценной карьеры технический бэкграунд сильно расширяет возможности.

2.2. Коммуникативные навыки: сердце DevRel

Если технические навыки — это foundation, то **коммуникативные навыки — это то, что делает вас DevRel**, а не просто разработчиком.

Публичные выступления (Public Speaking)

Почему критично:

Выступления на конференциях — один из главных инструментов DevRel. Это visibility,

credibility, networking в одном.

Что нужно уметь:

Базовый уровень:

- Говорить перед аудиторией без парализующего страха
- Структурировать доклад логично
- Отвечать на вопросы
- Уложиться в тайминг (обычно 30-45 минут)

Как прокачать:

- Начните с internal tech talks (5-10 человек)
- Перейдите на локальные митапы (20-50 человек)
- Потом — региональные конференции (100-300)
- И затем — крупные (500-2000+)

Практические советы из DevRel-сообщества:

Подготовка доклада:

- **Начинающий спикер:** 40 часов (с деврел-поддержкой)
 - 8 встреч по 1 часу (идея, тезисы, структура, слайды, прогоны)
 - 3 месяца оптимальный срок
- **Опытный спикер:** 20-30 часов, 6 встреч
- **Профессионал:** 5-6 часов, только финальная репетиция

Полезные фишки:

- **Репетиция перед своими:** за 1-2 недели до события сделайте прогон перед командой
- **Знай свой кликер:** Некоторые кнопки гасят экран!
- **Уточните развертку проектора:** Чаще всего 16:9, а презентации в Keynote по умолчанию 4:3
- **Походите по сцене заранее:** Возьмите микрофон, послушайте себя, почувствуйте пространство
- **Подчеркивайте глаза:** Помогает зрительный контакт с аудиторией
- **Читать со слайда можно** — если интонацией вы привносите дополнительный смысл

Чем могут помочь деньги:

- Коуч по публичным выступлениям: 2-5К/час

- Тренинг-день по публичным выступлениям: 100-150К
- Дизайн слайдов: 20К (базовый) – 200К (премиум)

Продвинутый уровень:

- Удерживать внимание большой аудитории (300+)
- Импровизировать и отвечать на сложные вопросы
- Адаптировать доклад под аудиторию на ходу
- Справляться с техническими проблемами (не работает демо, сбой презентации)

Реальность:

Страх публичных выступлений — нормален. Даже у опытных спикеров бывает волнение. Но это навык, который тренируется.

Техническое письмо (Technical Writing)

Почему важно:

Статьи — долгоиграющий контент. Одна хорошая статья на Habr может давать трафик годами.

Что нужно уметь:

Базовый уровень:

- Структурировать мысли логично
- Писать понятно и без воды
- Использовать примеры кода правильно
- Редактировать и вычитывать

Временные затраты на статью:

- **Максимум (если сам):** 20 часов работы эксперта (если пишет впервые)
- **Оптимально (с DevRel поддержкой):** 4-6 встреч с автором
 - Встреча 1: Тема и структура
 - Встреча 2-3: Черновик
 - Встреча 4: Редактура
 - Встреча 5: Финализация
 - Встреча 6: Продвижение
- **Средний срок:** 2 месяца от идеи до публикации

Как упростить жизнь авторам:

- Делать расшифровки их докладов (1-2К за час видео)

- Брать интервью и оформлять в статью
- Помогать с подбором картинок
- Редактировать существующий текст (10-20K за редактуру)
- Или заказать в контент-студии (50K за статью с нуля)

Продвинутый уровень:

- Писать глубокие технические статьи самостоятельно
- Адаптировать стиль под платформу (Habr vs Medium vs корпблог)
- SEO-оптимизация (технические статьи тоже нужно находить)
- Создавать вирусный контент

Лайфхак:

Если вы DevRel без сильных writing skills, сфокусируйтесь на **помощи авторам**, а не на написании самому. Это масштабируется лучше.

Слушание и эмпатия (Underrated!)

Почему критично:

DevRel — это **двусторонняя коммуникация**. Нужно не только транслировать месседжи компании, но и **слушать** сообщество.

Что нужно уметь:

- Активное слушание (не перебивать, уточнять, суммировать)
- Понимать боли разработчиков
- Отделять эмоции от фактов (в обратной связи)
- Видеть паттерны в разговорах ("Это не один человек жалуется, это системная проблема")

Где применяется:

- Разговоры на стенах и after-party
- Модерация сообществ
- Сбор feedback для продукта
- Работа с амбассадорами (помощь преодолеть синдром самозванца)

Практический кейс:

Плохой DevRel:

- Разработчик: "У вас API неудобный, эта функция работает странно"
- DevRel: "Нет, это вы неправильно используете. Вот документация."

Хороший DevRel:

- Разработчик: "У вас API неудобный, эта функция работает странно"
- DevRel: "Расскажи подробнее, что именно? Какой use case?" → Записал → Донес до команды → Вернулся с решением

Эмпатия:

DevRel-специалист должен понимать:

- Какие боли у разработчиков (дедлайны, legacy, невнятные требования)
 - Что их мотивирует (рост, интересные задачи, autonomy)
 - Что их бесит (продающие презентации, бюрократия, плохая документация)
-

2.3. Маркетинг и аналитика: понимать бизнес

DevRel находится на стыке технологий и бизнеса. Нужно понимать обе стороны.

Marketing Skills

Что нужно знать:

1. Понимание аудитории

- Кто ваша целевая аудитория? (junior, middle, senior?)
- Где они обитают? (Telegram, Habr, конференции, Reddit?)
- Что их волнует? (какие проблемы решают?)
- Как они потребляют контент? (статьи? видео? подкасты?)

2. Messaging и positioning

- Как формулировать key messages?
- Как позиционировать компанию/продукт?
- Tone of voice (официально? дружески? технично?)

3. Контент-маркетинг

- Контент-план (что, когда, где публиковать)
- Дистрибуция (как доставить контент до аудитории)
- SEO базовое (как сделать статью находимой)
- Promotion (как продвигать без спама)

4. SMM и комьюнити

- Telegram/Discord: как модерировать, вовлекать
- LinkedIn/Twitter: как общаться с профессиональной аудиторией
- Хэштеги, мемы, тренды

Analytics Skills

Что измерять:

Контент:

- Просмотры, дочитывания
- Лайки, комментарии, репосты
- Трафик на сайт с контента
- Конверсия (подписки, регистрации)

События:

- Количество участников
- Качество участников (релевантность)
- Feedback (NPS после митапа)
- Leads (сколько контактов, сколько в pipeline)

Сообщество:

- Рост подписчиков
- Активность (сколько пишут, как часто)
- Sentiment (позитив vs негатив)

Бизнес impact:

- Awareness (знают ли о компании?)
- Consideration (рассматривают ли как место работы?)
- Application (откликаются ли на вакансии?)
- Hire (нанимаются ли?)

Инструменты:

- Google Analytics для сайта
- Metabase/Tableau для дашбордов
- Social listening tools (Mention, Brand24)
- CRM для трекинга лидов

Что нужно уметь:

- Работать с Google Sheets / Excel
 - Создавать простые дашборды
 - Интерпретировать данные (что работает, что нет?)
 - Показывать ROI руководству
-

2.4. Soft Skills: что отличает хорошего DevRel

Технические и маркетинговые навыки можно прокачать курсами и практикой. Но есть soft skills, которые делают DevRel по-настоящему эффективным.

1. Эмпатия и emotional intelligence

Почему критично:

DevRel постоянно работает с людьми: инженеры, спикеры, амбассадоры, сообщество. Нужно чувствовать их состояние.

Примеры применения:

- Понять, что спикер не “сливается”, а просто выгорел — нужна помошь
- Почувствовать напряжение в сообществе до того как разгорелся конфликт
- Адаптировать тон в зависимости от ситуации (жесткий feedback vs поддержка)

2. Проактивность и initiative

Почему критично:

DevRel часто работает автономно. Никто не скажет “сделай 10 докладов” — вы сами планируете, инициируете, делаете.

Что нужно:

- Видеть возможности (“О, эта технология актуальна — давайте сделаем про нее доклад”)
- Не ждать указаний (“Надо обновить документацию” → берешь и делаешь)
- Предлагать новые форматы (подкаст? воркшоп? хакатон?)

3. Мультитаскинг и организованность

Реальность DevRel:

Одновременно готовите 3 спикеров к конференциям, редактируете 2 статьи, организуете митап, модерируете чат, отчитываетесь руководству.

Что помогает:

- Task management (Notion, Jira, Asana)
- Календарь (все дедлайны видны)
- Делегирование (“Делегируй или сдохнешь” — из практики)
- Приоритизация (не все срочное — важное)

Матрица Эйзенхауэра для DevRel:

	Срочно	Не срочно
Важно	Доделать слайды к завтрашнему докладу	Планирование на квартал
Не важно	Ответить на вопрос в чате	Прочитать очередную статью про DevRel

Фокус на: Важное (срочное и несрочное), делегирай/удали неважное.

4. Networking и relationship building

Почему критично:

DevRel — это про **отношения** (Relations в названии).

С кем строить отношения:

- **Внутри компании:** Инженеры, HR, маркетинг, руководство
- **В индустрии:** Другие DevRel-ы, организаторы конференций, журналисты
- **В сообществе:** Активисты, лидеры мнений, амбассадоры

Как строить:

- Быть искренним (люди чувствуют фальшь)
- Помогать без ожидания немедленной отдачи
- Регулярно поддерживать контакт (не только когда что-то нужно)
- Участвовать в профессиональных сообществах

Пример:

Вместо “Привет, давно не общались! Кстати, хочешь выступить у нас на митапе?”

Лучше: Регулярно комментировать посты, помогать советами, звать на кофе — и когда нужен спикер, человек сам предложит.

5. Адаптивность и learning agility

Почему критично:

Технологии меняются быстро. Форматы тоже. DevRel должен постоянно учиться.

Что нужно:

- Быстро разбираться в новых технологиях
- Тестировать новые форматы (Clubhouse → Spaces → Podcasts)
- Адаптировать тактики под изменения рынка
- Не бояться экспериментов

Пример:

2020: Все митапы переехали в онлайн → DevRel быстро освоили Zoom, YouTube Live, Discord.
2023: AI boom → DevRel начали изучать LLM и делать контент про AI-инструменты.

6. Стрессоустойчивость и anti-выгорание

Реальность:

DevRel — эмоционально затратная работа. Постоянно с людьми, публичность, множество задач, иногда негативный feedback.

Признаки выгорания DevRel:

- Не хочется идти на очередной митап
- Раздражают вопросы в сообществе
- Потеря креативности (делаешь на автомате)
- Ощущение "никто не ценит мою работу"

Как избежать:

Превентивно:

- Делегируй нижние 80% задач (из совета руководителя)
- Выделяй время на recovery (after conference — день отдыха)
- Разделяй ответственность с амбассадорами (не тащи все один)
- Работай в команде (даже если DevRel один, ищи support группу)

Когда уже выгораешь:

- Сходи к стейкхолдеру/СТО, обсуди нагрузку и приоритеты
- Делай то, что наполняет энергией, а не то, что "надо"
- Возьми отпуск (seriously, отдохни)
- Пересмотри список задач — что можно не делать?

Важно:

Выгоревший DevRel — это плохо для всех. Берегите себя.

2.5. Собираем все вместе: профиль идеального DevRel

Нет "одного идеального DevRel". Но есть **базовый набор** компетенций.

Минимально необходимые (Must-have):

Технические:

- Понимание программирования и tech stack
- Умение читать код
- Знание терминологии и трендов

Коммуникативные:

- Публичные выступления (хотя бы базово)
- Письмо (техническое или около-техническое)
- Служение и эмпатия

Бизнес:

- Понимание базовой аналитики
- Умение планировать и репортить

Soft:

- Проактивность
- Организованность
- Стессоустойчивость

Сильно желательные (Should-have):

-  Опыт разработки (1-3 года)
-  Опыт выступлений (5+ докладов)
-  Опыт работы с сообществами
-  English (для международных конференций)

Nice-to-have (бонусы):

-  Опыт в маркетинге или PR
-  Дизайн навыки (для слайдов, постов)
-  Видеомонтаж
-  Свой personal brand (блог, канал, подкаст)

Профили DevRel по специализации

Technical Developer Advocate

- Технические навыки: ★★★★★
- Спикерство: ★★★★
- Writing: ★★★★
- Community: ★★★★

Фокус: Deep technical content, конференции, thought leadership

Community Manager

- Технические навыки: ★★★★
- Спикерство: ★★
- Writing: ★★★★
- Community: ★★★★★

Фокус: Модерация, вовлечение, growth сообщества

DevRel Program Manager

- Технические навыки: ★★★★
- Спикерство: ★★★★
- Writing: ★★★★
- Community: ★★★★
- Strategy & PM: ★★★★★

Фокус: Стратегия, координация, метрики, масштабирование программ

Ключевые выводы главы

1. **Технические навыки нужны, но уровень зависит от роли.** B2D → нужен strong tech. EB → достаточно tech awareness.
2. **Коммуникация — core навык DevRel.** Спикерство и письмо можно и нужно прокачивать. Это тренируется.
3. **Слушание важнее говорения.** Хороший DevRel не только транслирует, но и собирает feedback.
4. **Soft skills критичны.** Эмпатия, проактивность, стрессоустойчивость — это то, что отличает хорошего DevRel от посредственного.

5. **Нет “одного идеального набора”.** В DevRel можно найти роль под свои сильные стороны: advocacy, community, content, programs.
 6. **Главное — баланс.** Не нужно быть экспертом во всем. Компенсируйте слабые стороны командой или подрядчиками.
-

Практическое задание: оцените свои компетенции

Для каждого навыка поставьте оценку 1-5:

Технические:

- [] Понимание программирования: ____ / 5
- [] Умение писать код: ____ / 5
- [] Знание tech stack: ____ / 5

Коммуникативные:

- [] Публичные выступления: ____ / 5
- [] Техническое письмо: ____ / 5
- [] Активное слушание: ____ / 5

Бизнес:

- [] Маркетинг basics: ____ / 5
- [] Аналитика и метрики: ____ / 5
- [] Стратегическое мышление: ____ / 5

Soft:

- [] Эмпатия: ____ / 5
- [] Проактивность: ____ / 5
- [] Стессоустойчивость: ____ / 5
- [] Networking: ____ / 5

Анализ:

- **Сильные стороны** (4-5): Это ваше конкурентное преимущество. Используйте!
- **Средние** (3): Нормально. Можно прокачать при необходимости.
- **Слабые** (1-2): Либо прокачивайте, либо делегируйте/компенсируйте командой.

План развития:

Выберите 2-3 навыка для прокачки в следующие 3-6 месяцев. Не пытайтесь улучшить все сразу.

Что дальше?

В следующей главе разберем **карьерный путь в DevRel**: как войти в профессию, какие роли существуют, как расти от Junior до Lead DevRel, и стоит ли переходить из разработки.

А пока — начните прокачивать навыки:

Публичные выступления:

- Предложите internal tech talk в компании (15 минут)
- Выступите на локальном митапе
- Запишите себя на видео и посмотрите (больно, но полезно)

Письмо:

- Напишите статью на Habr или Medium
- Ведите заметки/блог (даже если никто не читает)
- Редактируйте чужие тексты (глаз замыливается)

Community:

- Станьте активным участником профессионального чата
- Помогайте новичкам с вопросами
- Организуйте что-то малое (кофе-встречу, book club)

Помните: DevRel — это навыки, которые прокачиваются практикой. Не бойтесь начинать с малого.

Статус главы: Черновик v1.0

Объем: ~2,200 слов (~9 страниц)

Следующая редакция: Добавить больше практических упражнений, таблицу развития навыков, кейсы карьерных переходов

Глава 3. Карьерный путь в DevRel

Зачем эта глава: Как попасть в DevRel? Какие роли существуют? Как расти от Junior до Lead? Стоит ли переходить из разработки? На эти вопросы ответим в этой главе с примерами из российского рынка.

3.1. Как войти в профессию DevRel

Типичные пути входа

Путь 1: Из разработки (самый частый)

Профиль:

- 3-5+ лет опыта разработки
- Начали выступать на конференциях (как хобби)
- Написали несколько статей на Habr
- Активны в профессиональных сообществах

Почему переходят:

- Нравится общаться с людьми больше чем с кодом
- Хочется видеть wider impact (не только свой код)
- Интересно влиять на индустрию
- Устали от рутинного кодинга

Как переходить:

Шаг 1: Начните параллельно (не уходите сразу!)

- Делайте internal tech talks в компании
- Выступите на 2-3 митапах
- Напишите 3-5 статей на Habr
- Станьте активным в профсообществе

Шаг 2: Оцените нравится ли

- Есть ли энергия после выступлений?
- Или вы выгораете от общения с людьми?
- Готовы ли меньше кодить?

Шаг 3: Ищите возможности

- DevRel в своей компании (если есть или планируется)
- External DevRel позиции (особенно в B2D компаниях)
- Freelance DevRel консалтинг (для start)

Риски:

- Потеря технических навыков (кодите меньше)
- Зарплата может быть ниже чем у Senior+ разработчика
- Меньше позиций на рынке (DevRel вакансий < 1% от dev вакансий)

Плюсы:

- Разнообразие (каждый день разные задачи)
- Широкая сеть контактов (нетворкинг)
- Влияние на индустрию
- Публичность и узнаваемость

Путь 2: Из маркетинга/PR

Профиль:

- Опыт в tech marketing или PR
- Интерес к технологиям
- Готовность учить technical side

Подходит для:

- Community Manager ролей
- DevRel Program Manager
- Events & Marketing в DevRel

Как переходить:

- Прокачайте technical awareness (курсы, общение с разработчиками)
- Специализируйтесь на developer marketing
- Ищите роли "Developer Community Manager" (меньше требований к coding)

Путь 3: Из HR/ТА

Профиль:

- Technical recruiter с пониманием технологий
- Интерес к employer branding

- Опыт организации мероприятий

Подходит для:

- Employer Branding DevRel
- Events Manager в DevRel
- Talent Community Manager

Как переходить:

- Организуйте tech митапы для кандидатов
- Стройте talent community
- Изучите основы программирования
- Ищите роли на стыке HR и DevRel

Путь 4: Из смежных ролей

Варианты:

- Technical Writer → Developer Advocate (через контент)
- Product Manager → DevRel (через понимание продукта)
- Support Engineer → DevRel (через понимание user pain points)
- Community Manager → DevRel (через опыт с сообществами)

Требования к Junior DevRel (entry level)

Минимум:

- Понимание технологий (хотя бы базовое)
- Опыт публичных выступлений (2-5 докладов) ИЛИ опыт technical writing (5+ статей)
- Коммуникативные навыки
- Проактивность

Желательно:

- Опыт разработки (1-2 года)
- Участие в профсообществах
- Personal brand (блог, канал)

Реалистично:

Junior DevRel позиций мало. Большинство компаний нанимают сразу Middle/Senior уровень.

Совет:

Не ждите джуниор-позиции. Начните строить портфолио (выступления, статьи, community

work) и сразу претендуйте на Middle.

3.2. Типичные роли в DevRel

DevRel — это не одна роль, а семейство ролей.

1. Developer Advocate / Technical Evangelist

Обязанности:

- Публичные выступления на конференциях
- Написание технических статей и туториалов
- Демонстрация продукта/технологий
- Сбор feedback от разработчиков

Требования:

- Сильный technical background (обычно Senior dev level)
- Отличные presentation skills
- Technical writing
- Credibility в tech community

Типичная зарплата (РФ, 2025):

- Junior: 150-200К/месяц (редко)
- Middle: 250-350К/месяц
- Senior: 400-600К/месяц

2. Community Manager

Обязанности:

- Модерация сообществ (Telegram, Discord, форумы)
- Вовлечение участников
- Организация активностей в сообществе
- Growth сообщества

Требования:

- Коммуникативные навыки
- Опыт модерации
- Понимание tech audience

- Организаторские способности

Типичная зарплата (РФ):

- Junior: 100-150K/месяц
- Middle: 180-250K/месяц
- Senior: 300-400K/месяц

3. Technical Content Creator / Writer

Обязанности:

- Написание документации
- Создание туториалов и guides
- Видео-контент
- Курирование контента от команды

Требования:

- Excellent writing skills
- Technical understanding
- Умение упрощать сложное
- Опыт с developer tools

Типичная зарплата (РФ):

- Middle: 200-300K/месяц
- Senior: 350-500K/месяц

4. DevRel Program Manager / Director

Обязанности:

- Стратегия DevRel программ
- Управление бюджетом
- Координация команды
- Метрики и отчетность
- Работа со стейкхолдерами

Требования:

- Опыт в DevRel (3-5+ лет)
- Program/project management skills

- Strategic thinking
- Leadership

Типичная зарплата (РФ):

- Senior PM: 400-600К/месяц
- Director: 600-1000К/месяц

5. Developer Experience (DX) Engineer

Обязанности:

- Улучшение developer experience продукта
- Работа над SDK, CLI, документацией
- Создание example apps
- Troubleshooting для разработчиков

Требования:

- Сильный technical background
- Product sense
- Empathy для developers
- Coding skills (активное)

Типичная зарплата (РФ):

- Как у разработчиков (200-600К в зависимости от уровня)
-

3.3. Траектория развития: от Junior до Lead

Junior DevRel (0-2 года опыта)

Фокус:

- Обучение (понять как работает DevRel)
- Execution (делать под руководством)
- Building portfolio (выступления, статьи)

Типичные задачи:

- Помощь в организации митапов
- Написание простых туториалов

- Модерация сообщества
- Работа с амбассадорами (под supervision)

Метрики успеха:

- Количество выполненных задач
- Качество execution
- Learning progress

Развитие:

- Сделать 10+ докладов (начать с митапов)
- Написать 15+ статей
- Провести 3+ митапа (как со-организатор)
- Выстроить relationships в индустрии

Middle DevRel (2-4 года опыта)

Фокус:

- Самостоятельность (планируете и делаете сами)
- Impact (не только quantity, но и quality)
- Специализация (выбираете сильную сторону: content, events, community)

Типичные задачи:

- Самостоятельная организация митапов
- Управление контент-планом
- Поиск и подготовка спикеров
- Работа со стратегией (с помощью менеджера)

Метрики успеха:

- Engagement и reach
- Quality of content/events
- Growth метрики (подписчики, attendance)

Развитие:

- Выступить на крупной конференции (500+ участников)
- Написать viral статью (10K+ просмотров)
- Построить сообщество (500+ участников)

- Начать mentoring junior DevRel

Senior DevRel (4-7 лет опыта)

Фокус:

- Стратегия (планируете на год вперед)
- Масштабирование (строите программы, не делаете разовые активности)
- Leadership (ментор для команды)
- Thought leadership (известны в индустрии)

Типичные задачи:

- Создание DevRel стратегии
- Построение DevRel программ (ambassador program, community program)
- Управление бюджетом
- Hiring и развитие DevRel команды
- Работа с C-level

Метрики успеха:

- Business impact (влияние на найм, retention, product adoption)
- Team performance (если есть команда)
- Industry recognition (vas знают как эксперта)

Развитие:

- Построить DevRel с нуля в компании
- Масштабировать DevRel (от 1 до 5+ человек команды)
- Стать thought leader (известный в индустрии)
- Переход в management (Lead/Director)

Lead / Director DevRel (7+ лет)

Фокус:

- Стратегия на уровне компании
- Управление командой (5-20+ человек)
- Бюджеты (миллионы)
- Влияние на продуктовую стратегию

Типичные задачи:

- DevRel vision и roadmap на 2-3 года
- Hiring и development команды
- Работа с Board/C-suite
- Industry partnerships
- Speaking на крупнейших конференциях

Метрики успеха:

- Company-level impact (DevRel как конкурентное преимущество)
 - Team success (команда растет и достигает целей)
 - Personal brand (ваше имя = бренд)
-

3.4. Переход из разработки в DevRel: стоит ли?

Честный разговор про плюсы и минусы

Плюсы перехода:

Разнообразие

- Каждый день разные задачи (не монотонный coding)
- Выступления, написание, организация, стратегия
- Меньше рутинны

Люди и networking

- Постоянное общение с интересными людьми
- Широкая сеть контактов в индустрии
- Знакомства с экспертами и лидерами

Влияние

- Видимый impact (выступление → 500 человек узнали)
- Влияние на индустрию (thought leadership)
- Помощь другим (разработчики благодарят за контент)

Публичность

- Узнаваемость в сообществе
- Личный бренд растет органически
- Возможности для развития (приглашения, оффералы, партнерства)

Гибкость

- Часто remote-friendly
- Гибкий график (митапы вечером, но можно работать днем когда удобно)

Минусы перехода:

Меньше кодинга

- Coding skills могут атрофироваться
- Сложнее вернуться в разработку если передумаете
- Не следите за новыми технологиями так глубоко

Эмоциональная нагрузка

- Постоянное общение (для интровертов — выматывает)
- Публичность (критика в соцсетях, комментариях)
- Управление ожиданиями многих людей

Размытость результата

- Сложнее измерить impact (не как “запустил фичу в production”)
- Отложенный эффект (результаты через месяцы)
- Зависимость от многих факторов (не только от вас)

Зарплатный потолок

- Senior/Lead Dev может зарабатывать 500-800K+
- Senior DevRel обычно 400-600K
- Меньше вакансий → меньше возможностей роста зарплаты

Burnout риск

- Постоянная публичность
- Много контекстных переключений (events, content, community, strategy)
- Эмоциональное выгорание от работы с людьми

Кому подходит DevRel?

Вы скорее всего подходите если:

- Вам нравится выступать публично (заряжаетесь от аудитории)
- Любите писать и объяснять (получаете кайф от хорошей формулировки)
- Экстраверт или амбиверт (получаете энергию от общения)

- Хочется wider impact (не только код)
- Интересна стратегия, а не только execution
- Готовы к меньшему кодингу

Вы скорее всего НЕ подходите если:

- Интроверт который выгорает от общения с людьми
- Любите deep technical work (архитектура сложных систем)
- Не нравится публичность
- Хочется стабильности и четких задач
- Максимизируете зарплату (Senior dev платят больше)

Самопроверка: готовы ли вы к DevRel?

Ответьте честно:

1. Вы сделали 3+ публичных доклада? **Да / Нет**
2. Вам нравился процесс подготовки и выступления? **Да / Нет**
3. Вы написали 5+ технических статей? **Да / Нет**
4. Вам нравился процесс написания? **Да / Нет**
5. Вы активны в профессиональных сообществах? **Да / Нет**
6. Вы получаете энергию от общения с людьми (не выгораете)? **Да / Нет**
7. Вы готовы кодить меньше? **Да / Нет**
8. Вас не пугает публичность и возможная критика? **Да / Нет**

Результат:

- 7-8 “Да”: Вы готовы! Ищите DevRel позиции
 - 5-6 “Да”: Попробуйте part-time сначала (20-30% времени на DevRel в текущей компании)
 - < 5 “Да”: Возможно, DevRel не для вас. Или начните с малого (tech talks, статьи) и переоцените через полгода
-

3.5. Развитие карьеры: конкретные шаги

Для начинающих (вход в DevRel)

Шаг 1: Создайте портфолио (6-12 месяцев)

Публичные выступления:

- [] 2-3 internal tech talks в компании
- [] 3-5 докладов на митапах
- [] 1 доклад на конференции (если примут)

Письмо:

- [] 5-7 статей на Habr или Medium
- [] Ведите blog / Telegram канал (регулярно, 2-4 поста в месяц)

Community:

- [] Активное участие в 1-2 профсообществах
- [] Помощь новичкам (отвечайте на вопросы)
- [] Возможно, со-организация митапа

Шаг 2: Networking

- [] Познакомьтесь с 3-5 DevRel-ами из разных компаний
- [] Попросите coffee chat (узнайте как работает их роль)
- [] Посещайте DevRel встречи (есть DevRel сообщество в РФ!)

Шаг 3: Ищите возможности

- [] DevRel в своей компании (предложите идею руководству)
- [] External позиции (LinkedIn, Habr Карьера, DevRel чаты)
- [] Freelance (консалтинг для стартапов)

Для Middle → Senior

Фокус на impact:

- Не только quantity (20 докладов), но и quality (viral статья)
- Не только execution, но и strategy
- Не только свои активности, но и enable других (амбассадоры)

Прокачка:

- [] Выступить на международной конференции (DevRelCon, дальнее зарубежье)
- [] Запустить программу (Ambassador Program, Community Program)
- [] Построить что-то с нуля (сообщество 1000+, митап series)
- [] Стать признанным экспертом (вас приглашают, вы не питчите)

Для Senior → Lead/Director

Фокус на leadership:

- Нанять и развить команду
- Построить DevRel function с нуля
- Создать стратегию aligned с бизнесом
- Доказать ROI и value DevRel

Прокачка:

- [] Управление командой (3-5+ человек)
 - [] Бюджет management (5+ млн)
 - [] Board-level presentations
 - [] Industry speaking (keynotes на крупных конференциях)
 - [] Mentoring других DevRel-ов
-

3.6. Альтернативные пути: не только vertical growth

Не обязательно идти по вертикали Junior → Senior → Lead.

Lateral moves (горизонтальные переходы)

DevRel → Product Management

Почему логично:

- DevRel глубоко понимает пользователей (разработчиков)
- Собирает feedback
- Понимает market и конкурентов

Что нужно:

- Прокачать product sense
- Научиться prioritization frameworks
- Понимание бизнес-метрик (revenue, retention, etc.)

DevRel → Engineering Management

Почему логично:

- DevRel развивает leadership skills
- Опыт работы с людьми
- Strategic thinking

Что нужно:

- Вернуться к coding (хотя бы part-time)
- Прокачать people management
- Изучить engineering best practices глубже

DevRel → Marketing/Growth

Почему логично:

- DevRel — это marketing для developers
- Опыт с метриками, campaigns, content

Что нужно:

- Прокачать growth marketing
- Performance marketing (если не было опыта)
- Работа с broader audience (не только developers)

Специализация внутри DevRel

Не обязательно быть "full-stack DevRel". Можно специализироваться:

Content Specialist:

- Глубокая экспертиза в technical writing
- Редактура и контент-стратегия
- Меньше events и community

Events Specialist:

- Профессиональная организация митапов/конференций
- Event strategy
- Меньше контента и community management

Community Specialist:

- Построение и масштабирование сообществ
- Модерация и conflict management
- Ambassador programs

Strategy Specialist (Program Manager):

- DevRel стратегия и roadmap
- Координация программ

- Метрики и ROI
 - Меньше hands-on execution
-

Ключевые выводы главы

1. **Путей в DevRel много.** Самый частый — из разработки, но можно и из маркетинга, HR, community management.
 2. **Портфолио важнее резюме.** Докладана конференциях, статьи на Habr, active community participation — это ваше портфолио.
 3. **Junior DevRel позиций мало.** Не ждите, стройте опыт параллельно и претендуйте на Middle.
 4. **Траектория не только вертикальная.** Можно расти в глубину (специализация) или переходить в смежные области (PM, EM, Growth).
 5. **DevRel подходит не всем.** Интроверты, любители deep technical work, максимизаторы зарплаты — подумайте дважды.
 6. **Зарплаты ниже чем у Senior+ разработчиков.** Но есть другие компенсации: разнообразие, влияние, networking.
 7. **Выгорание — реальный риск.** Берегите себя, не пытайтесь делать все сразу, делегируйте.
-

Практические задания

Для тех, кто хочет войти в DevRel:

Следующие 3 месяца:

- [] Сделайте 2 доклада (internal + 1 митап)
- [] Напишите 3 статьи
- [] Станьте активным в 1 профсообществе
- [] Познакомьтесь с 2-3 DevRel-ами (coffee chat)
- [] Начните вести blog или Telegram канал

Через 6 месяцев:

- Оцените: нравится ли? Есть ли энергия? Хотите ли продолжать?
- Если да → ищите DevRel позиции или предложите в своей компании

Для текущих DevRel:

Оцените свой уровень:

- [] Junior, Middle, Senior? (по опыту и impact)
- [] Где хотите быть через год?
- [] Что нужно прокачать для следующего уровня?

План развития на год:

- [] Выберите 2-3 компетенции для прокачки
 - [] Поставьте конкретные цели (10 докладов, 1 программа, etc.)
 - [] Найдите ментора (Senior/Lead DevRel)
 - [] Начните mentoring других (если вы Middle+)
-

Что дальше?

В следующей части книги мы перейдем к **построению бренда работодателя** (Главы 4-6). Вы уже знаете что такое DevRel, какие навыки нужны, и как развиваться. Теперь применим это к конкретной задаче: сделать вашу компанию привлекательной для талантов.

А пока — если вы думаете о карьере в DevRel:

1. Начните строить портфолио (выступления, статьи)
2. Попробуйте part-time (20-30% времени в текущей роли)
3. Через полгода переоцените: нравится ли? Хотите ли full-time?

Помните: DevRel — это не для всех. И это нормально. Лучше понять это заранее через эксперименты, чем переходить и разочароваться.

Статус главы: Черновик v1.0

Объем: ~2,100 слов (~8-9 страниц)

Следующая редакция: Добавить интервью с DevRel-ами о их карьерном пути, статистику зарплат из исследования DevRel-сообщества, больше про international opportunities

ЧАСТЬ 2. БРЕНД РАБОТОДАТЕЛЯ

Глава 4. Что такое Employer Brand

Ключевой принцип: Построение бренда работодателя — это долгосрочная стратегия, требующая системного подхода и постоянных инвестиций в отношения с аудиторией. Это не разовые акции или тактические ходы, а непрерывный процесс выстраивания доверия и репутации.

4.1. Определение и составляющие Employer Brand

Employer Brand (бренд работодателя) — это репутация компании как места работы в глазах текущих и потенциальных сотрудников, а также внешней аудитории. Это то, что люди думают и говорят о вашей компании как о работодателе.

Почему это важно?

В IT-индустрии, где конкуренция за таланты достигает пика, сильный бренд работодателя становится критическим конкурентным преимуществом. Компании с узнаваемым и привлекательным брендом:

- **Получают больше откликов** на вакансии — до 50% больше по исследованиям [LinkedIn Talent Solutions](#)
- **Снижают стоимость найма** — до 43% по данным [Glassdoor Economic Research](#)
- **Удерживают сотрудников дольше** — выше лояльность и вовлеченность (eNPS коррелирует с retention)
- **Привлекают лучших кандидатов** — 75% job seekers учитывают employer brand перед откликом ([CareerArc, 2023](#))

Составляющие Employer Brand

Бренд работодателя — это не просто логотип или слоган. Это комплексное восприятие, которое складывается из:

1. EVP (Employee Value Proposition)

Ценностное предложение для сотрудников — что вы даете взамен их таланта и времени:

- Интересные проекты и технологии
- Возможности роста и развития
- Культура и атмосфера в команде
- Компенсация и бенефиты

- Work-life balance
- Миссия и ценности компании

2. Культура компании

Реальная, а не декларируемая культура:

- Как принимаются решения
- Как общаются люди между собой
- Какие ценности реально поддерживаются
- Как компания относится к ошибкам и экспериментам
- Баланс между процессами и свободой

3. Репутация на рынке

Что говорят о компании:

- Отзывы на Glassdoor, Хабр Карьере, [vc.ru](#)
- Упоминания в соцсетях и профессиональных сообществах
- Истории бывших и текущих сотрудников
- Медийность и публичность (статьи, выступления, митапы)

4. Визуальная идентичность

Как компания выглядит:

- Корпоративный стиль
- Оформление офиса (для онлайн)
- Дизайн карьерного сайта
- Контент в соцсетях
- Мерч и сувенирка

5. Опыт кандидата (Candidate Experience)

Как проходит взаимодействие с компанией:

- Скорость и качество ответа на отклик
- Процесс собеседований (уважение, прозрачность)
- Обратная связь (даже при отказе)
- Онбординг новых сотрудников

4.2. Почему это важно для IT

Война за таланты

IT-индустрия сталкивается с уникальным вызовом: **спрос на квалифицированных разработчиков значительно превышает предложение**. Особенно это касается:

- Senior и Lead разработчиков
- Специалистов по редким технологиям
- Архитекторов и системных аналитиков
- DevOps и Site Reliability Engineers

В такой ситуации **компании конкурируют не только зарплатами**, но и всем опытом работы, который они предлагают.

Скорость найма критична

Для растущих IT-компаний скорость закрытия вакансий напрямую влияет на бизнес:

- **Каждый день открытой вакансии** — это упущенная выгода
- **Медленный найм** — это проекты, которые буксуют
- **Долгий поиск** — это выгоревшая команда в перегрузке

Сильный бренд работодателя **ускоряет найм** за счет того, что кандидаты сами приходят и уже замотивированы работать именно у вас.

Удержание дороже найма

Известный факт: **удержать текущего сотрудника в 5-25 раз дешевле, чем нанять нового**.

Но в IT это усугубляется:

- **Рынок горячий** — рекрутеры постоянно охотятся за вашими людьми
- **Увольнение Senior** — это потеря знаний, проектов, связей
- **Текучка разрушает команду** и атмосферу

Сильный бренд работодателя **повышает лояльность**:

- Люди гордятся тем, где работают
- Делятся успехами компании как своими
- Защищают компанию в профессиональных дискуссиях
- Реже рассматривают предложения от конкурентов

Employer Brand = Competitive Advantage

В эпоху, когда технологии доступны всем, а бизнес-модели легко копируются, **люди становятся главным конкурентным преимуществом**.

Компании с сильным брендом работодателя:

- Собирают лучшие команды
- Быстрее выводят продукты на рынок
- Имеют более высокую продуктивность
- Создают инновации

Пример: Google, Яндекс, JetBrains — эти компании могут выбирать из лучших кандидатов не только благодаря высоким зарплатам, но и потому что работа там — это статус, история в резюме, возможность учиться у лучших.

4.3. Связь Employer Brand и DevRel

На первый взгляд может показаться, что **Employer Brand** — это задача HR-отдела, а **DevRel** — отдельная функция, связанная с разработчиками и сообществом. Но на практике эти направления **тесно переплетены и взаимно усиливают друг друга**.

DevRel как инструмент EB

Developer Relations **является одним из ключевых каналов построения бренда работодателя** в IT-компаниях. Вот как это работает:

1. Публичные выступления = Демонстрация экспертизы

Когда ваши разработчики выступают на конференциях:

- Показывают технологический уровень команды
- Демонстрируют подход к решению задач
- Создают ассоциацию: "В этой компании работают крутые эксперты"
- Привлекают внимание таких же сильных специалистов

Реальные затраты времени: Опытному спикеру требуется около 20 часов на подготовку доклада с учетом репетиций. Профессиональные спикеры с большим опытом могут сократить это время до 5-6 часов, особенно если доклад на уже знакомую тему.

2. Технические статьи = Прозрачность и культура

Публикации в корпоративном блоге или на Habr:

- Раскрывают технологический стек и архитектуру
- Показывают подход к проблемам и их решениям

- **Демонстрируют культуру знаниями и экспертизы**
- **Привлекают кандидатов**, которым интересны эти технологии

Площадки для публикаций: Корпоративные блоги на крупных платформах (например, Habr) обычно стоят 450-650 тысяч рублей в год, но дают готовую аудиторию и органический трафик. Собственный блог на сайте компании требует дополнительных инвестиций в продвижение и привлечение читателей.

3. Митапы и конференции = Личный контакт

Организация и участие в мероприятиях:

- **Показывают компанию изнутри** (спикеры = реальные сотрудники)
- **Создают доверие** через личное общение
- **Формируют нетворк** с потенциальными кандидатами
- **Собирают обратную связь** о восприятии компании

Эффективность стендов: Простой сбор контактов и раздача сувениров редко дает результат. Гораздо эффективнее, когда на стенде работают реальные инженеры компании, которые увлеченно рассказывают о технических проектах и собирают обратную связь от посетителей. Такой подход не только формирует положительный имидж, но и дает ценные данные для создания контента на месяцы вперед.

4. Open Source = Доказательство качества

Активность в Open Source сообществе:

- **Подтверждает технологический уровень** команды
- **Показывает готовность делиться** знаниями
- **Привлекает контрибьюторов**, которые могут стать сотрудниками
- **Создает долгосрочную репутацию**

ЕВ усиливает DevRel

Обратная связь тоже работает — **сильный бренд работодателя облегчает работу DevRel**:

1. Спикерам проще выступать

Когда компания известна:

- **Доклады принимают охотнее** на конференции
- **Спикерам задают больше вопросов** (интересно, как решаете задачи)
- **Растет органический охват** (люди сами делятся)

2. Контент распространяется быстрее

Статьи от известной компании:

- **Получают больше просмотров** изначально
- **Чаще попадают в топ** на Habr или Medium
- **Вызывают больше обсуждений**

3. Проще собирать амбассадоров

Сотрудники гордятся компанией:

- **Охотнее участвуют** в DevRel активностях
- **Сами предлагают темы** для докладов и статей
- **Защищают репутацию** компании публично

Синергия: 1+1=3

Когда Employer Brand и DevRel работают вместе, создается **мощный эффект усиления**:

1. **DevRel делает бренд видимым** → больше кандидатов знают о компании
2. **Сильный бренд привлекает таланты** → больше экспертов в команде
3. **Больше экспертов** → больше качественного контента от DevRel
4. **Больше контента** → еще сильнее бренд
5. И так по кругу ↗

Практический пример: Яндекс активно инвестирует в DevRel (конференции, митапы, блог, Open Source). Это создает образ компании, где работают сильные технологии. В результате:

- На вакансии Яндекса откликается больше сильных кандидатов
- Удержание выше (люди гордятся тем, что работают в Яндексе)
- Спикеры Яндекса автоматически получают больше внимания
- Цикл замыкается и усиливается

4.4. Особенности для компаний 200+ разработчиков

Компании разного размера сталкиваются с разными вызовами при построении бренда работодателя. Для **компаний с командой 200+ разработчиков** есть специфические нюансы.

Почему именно 200+?

Это порог, когда:

- **Недостаточно одного "лица компании"** — нужна команда бренд-амбассадоров

- Сложно поддерживать единую культуру — появляются подкультуры команд
- HR не справляется один — нужны выделенные функции (EB, TA, HRBP)
- Репутация критична — компания уже заметна на рынке, о ней говорят

Вызов 1: Масштабирование контента

Проблема:

Один DevRel не может обеспечить достаточный поток контента для поддержания видимости большой компании.

Реальность масштабирования: Практика показывает, что обычно не более 10% инженерной команды готовы активно участвовать в DevRel активностях, и лишь около 1% остаются постоянными бренд-амбассадорами дольше года. Это требует постоянного поиска и вовлечения новых участников.

Решение для компаний 200+:

- Выращивать пул бренд-амбассадоров (20-30 человек минимум)
- Ротация: не перегружать одних и тех же людей
- Разделить ответственность с руководителями команд
- Официально выделить рабочее время на бренд-активности

Практический чек-лист:

- ✓ Определите количество докладов от каждого отдела на квартал
- ✓ Согласуйте план с руководителем и включите в KPI отдела
- ✓ Каждый амбассадор должен быть учтен в квартальных целях
- ✓ Внесите работу бренд-амбассадора как проект в систему управления задачами
- ✓ Согласуйте с лицом и менеджером выделение времени на бренд-активности

Ключевой принцип: Бренд-активности должны рассматриваться менеджерами как равнозначные другим рабочим проектам. Только при официальном признании их важности сотрудники смогут заниматься ими в рабочее время без чувства вины или конфликта приоритетов.

Вызов 2: Согласованность сообщений

Проблема:

При 200+ разработчиках сложно контролировать, что и как говорят про компанию публично. Разные команды могут транслировать противоречивые месседжи.

Решение:

- Создать четкую “Саморепрезентацию 1+1+1” (см. ниже)

- Провести онбординг для всех потенциальных спикеров
- Регулярные синки DevRel с командой амбассадоров
- Единая база тем, месседжей, кейсов

Важно:

Не душить инициативу! Согласованность ≠ контроль. Дайте людям свободу, но обеспечьте общую канву.

Вызов 3: Внутренняя VS внешняя репутация

Проблема:

В больших компаниях часто бывает разрыв между тем, как компанию видят снаружи (классные проекты, крутые спикеры) и тем, как ее ощущают изнутри (бюрократия, выгорание).

Критичное предупреждение: Построение сильного внешнего бренда при проблемах с внутренней культурой и работой с людьми не только бесполезно, но и опасно. При попытках публично позиционировать компанию в позитивном ключе, текущие и бывшие сотрудники могут активно критиковать в комментариях и соцсетях, разрушая все усилия. Внутренняя культура должна быть выстроена прежде внешней коммуникации.

Решение:

- HR должен работать реально, а не номинально
- Нужен HRBP для IT функции раньше, чем DevRel
- Мониторить внутренний NPS и работать с проблемами
- Транспарентность: не скрывать проблемы, показывать как их решаете

Чек-лист старта DevRel в большой компании:

1. Метрики по лояльности, вовлеченности, eNPS
2. Кто и где уже выступал от имени компании?
3. Кто и где уже писал статьи?
4. Какие внешние площадки компания использует?
5. Какие внутренние сообщества существуют?
6. Выделено ли рабочее время на бренд-активности?
7. Поддерживают ли тимлиды идею тратить время на бренд?
8. Функция HR существует реально или номинально?

Вызов 4: Ресурсы и бюджеты

Реальность:

При команде 200+ разработчиков бюджет на Employer Brand может быть существенным, но

важно тратить его эффективно.

Базовый расчет DevRel активностей:

Первый год (10 докладов + 15 статей):

- 480 часов на 10 докладов
- 390 часов на 15 статей
- **Итого: 870 часов ≈ 5.4 месяца работы**

Множим на среднюю зарплату эксперта → получаем базу.

Дополнительные затраты:

- Дизайнеры для слайдов: 20К - 200К
- Коучинг спикеров: 2-5К/час
- Редактура статей: 10-20К
- Корпоративный блог на Хабр: 450-650К/год
- Стенды на конференциях: от 100К до нескольких миллионов

Распределение бюджета: Дорогостоящие стенды на крупных конференциях — это в первую очередь имиджевая инвестиция. При ограниченном бюджете (до 10 млн) стоит выбирать компактные стенды и инвестировать остаток в контент и других амбассадоров. При бюджете 30+ млн можно рассмотреть премиум-размещение, но всегда с учетом общей стратегии, а не как отдельную изолированную активность.

Вызов 5: Измерение эффективности

Проблема:

Руководство хочет видеть ROI от инвестиций в Employer Brand и DevRel.

Что измерять в компаниях 200+:

Входящие метрики:

- Количество органических откликов на вакансии
- Качество кандидатов (доля прошедших на оффер)
- Скорость закрытия вакансий
- Cost per hire (стоимость найма)

Репутационные метрики:

- Упоминания бренда в соцсетях и сообществах
- NPS внутренний и внешний

- Рейтинги на Glassdoor, Хабр Карьера
- Доля кандидатов, знающих о компании до отклика

DevRel метрики:

- Количество докладов/статьей/митапов
- Охваты и вовлеченность
- Рост подписчиков в корпоративных каналах
- Упоминания технологий компании в профсообществе

Важно:

В компаниях 200+ можно и нужно **нанять аналитика**, который будет трекать все эти метрики и показывать динамику.

Ключевые выводы главы

1. **Employer Brand — это марафон, не спринт.** Строится годами, разрушается мгновенно.
 2. **В IT бренд работодателя = конкурентное преимущество.** Война за таланты выигрывается репутацией, не только зарплатами.
 3. **DevRel и EB усиливают друг друга.** Это не две отдельные функции, а единая система построения репутации.
 4. **Для компаний 200+ нужна системность:**
 - Команда бренд-амбассадоров (не один DevRel)
 - Официальное выделение времени (в KPI и Jira)
 - Поддержка тимлидов и руководителей
 - Работающий HR (иначе все усилия впустую)
 5. **Контент — основа.** Доклады, статьи, митапы, Open Source — все это кирпичики репутации.
 6. **Главное — искренность.** Люди чувствуют фальшь. Показывайте реальных людей, реальные проекты, реальную культуру.
-

Чек-лист для самопроверки

Оцените бренд работодателя вашей компании:

Базовые вопросы:

- [] Можете ли вы за 1 предложение объяснить, кто вы и что делаете?
- [] Знаете ли вы свой employee NPS?
- [] Есть ли у вас метрики по качеству кандидатов и скорости найма?
- [] Работает ли HR реально (HRBP, регулярные one-on-ones, карьерное планирование)?

DevRel активности:

- [] Есть ли у вас бренд-амбассадоры (хотя бы 5-10 человек)?
- [] Выделено ли официально рабочее время на бренд-активности?
- [] Поддерживают ли тимлиды участие команды в DevRel?
- [] Есть ли план докладов/статей на квартал?

Внешняя видимость:

- [] Выступаете ли на конференциях регулярно (хотя бы 1 раз в квартал)?
- [] Пишете ли статьи в блог (хотя бы 1-2 в месяц)?
- [] Есть ли активность в Open Source?
- [] Участвуете ли в профессиональных сообществах?

Внутренняя культура:

- [] Гордятся ли сотрудники тем, где работают?
- [] Рекомендуют ли они компанию друзьям?
- [] Остаются ли ключевые люди в компании долго (2+ года)?
- [] Есть ли внутренние сообщества и knowledge sharing?

Если менее 50% ответов “да” — есть над чем работать.

Если более 80% — вы на правильном пути!

Что дальше?

В следующей главе мы разберем **как конкретно построить бренд работодателя с нуля**: от аудита текущего состояния до создания EVP и коммуникационной стратегии.

А пока — начните с малого:

1. Сформулируйте “Саморепрезентацию 1+1+1” (кто мы, что делаем, для кого, как)
2. Соберите чек-лист из раздела 4.4 про текущее состояние
3. Найдите первых потенциальных бренд-амбассадоров в команде
4. Договоритесь с одним тимлидом о пилоте: 1 доклад в квартал от команды

Помните: Великий бренд работодателя начинается с маленьких шагов. Главное — начать.

Статус главы: Черновик v1.0

Объем: ~1,800 слов

Следующая редакция: Добавить больше примеров российских компаний, схемы, чек-листы в приложение

Глава 5. Построение бренда работодателя

Ключевая идея: Построение бренда работодателя — это не абстрактная концепция из презентаций, а конкретный набор практических шагов. В этой главе вы получите пошаговый план: от аудита текущего состояния до создания коммуникационной стратегии.

5.1. Аудит текущего состояния: с чего начать

Прежде чем строить что-то новое, важно понять **где вы сейчас находитесь**. Employer Brand Audit — это не формальность, а критичный этап, который определяет всю дальнейшую стратегию.

Чек-лист DevRel для аудита компании

Каждая компания уникальна, но есть минимальный набор вопросов, с которого стоит начать:

1. Базовые метрики

Внутренние:

- [] Какой у вас employee NPS?
- [] Уровень вовлеченности сотрудников (engagement score)
- [] Соотношение критиков и промоутеров (по методологии NPS)
- [] Основные проблемные параметры компании (что люди называют)
- [] Основные привлекательные параметры (почему остаются)

Внешние:

- [] Рейтинги на Glassdoor, Хабр Карьера, [vc.ru](#)
- [] Количество органических откликов на вакансии (без рекламы)
- [] Время закрытия вакансий (в днях)
- [] Cost per hire (средняя стоимость найма одного сотрудника)
- [] Offer acceptance rate (сколько принимают офферы)

2. Текущая активность

Публичные выступления:

- [] Кто и где уже выступал от имени компании за последний год?
- [] Есть ли записи видео? Делали ли расшифровки?
- [] Планируются ли выступления на будущее?

- [] Какой feedback получили (от организаторов, аудитории)?

Контент:

- [] Кто и где писал статьи от имени компании?
- [] Есть ли корпоративный блог? Регулярность публикаций?
- [] Какие темы поднимались? Какой охват?
- [] Есть ли контент-план на следующий квартал?

Коммуникационные каналы:

- [] Какие внешние площадки компания уже использует?
 - Telegram канал/чат
 - VK, LinkedIn, Facebook
 - YouTube
 - Подкасты
 - Другое
- [] Кто за них отвечает?
- [] Какая регулярность публикаций?
- [] Сколько подписчиков? Какая динамика?

3. Внутренние сообщества

- [] Какие внутренние сообщества существуют?
 - По технологиям (Frontend, Backend, ML, etc.)
 - По интересам (книжный клуб, спорт, игры)
 - Профессиональные (code review club, архитектурный комитет)
- [] Кто лидеры этих сообществ? Кто ядро?
- [] Что делают? Как часто встречаются?
- [] Какие есть площадки для внутреннего knowledge sharing?

4. Инфраструктура развития

- [] Есть ли курсы по публичным выступлениям?
- [] Есть ли курсы по техническому письму/ведению блогов?
- [] Есть ли менторские программы?
- [] Выделено ли рабочее время инженеров на бренд-активности?

- [] Поддерживают ли руководители отделов и тимлиды идею тратить время на бренд-активности?

5. HR-функция (критичная проверка!)

- [] Функция HR существует реально или номинально?
- [] Есть ли HRBP (HR Business Partner) для IT-функции?
- [] Проводятся ли регулярные one-on-ones?
- [] Есть ли система карьерного планирования?
- [] Как быстро решаются проблемы сотрудников?

Критичное правило:

Если HR работает номинально, начинать DevRel и строить внешний бренд преждевременно и опасно. DevRel-специалисту придется тратить половину времени на HR-задачи, а при попытках публично позиционировать компанию могут возникнуть негативные отзывы от текущих и бывших сотрудников.

Принцип: Удерживать текущих сотрудников выгоднее, чем привлекать новых. Поэтому **HRBP для IT нужен раньше, чем DevRel**.

Как проводить аудит

Шаг 1: Соберите количественные данные (1-2 дня)

- Запросите метрики у HR и рекрутинга
- Проверьте рейтинги на внешних платформах
- Посмотрите статистику по текущим каналам

Шаг 2: Проведите качественные интервью (1 неделя)

- 5-10 сотрудников (разные отделы, разный стаж)
- 3-5 недавно пришедших (свежий взгляд)
- 2-3 руководителя (их видение)
- Можно анонимно через форму или лично

Вопросы для интервью:

- Почему вы выбрали эту компанию?
- Что вам нравится больше всего?
- Что хотелось бы изменить?
- Рекомендовали бы компанию другу-разработчику?
- Как бы описали компанию одним предложением?

Шаг 3: Сравните восприятие (1 день)

- Внутреннее vs внешнее
- Желаемое vs реальное
- Сильные стороны vs проблемные зоны

Результат аудита:

Документ на 5-10 страниц с:

- Текущими метриками (baseline)
 - Картой восприятия (как видят компанию)
 - Gaps анализом (что не так, что улучшить)
 - Приоритетами для работы
-

5.2. "Саморепрезентация 1+1+1": кто мы и что делаем

Один из самых мощных инструментов для построения бренда работодателя — это четкий, проработанный ответ на вопрос: **"Кто мы такие и что мы делаем?"**

Проблема

Практика показывает, что большинство компаний не имеют готового, актуального и проработанного ответа на этот базовый вопрос. Обычно бывает три сценария:

Сценарий 1: Ответа нет de-facto

Есть понимание в головах команды, но не сформулировано. Когда спрашиваешь, команда начинает формулировать на ходу, и у каждого получается по-разному.

Сценарий 2: Ответ неполный или неудачный

Есть какая-то формулировка, но сама команда ей недовольна. Не хватает точности, емкости, или она не отражает реальность.

Сценарий 3: Ответ устарел

Когда-то (возможно годы назад) сформулировали, но компания изменилась, а формулировка осталась прежней. Никто ей реально не пользуется.

Ценность четкого ответа

Хорошая саморепрезентация компании нужна для:

Внешних коммуникаций:

- Посты в соцсетях
- Презентации на конференциях

- Брифы для журналистов
- Пресс-релизы
- Карьерный сайт
- Вакансии и job descriptions

Внутренней ясности:

- Команда отчетливо осознает себя и свою работу
- Единое понимание миссии и ценностей
- Проще онбординг новых сотрудников
- Проще принимать стратегические решения ("Это в духе нашей компании?")

Методология "Саморепрезентация 1+1+1"

Это простой, но мощный framework для создания описания компании на трех уровнях детализации.

Что это:

Три версии ответа на вопрос "Кто мы и что делаем?":

1. **1 предложение** (elevator pitch)
2. **1 абзац** (краткое описание)
3. **1 страница** (полное описание)

Каждая версия отвечает на 4 вопроса:

- **Кто мы?** (тип компании, позиция на рынке)
- **Что мы делаем?** (продукт, услуга, технология)
- **Для кого?** (целевая аудитория, клиенты)
- **Как?** (подход, методология, отличие от других)

Шаг 1: Одно предложение

Формат: 1 предложение, максимум 2 строки.

Где используется:

- Представление на нетворкинге
- Начало выступления на конференции
- Twitter bio / LinkedIn tagline
- Первый экран карьерного сайта

Как писать:

Представьте, что вам нужно представиться на деловом мероприятии или выступлении TEDx. У вас есть 10 секунд внимания.

Плохой пример: "Мы IT-компания, которая делает крутые продукты."

Хороший пример (реальный): "Avito — крупнейшая платформа объявлений в России, которая помогает 50 миллионам пользователей ежемесячно покупать и продавать товары и услуги через технологии машинного обучения и высоконагруженные системы."

Формула:

[Название] – [позиция на рынке] [тип компании],
которая [основная ценность для пользователей]
через [ключевые технологии/подход].

Практическое упражнение:

Заполните шаблон для вашей компании:

- Кто мы: _____
- Что делаем: _____
- Для кого: _____
- Как: _____

Теперь объедините в одно предложение.

Критерий качества:

Если после вашего предложения человек спрашивает "А поподробнее?" — вы на правильном пути. Если "Ясно" и теряет интерес — переформулируйте.

Шаг 2: Один абзац

Формат: 4-5 предложений, примерно 100-150 слов.

Где используется:

- Раздел "О компании" в презентациях
- Пресс-релизы
- Описание для организаторов конференций
- Брифы для журналистов
- "About" в корпоративных соцсетях

Как писать:

Берем наше предложение из Шага 1 и **расширяем каждую часть** до одного предложения.

Пример расширения:

Было:

"Компания X — лидер рынка SaaS-решений для логистики в России."

Стало:

Кто мы (расширенное):

"Компания X — лидер рынка SaaS-решений для логистики в России, основанная в 2015 году командой бывших разработчиков из Яндекса и Ozon. Сегодня нашими продуктами пользуется более 5,000 компаний в РФ и СНГ."

Что делаем (расширенное):

"Мы создаем облачную платформу для управления складской логистикой, которая автоматизирует учет, маршрутизацию и аналитику, сокращая операционные расходы клиентов на 30-40%."

Для кого (расширенное):

"Наши клиенты — это средний и крупный бизнес с собственными складами: от интернет-магазинов до производственных предприятий и дистрибуторов."

Как (расширенное):

"В основе нашего решения — микросервисная архитектура, ML-алгоритмы для оптимизации маршрутов и real-time аналитика. Мы используем современный стек (Go, Python, React, Kubernetes) и активно развиваем open-source компоненты."

Итого: 4 предложения, которые дают полное понимание компании.

Шаг 3: Одна страница

Формат: 300-400 слов, примерно полторы страницы.

Где используется:

- Страница "О компании" на сайте
- Раздел "About" в презентациях для инвесторов
- Детальное описание для медиа
- Onboarding материалы для новых сотрудников
- Презентации на pitch-сессиях

Как писать:

Берем наш абзац из Шага 2 и **каждое предложение расширяем до маленького абзаца** (3-5 предложений).

Структура:

Абзац 1: Кто мы

- Позиция на рынке
- История создания (краткая)
- Ключевые цифры (размер команды, клиенты, выручка)
- Достижения и награды

Абзац 2: Что мы делаем

- Подробнее про продукт/услугу
- Ключевые фичи и преимущества
- Технологический стек (детально)
- Уникальность решения

Абзац 3: Для кого

- Детальное описание целевой аудитории
- Примеры конкретных клиентов (если можно)
- Какие проблемы решаем
- Какую ценность создаем

Абзац 4: Как

- Наш подход к разработке
- Культура и ценности команды
- Процессы и методологии
- Что делает нас особенными как работодателя

Критерий качества:

После прочтения человек должен четко понимать:

- Чем занимается компания
- Почему это важно/интересно
- Какие технологии используются
- Почему здесь может быть интересно работать

Как создавать: процесс важнее результата

Рекомендация: Не делегируйте это одному человеку (даже если это вы, DevRel). Делайте **командой**.

Почему:

- Разные люди видят компанию по-разному
- В дискуссии рождаются лучшие формулировки
- Команда начинает лучше осознавать себя
- Создается alignment (единое понимание)
- Люди сами становятся носителями месседжа

Формат воркшопа (2-3 часа):

1. Собрать команду (5-15 человек)

- DevRel, HR, маркетинг
- 2-3 руководителя (CTO, CEO)
- 3-5 инженеров (разные команды)

2. Индивидуальная работа (20 минут)

- Каждый пишет свою версию "1 предложения"
- На стикерах или в shared doc

3. Обсуждение и синтез (40 минут)

- Зачитываем все варианты
- Выбираем лучшие элементы
- Комбинируем в финальную версию

4. Расширение до абзаца (30 минут)

- Работаем вместе или в группах
- Расширяем каждую часть до предложения

5. Расширение до страницы (30 минут)

- Каждая группа берет один из 4 абзацев
- Пишут детальное описание

6. Финализация (20 минут)

- Собираем вместе
- Читаем вслух
- Корректируем

После воркшопа:

Один человек (DevRel/маркетолог) причесывает текст, унифицирует стиль, отправляет на

априв команде.

Итог:

У вас есть 3 версии описания компании, которые все понимают и поддерживают.

5.3. Создание EVP (Employee Value Proposition)

После того как вы четко сформулировали **кто вы как компания**, следующий шаг — сформулировать **почему здесь стоит работать**.

Что такое EVP?

Employee Value Proposition — это ценностное предложение для сотрудников. Ответ на вопрос: **“Что я получу, если приду работать в вашу компанию?”**

Это НЕ просто список бенефитов (ДМС, печеньки, пухфики). Это комплексное ценностное предложение, которое включает:

Компоненты EVP

1. Компенсация и бенефиты

- Зарплата (конкурентная на рынке?)
- Бонусы и опционы
- ДМС (какой уровень?)
- Другие материальные бенефиты

Для IT это:

- Компенсация обучения и конференций
- Подписки на инструменты (JetBrains, GitHub, etc.)
- Мощное железо для работы
- Budget на книги и курсы

2. Интересные проекты и технологии

Вопросы для формулировки:

- Над чем работают ваши инженеры?
- Какие технологии используете? (современные? legacy?)
- Какой масштаб задач? (сколько пользователей, RPS, данных)
- Какая степень влияния? (могу ли я реально изменить продукт?)

Пример плохого EVP:

“Работаем на современных технологиях и решаем интересные задачи.”

(Слишком общо, ничего конкретного)

Пример хорошего EVP:

“Разрабатываем высоконагруженные системы, обрабатывающие 100K RPS. Используем Go, Kafka, Kubernetes. Каждый инженер владеет feature от идеи до production и видит impact на 10 млн пользователей.”

3. Возможности роста и развития

- Карьерные треки (что technology track, что management)
- Менторство (есть ли сеньоры, готовые учить?)
- Обучение (курсы, конференции, сертификации)
- Ротация между проектами (можно ли попробовать разное?)

Конкретизация:

- “Оплачиваем до 200K в год на обучение”
- “Каждому инженеру выделяется 1 день в неделю на self-development”
- “Менторская программа: каждый junior получает ментора”

4. Культура и атмосфера

Вопросы:

- Как принимаются решения? (демократично? top-down?)
- Какой баланс процессов и свободы?
- Как относятся к ошибкам? (“move fast and break things” или “zero tolerance”?)
- Какой work-life balance? (реальный, не декларируемый)

Примеры формулировок:

- “Инженер-driven культура: технические решения принимаются командой”
- “Поощряем эксперименты: 20% времени на pet-projects”
- “Асинхронная коммуникация: минимум встреч, максимум focus time”
- “Полностью remote-first: работай откуда удобно”

5. Миссия и impact

Это не про “изменить мир” (хотя иногда и это). Это про **ощущение значимости** своей работы.

Вопрос:

Зачем компания существует? Какую проблему решает? Почему это важно?

Примеры:

Для финтех:

"Делаем финансовые услуги доступными миллионам людей, у которых раньше не было доступа к кредитам и инвестициям."

Для edtech:

"Помогаем 2 миллионам студентов учиться эффективнее через персонализированные AI-треки обучения."

Для инфраструктурного продукта:

"Наши инструменты используют 10,000 разработчиков, ускоряя их работу и позволяя фокусироваться на бизнес-логике вместо boilerplate."

Формулировка EVP: пошаговый процесс

1. Соберите данные из аудита (что уже есть)

2. Проведите фокус-группы с сотрудниками (3-5 сессий по 5-7 человек)

Вопросы:

- Почему вы выбрали нашу компанию?
- Что вам нравится больше всего?
- Что рассказываете друзьям про работу?
- Что бы изменили?
- Почему остаетесь? (для тех, кто 2+ года)

3. Выделите паттерны

Что люди называют чаще всего? Группируйте ответы:

- Технологии: 40% упомянули "современный стек"
- Команда: 60% упомянули "сильные коллеги"
- Проекты: 50% упомянули "масштаб задач"
- Культура: 70% упомянули "свобода и доверие"

4. Сформулируйте 3-5 ключевых ценностей

На основе паттернов.

Пример:

1. **Технологическое лидерство** — используем лучшие практики и инструменты
2. **Сильная команда** — учимся у лучших инженеров индустрии

- 3. Реальный impact** — наш код влияет на жизнь миллионов пользователей
- 4. Свобода и доверие** — минимум бюрократии, максимум автономии
- 5. Рост и развитие** — инвестируем в обучение каждого сотрудника

5. Опишите каждую ценность конкретно

Не абстрактно ("мы ценим людей"), а конкретно ("что это значит в реальности").

Шаблон:

[Ценность]
Что это значит: [конкретика]
Как проявляется: [примеры]
Что вы получите: [benefit для сотрудника]

Пример:

Технологическое лидерство

Что это значит:

Мы не боимся использовать новые технологии, если они решают задачу лучше. У нас нет legacy "потому что всегда так было".

Как проявляется:

- Мигрировали с монолита на микросервисы за 2 года
- Используем Go, Rust, Kubernetes в production
- Регулярные tech talks внутри компании
- Выделенное время на изучение новых технологий

Что вы получите:

- Работа с современным стеком
- Возможность предлагать и внедрять новые решения
- Обучение от сильных инженеров
- Опыт, который ценится на рынке

6. Проверьте на честность

Критичный вопрос: Это правда или wishful thinking?

Если вы пишете "минимум бюрократии", но при этом требуется 5 апрувов для деплоя — это ложь. Сотрудники это почувствуют, и ваш EVP превратится в демотиватор.

Правило:

Лучше честно описать реальность (даже если не идеальная), чем создать false expectations.

Честный EVP:

"У нас есть legacy код (50% кодовой базы на Java 8), но мы активно мигрируем на современный стек. Каждый новый проект стартует на новых технологиях, и у вас будет возможность работать и с legacy (прокачать навык рефакторинга), и с новыми решениями."

Это звучит реалистично и привлекательно для определенной аудитории (тех, кто хочет учиться рефакторингу).

Распространенная ошибка: "Мы не rocket science"

Многие компании считают, что для сильного EVP нужны "космические проекты" или уникальные технологии.

Реальность: Не обязательно.

Google, Яндекс, Tesla — да, у них впечатляющие проекты. Но в Яндексе работают 20,000 инженеров. Сколько из них занимаются self-driving автомобилями? Малый процент.

Привлекательный бренд работодателя складывается из множества факторов, и "rocket science" — лишь один из них.

Что реально важно для инженеров (по исследованиям [Stack Overflow Developer Survey 2023](#)):

1. **Команда и коллеги** (№1 фактор)
2. **Work-life balance**
3. **Возможности обучения и роста**
4. **Интересные задачи** (не обязательно "космос")
5. **Современные технологии**
6. **Компенсация** (да, но не на первом месте!)
7. **Удаленка / гибкий график**

Вывод:

Стройте EVP из того, что у вас **есть прямо сейчас**. У каждой компании есть что предложить — просто нужно это правильно сформулировать.

5.4. Коммуникационная стратегия

После того как у вас есть четкая саморепрезентация и EVP, следующий шаг — **коммуникационная стратегия**: как, где и когда вы будете транслировать ваш месседж.

Что такое коммуникационная стратегия для DevRel

По сути, это ответ на три вопроса:

1. **Каких результатов хотим достичь?** (цели, метрики)
2. **Сколько это будет стоить?** (бюджет, ресурсы)
3. **Когда это произойдет?** (сроки, занятость людей)

Важно:

DevRel менеджера стейкхолдеры оценивают по результатам, деньгам и срокам — это главное. Какие конкретно активности (стенд, статья, митап) — это инструменты для достижения результата, не самоцель.

Как составлять: пошаговый процесс

Шаг 1: Определить цели

Бизнес-цели компании → DevRel цели

Пример:

- **Бизнес:** Нанять 50 разработчиков в этом году
- **DevRel цель:** Увеличить awareness компании среди разработчиков в 2x, получить 500+ качественных лидов
- **Бизнес:** Запустить новый продукт для разработчиков
- **DevRel цель:** Создать сообщество early adopters (100 активных пользователей), собрать feedback

SMART-форматирование:

- Specific: конкретная метрика
- Measurable: измеримая
- Achievable: достижимая
- Relevant: связана с бизнес-целью
- Time-bound: с дедлайном

Плохо:

"Улучшить бренд работодателя"

Хорошо:

"Увеличить eNPS с 20 до 40 к концу года. Попасть в топ-10 Best IT Employers рейтинга Хабра."

Шаг 2: Выбрать инструменты

Для каждой цели определите **какие активности** помогут ее достичь.

Пример:

Цель: Awareness среди разработчиков в России

Инструменты:

- 12 докладов на конференциях (1 в месяц)
- 24 статьи в корпблог на Habr (2 в месяц)
- 4 митапа собственных (1 в квартал)
- Участие в 3 крупных конференциях со стендами
- Telegram канал с регулярными постами (3 в неделю)

Цель: Качественные лиды для найма

Инструменты:

- Геймифицированные стенды (сбор контактов)
- Митапы с нетворкингом
- Open source проекты (контрибьюторы → кандидаты)
- Хакатоны

Шаг 3: Посчитать ресурсы

Для каждого инструмента определите:

Время людей:

- Сколько часов инженеров? (из расчета Главы 4)
- Сколько часов DevRel?
- Сколько часов дизайнеров/маркетологов?

Деньги:

- Площадки (блог на Habr, участие в конференциях)
- Призы и мерч
- Подрядчики (дизайн, видео, редактура)

Пример расчета (12 докладов в год):

12 докладов × 40 часов подготовки = 480 часов инженеров

12 докладов × 8 часов поддержки = 96 часов DevRel

Итого: 576 часов = 3.6 месяца (FTE)

Деньги:

- Дизайн слайдов: $12 \times 30\text{K} = 360\text{K}$
- Коучинг спикеров: $50 \text{ часов} \times 3\text{K} = 150\text{K}$
- Поездки на конференции: $12 \times 50\text{K} = 600\text{K}$

Итого: ~1.1 млн

Общая стоимость:

Время ($3.6 \text{ FTE} \times \text{средняя зарплата}$) + 1.1 млн

Вывод: Теперь вы можете обосновать бюджет.

Шаг 4: Приоритизация

Скорее всего, на все сразу не хватит ресурсов. Приоритизируйте:

Матрица Impact vs Effort:

Инструмент	Impact	Effort	Приоритет
Habr блог	Высокий	Средний	🔥 1
Митапы	Средний	Высокий	🟡 3
Telegram	Средний	Низкий	🟢 2
Стенды	Высокий	Очень высокий	🟡 4

Quick wins (низкий effort, хороший impact) — делайте первыми.

Шаг 5: Roadmap и timeline

Пример квартального плана:

Q1:

- Запуск Telegram канала (неделя 1-2)
- Первая статья на Habr (неделя 3-4)
- Подготовка 3 спикеров к конференциям (весь квартал)
- Внутренний митап для тестирования докладов (неделя 10)

Q2:

- 3 доклада на конференциях
- 6 статей в блог
- Первый внешний митап (собственный)

- Участие в 1 крупной конференции со стендом

Q3-Q4:

- Масштабирование...

Виды коммуникационных стратегий

Стратегии можно составлять по разным осям:

1. По технологическим стримам

- DevRel для Backend команды
- DevRel для Frontend команды
- DevRel для ML/Data Science

Каждая — свои конференции, свои каналы, свои месседжи.

2. По географии

- DevRel для Москвы
- DevRel для регионов
- DevRel для СНГ
- DevRel международный (если есть)

3. По бизнес-юнитам

- Если у вас несколько продуктов, каждому — свою стратегию

4. По инициативам

- “Мы хотим стать лидерами в Open Source”
- “Мы хотим собрать сообщество вокруг нашего API”

Совет: Складываете все стратегии вместе → получаете **годовой коммуникационный план**.

Шаблон коммуникационной стратегии

```
# Коммуникационная стратегия: [Название]
```

```
## Цели
- Цель 1: [SMART-формулировка]
- Цель 2: [SMART-формулировка]
```

```
## Целевая аудитория
```

- Кто: [описание]
- Где: [площадки, где они обитают]
- Что их волнует: [pain points]

Key Messages

- Месседж 1: [что хотим донести]
- Месседж 2: [что хотим донести]

Активности

Активность	Срок	Ресурсы	Ответственный
12 докладов	Год	480ч + 360К	DevRel + Team
Habr блог	Год	390ч + 650К	DevRel + Authors

Бюджет

- Люди: [FTE]
- Деньги: [сумма]
- **Итого:** [общая стоимость]

Метрики успеха

- Метрика 1: [baseline → target]
- Метрика 2: [baseline → target]

Риски

- Риск 1: [описание] → Митигация: [как снизить]

Ключевые выводы главы

- 1. Аудит — first things first.** Нельзя строить бренд, не понимая текущее состояние.
Используйте чек-лист из раздела 5.1.
- 2. “Саморепрезентация 1+1+1” — мощный инструмент.** Потратьте время на воркшоп с командой. Процесс создания ценнее самого результата.
- 3. EVP должен быть честным.** Лучше реалистичное предложение, чем красивая ложь.
Люди быстро распознают фальшь.
- 4. Каждая компания может построить сильный бренд.** Не нужен “rocket science” — нужна ясность в том, что вы предлагаете.
- 5. Коммуникационная стратегия = конкретика.** Результаты, деньги, сроки. Все остальное — детали реализации.

6. Приоритизация критична. Не пытайтесь делать все сразу. Выберите 2-3 инструмента и сделайте их хорошо.

Практическое задание

Выполните прямо сейчас (или после прочтения главы):

Задание 1: Саморепрезентация (30 минут)

Напишите версию "1 предложение" для вашей компании:

- Кто мы: _____
- Что делаем: _____
- Для кого: _____
- Как: _____

→ Объедините в 1 предложение

Задание 2: EVP (1 час)

Выпишите 3-5 ключевых ценностей вашей компании как работодателя. Для каждой:

- Что это значит конкретно
- Как проявляется в реальности
- Что получает сотрудник

Задание 3: Первая стратегия (2 часа)

Создайте коммуникационную стратегию на 1 квартал:

- 1 бизнес-цель
- 2-3 DevRel активности
- Расчет ресурсов (время + деньги)
- 3 ключевые метрики

После выполнения:

У вас будет готовый план действий на ближайшие 3 месяца!

Чек-лист главы 5

Аудит:

- [] Собраны базовые метрики (NPS, engagement, найм)
- [] Проведены интервью с сотрудниками (5-10 человек)
- [] Выявлены gaps (проблемные зоны)
- [] Определены сильные стороны

Саморепрезентация:

- [] Написана версия "1 предложение"
- [] Расширена до "1 абзац"
- [] Создана версия "1 страница"
- [] Протестирована на команде

EVP:

- [] Выделены 3-5 ключевых ценностей
- [] Каждая описана конкретно (что, как, benefit)
- [] Проверена на честность (это правда?)
- [] Согласована с руководством

Стратегия:

- [] Определены цели (SMART)
 - [] Выбраны инструменты
 - [] Посчитаны ресурсы (время + деньги)
 - [] Создан roadmap на квартал/год
 - [] Определены метрики успеха
-

Что дальше?

В следующей главе мы детально разберем **каналы продвижения employer brand**: от социальных сетей до конференций, от технических блогов до программ амбассадорства.

А пока — начните с фундамента:

1. Проведите аудит по чек-листву из 5.1
2. Соберите команду на воркшоп "Саморепрезентация 1+1+1"
3. Сформулируйте ваш EVP
4. Создайте коммуникационную стратегию на 1 квартал

Помните: Без четкого понимания **кто вы и что предлагаете**, любые коммуникационные активности будут хаотичными и неэффективными. Потратьте время на фундамент — окупится

многократно.

Статус главы: Черновик v1.0

Объем: ~2,700 слов (~11 страниц)

Следующая редакция: Добавить больше примеров EVP для разных типов IT-компаний, шаблоны документов в приложение

Глава 6. Каналы продвижения Employer Brand

Зачем эта глава: У вас есть четкий EVP и коммуникационная стратегия (Глава 5). Теперь нужно донести это до аудитории. Какие каналы работают для IT-аудитории? Как выбрать правильные? Сколько это стоит? Разбираем в этой главе.

6.1. Социальные сети для IT-аудитории

Telegram: король в России и СНГ

Почему важен:

- 80%+ русскоязычных разработчиков используют Telegram
- Профессиональные чаты и каналы живут здесь
- Низкий порог входа (у всех установлен)

Форматы:

1. Корпоративный канал

Что публиковать:

- Анонсы вакансий (но не только!)
- Технические статьи и туториалы
- Новости о проектах компании
- Behind the scenes (как работаем)
- Офферы и бонусы

Frequency: 2-4 поста в неделю

Как растить:

- Анонсировать на митапах и конференциях
- Кросс-промо с другими каналами
- Ценный контент (не только "мы наняли", но и "вот полезная статья")

Метрики:

- Подписчики (рост)
- Просмотры постов (reach)
- ERR (engagement rate)

2. Профессиональный чат

Формат:

- Обсуждения технологий
- Вопросы-ответы
- Анонсы митапов
- Нетворкинг

Модерация: Нужна! (см. Главу 7)

Риски:

- Превращение в "вакансационный спам"
- Офтоп и политика
- Токсичные участники

Как использовать для ЕВ:

- Быть активным участником (не только промо)
- Помогать с вопросами
- Организовывать встречи для участников

LinkedIn: для международной аудитории

Особенности:

- Профессиональная сеть
- Хорошо работает для senior уровня
- Англоязычный контент предпочтительнее

Форматы:

- Company page (официальные новости)
- Personal branding сотрудников (амбассадоры)
- Статьи и посты
- Вакансии

Что работает:

- Employee spotlights ("Познакомьтесь с нашим Senior Engineer")
- Technical posts от инженеров
- Behind the scenes
- Culture и ценности

Frequency: 3-5 постов в неделю

VK: уходящая платформа, но еще relevant

Реальность:

Молодые разработчики уже не на VK. Но аудитория 30-40+ еще есть.

Стоит ли инвестировать:

Если ваша ЦА 35+ лет — да. Если моложе — фокус на Telegram и LinkedIn.

Twitter/X: нишевая для РФ

Особенности:

- Англоязычная аудитория
- Tech Twitter community активно
- Хорошо для personal branding

Стоит ли:

Только если вы хотите international visibility. Для РФ рынка — не приоритет.

6.2. Технические блоги и публикации

Habr: must-have для российских IT-компаний

Почему критичен:

- Крупнейшая русскоязычная tech площадка
- Высокий trust от аудитории
- Долгоживущий контент (статьи гуглятся годами)
- Интеграция с Хабр Карьера (вакансии)

Варианты:

1. Корпоративный блог

Стоимость: 450-650К/год

Что дает:

- Брендированная страница
- Organic reach (статьи попадают в ленту)
- Ассоциация с качественным контентом
- Трафик на карьерный сайт

Требования:

- Регулярность (минимум 2-4 статьи в месяц)
- Качество (редакторы Habr модерируют)
- Не продающий контент (образовательный!)

Стоит ли:

Да, если:

- Компания 200+ человек
- Есть команда авторов (5-10 активных)
- Готовы поддерживать регулярность

Нет, если:

- Маленькая команда (< 50)
- Нет авторов
- Хотите quick win (это долгосрочная инвестиция)

2. Публикации в общей ленте (бесплатно)

Как:

- Сотрудники публикуют от своего имени
- Указывают компанию в профиле
- Ставят теги компании

Плюсы:

- Бесплатно
- Более authentic (от реальных инженеров)

Минусы:

- Нет брендированной страницы
- Сложнее контролировать регулярность

Стратегия:

Начните с бесплатного. Если есть 10+ статей в год и они заходят → переходите на корпблог.

Medium: для англоязычного контента

Особенности:

- Международная аудитория

- Paywall (часть контента платная)
- Хорошо индексируется Google

Стоит ли:

- Да, если international hiring
- Да, если хотите broader tech audience
- Нет, если только РФ рынок

Собственный блог на сайте

Плюсы:

- Полный контроль
- Свой дизайн и features
- Весь трафик на ваш сайт

Минусы:

- Нет готовой аудитории (нужно гнать трафик)
- Требует ресурсов (CMS, дизайн, поддержка)
- SEO promotion нужен

Стоит ли:

Как дополнение к Habr, не вместо. Репостите статьи с Habr на свой блог.

6.3. Конференции и митапы: offline visibility

Выступления на конференциях

Типы конференций:

Крупные (500-2000+ участников):

- Примеры: HighLoad++, Saint TeamLead Conf, Mobius
- Visibility: Очень высокая
- Сложность попадания: Высокая (конкурентный CFP)
- Стоимость: Обычно бесплатно для спикеров

Средние (200-500):

- Региональные конференции
- Нишевые по технологиям

- Легче попасть

Митапы (30-150):

- Локальные, неформальные
- Легко попасть (часто ищут спикеров)
- Хороший start для начинающих

Стратегия:

Начните с митапов → Средние конференции → Крупные.

Как выбирать площадки:

Критерии:

- Совпадают ли ценности?
- Качественная ли организация?
- Адекватная ли программа?
- Какая аудитория? (ваша ли ЦА?)
- Прозрачные ли коммуникации?

Правило:

Чем публичнее компания/спикер, тем тщательнее выбирайте площадки. Иногда лучше отказаться, если есть сомнения в качестве или репутации.

Собственные митапы

Зачем организовывать свои:

- Полный контроль (тема, формат, timing)
- Брендинг (проходит в вашем офисе / под вашим брендом)
- Прямой доступ к аудитории

Стоимость: 50-200K за митап (см. Главу 11)

Frequency:

- Раз в квартал (для начала)
- Раз в месяц (если есть ресурсы)
- Раз в 2 месяца (золотая середина)

Стенды на конференциях

Зачем:

- Visibility (тысячи людей видят бренд)
- Личный контакт с аудиторией
- Сбор лидов
- Демо продукта

Стоимость:

- Малый стенд (2x2м): 100-300К
- Средний (3x4м): 400-800К
- Большой (5x6м+): 1-3 млн+

Эффективность:

Зависит НЕ от размера стенда, а от активности:

- Пассивный большой стенд (просто логотип и флаеры) → low ROI
- Активный малый стенд (инженеры общаются, геймификация) → high ROI

См. Главу 12 для деталей про геймификацию стендов.

6.4. Амбассадоры и employee advocacy

Что такое employee advocacy?

Идея:

Вместо того чтобы компания говорила о себе, **сотрудники** говорят о компании.

Почему работает:

- Доверие выше (реальные люди vs корпоративный аккаунт)
- Органический reach (их сети контактов)
- Authentic (настоящие истории, не PR)

Типы амбассадоров

1. Спикеры и авторы

Кто:

- Инженеры, которые выступают на конференциях
- Пишут статьи в корпблог или свой blog

Как поддерживать:

- Помощь в подготовке (8 встреч для новичков, см. Главу 2)

- Дизайн слайдов (если нужно)
- Продвижение их контента
- Recognition (внутренние награды, публичные благодарности)

Проблема выгорания амбассадоров:

Ключевые люди на сложных проектах часто выгорают от подготовки в личное время (вместо сна, выходных, нормального питания).

Решение:

- Официально выделить рабочее время (внести в Jira как проект)
- Включить в KPI отдела
- Согласовать с тимлидом и менеджером
- Планировать как обычные рабочие проекты

Критичный принцип:

Только когда менеджеры считают бренд-активности равнозначными другим проектам, амбассадоры смогут заниматься ими без чувства вины и конфликта приоритетов.

2. Social media амбассадоры

Кто:

- Сотрудники с активными соц сетями
- Делятся жизнью в компании

Форматы:

- Посты про рабочие проекты
- Stories из офиса (или remote setup)
- Участие в tech дискуссиях (упоминая компанию в контексте)

Как стимулировать:

- Guidelines (что можно, что нельзя)
- Content ideas (подсказки о чем писать)
- Amplification (репостите их контент с корп аккаунтов)
- Rewards (не обязательно monetary, признание тоже ценно)

3. Referral амбассадоры

Кто:

- Сотрудники, которые активно рекомендуют компанию друзьям

Метрики:

- % найма через referrals (цель: 20-30%+)
- Quality of referrals (обычно выше чем cold applications)

Как стимулировать:

- Referral bonuses (30-100K за успешный найм)
- Упростить процесс (легкая форма для рекомендации)
- Transparency (что происходит с рекомендацией? feedback)

Employee Advocacy программы

Что это:

Структурированная программа вовлечения сотрудников в продвижение бренда.

Компоненты:

1. Guidelines и training

- Что можно публиковать, что нельзя
- Как правильно представлять компанию
- Social media best practices

2. Content library

- Готовые посты для репоста
- Картинки и видео
- Hashtags и mentions

3. Gamification

- Points за посты, репосты, комментарии
- Leaderboard активных амбассадоров
- Rewards (мерч, бонусы, recognition)

4. Measurement

- Social reach от амбассадоров
- Engagement
- Attributed hires/leads

Инструменты:

- EveryoneSocial (платформа для advocacy)
- Ambassify

- Or custom (Google Sheets + manual tracking)

Бюджет: 200-500К/год (платформа + время на management)

6.5. Выбор каналов: как приоритизировать

Матрица: Reach × Cost × Relevance

Канал	Reach	Cost	Relevance	Приоритет
Habr	Высокий	Средний	Очень высокий	🔥 1
Telegram	Средний	Низкий	Высокий	🔥 2
Конференции	Высокий	Высокий	Высокий	🟡 3
LinkedIn	Средний	Низкий	Средний	🟡 4
Митапы	Низкий	Средний	Очень высокий	🟢 5

Для российского рынка, employer branding, компания 200+

Рекомендация для разных размеров компаний

Стартап (<50 человек):

- Telegram канал (бесплатно)
- Статьи на Habr в общей ленте (бесплатно)
- Участие в митапах как спикеры (бесплатно или cheap)

Total: 50-100К/год

Средняя компания (50-200):

- Telegram канал + чат
- Habr: начать с бесплатного, перейти на корпблог когда есть volume
- 2-4 митапа в год собственных
- Участие в 3-5 конференциях

Total: 500К - 1.5 млн/год

Крупная (200+):

- Habr корпблог (обязательно)
- Telegram ecosystem (канал + чат + возможно каналы по технологиям)

- LinkedIn active presence
- Митапы регулярные (раз в месяц-два)
- Стенды на крупных конференциях (2-4 в год)
- Employee advocacy программа

Total: 2-5 млн/год

Ключевые выводы главы

1. **Telegram — король в РФ/СНГ.** Если выбираете один канал — это Telegram.
 2. **Habr — must-have для серьезного tech employer brand.** Корпблог дорогой (650К/год), но эффективный.
 3. **LinkedIn для international.** Если нанимаете только в РФ, не приоритет.
 4. Конференции дают **visibility, митапы — depth.** Сочетайте оба формата.
 5. **Employee advocacy мощнее корпоративного.** Люди доверяют людям больше чем брендам.
 6. **Выбор каналов зависит от размера и бюджета.** Стартап: бесплатные. Крупная: полный набор.
 7. **Качество > Количество каналов.** Лучше 2-3 канала хорошо, чем 10 плохо.
-

Чек-лист выбора каналов

Для вашей компании:

- [] Определите ЦА (junior/middle/senior? География?)
- [] Определите бюджет
- [] Выберите 2-3 primary каналов
- [] Определите frequency и format для каждого
- [] Назначьте ответственных
- [] Настройте метрики
- [] Тестируйте 3 месяца
- [] Анализируйте и корректируйте

Статус главы: Черновик v1.0

Объем: ~1,500 слов (~6 страниц)

Следующая редакция: Добавить больше примеров контента для каждого канала, case studies успешных каналов, больше про employee advocacy программы

ЧАСТЬ 3. СООБЩЕСТВО И КОНТЕНТ

Глава 7. Community Building: как строить сообщества разработчиков

Ключевая мысль: Построение крепкого профессионального сообщества занимает **годы**. Это много сил, разбор конфликтов, споры про правила, и периодическое желание все бросить и уйти в лес. Но эффект от сильного комьюнити огромен — это и талант-пайплайн, и органическое продвижение, и обратная связь, и technical thought leadership.

7.1. Типы сообществ разработчиков

Не все сообщества одинаковы. Понимание типов помогает выбрать правильный подход.

По организации

1. Официальные корпоративные сообщества

Примеры:

- VK Developers (сообщество вокруг API ВКонтакте)
- Яндекс.Толока сообщество
- Google Developer Groups

Характеристики:

- Организуются компанией
- Фокус на продукте/технологиях компании
- Модерация компанией
- Ресурсы компании (площадки, бюджет)

Плюсы:

- Контроль над messaging
- Прямой канал коммуникации с пользователями
- Легче организовывать события

Минусы:

- Могут восприниматься как "маркетинг"
- Сложнее привлечь если продукт не популярен
- Нужны постоянные ресурсы

2. Независимые профессиональные сообщества

Примеры:

- Moscow Python
- Frontend Crew
- DevOps чат
- Kotlin Moscow

Характеристики:

- Организуются энтузиастами
- Фокус на технологию/практику, не на компанию
- Независимая модерация
- Часто волонтерская основа

Плюсы:

- Высокое доверие участников
- Органический рост
- Разнообразие мнений и опыта

Минусы:

- Сложнее модерировать (conflict management)
- Зависимость от энтузиазма организаторов
- Нет гарантии continuity

Как компании работать с независимыми сообществами:

Правильно (принято):

- Спонсировать митапы (площадка, кейтеринг)
- Предлагать спикеров (если relevant темы)
- Помогать с организацией (не навязываясь)
- Быть active member (не только когда что-то нужно)

Неправильно (будут против):

- Спамить вакансиями в чат
- Агрессивный найм на митапах
- Пытаться контролировать agenda
- Продавать продукт ("А у нас есть решение для этого!")

Совет из практики:

Прежде чем создавать "yet another community", посмотрите: может оно уже существует? Можно присоединиться и предложить помочь организаторам. Так вы окажетесь в ядре крупного сообщества с готовыми охватами. Чистый win-win.

3. Гибридные (компания + сообщество)

Примеры:

- DevRel русскоязычное сообщество (не привязано к одной компании, но модераторы часто DevRel-ы из разных компаний)

Характеристики:

- Несколько компаний участвуют
- Общие для индустрии цели
- Распределенная модерация

Идеально для:

- Нишевых технологий (мало специалистов)
- Professional development (обмен опытом между компаниями)
- Стандартизация практик

По формату взаимодействия

Online сообщества

Платформы:

- Telegram (самое популярное в РФ/СНГ)
- Discord (для более сложной структуры)
- Slack (корпоративный стандарт)
- Reddit / Discourse (форумные)

Плюсы:

- Легко масштабируется (может быть 1000+ участников)
- Async коммуникация (люди отвечают когда удобно)
- Geographic reach (можно быть из любого города)
- Низкий порог входа

Минусы:

- Сложнее модерировать (троллинг, спам)

- Меньше "теплоты" чем offline
- Легко уйти/заглушить если надоело

Offline сообщества

Форматы:

- Регулярные митапы (раз в месяц)
- Воркшопы и хакатоны
- Неформальные встречи (beer.js, coffee&code)

Плюсы:

- Сильные связи (личное знакомство)
- Высокое вовлечение
- Запоминается лучше

Минусы:

- Ограничение по географии (только город)
- Высокие организационные затраты
- Меньше масштаб (30-100 обычно)

Гибридные

Формат:

- Online чат для ежедневного общения
- Offline встречи раз в месяц/квартал

Идеально: Сочетает преимущества обоих форматов.

7.2. Запуск сообщества с нуля

Итак, вы решили создать сообщество. Пошаговый план.

Шаг 1: Определите цель и аудиторию

Вопросы:

- **Зачем?** Какую проблему решает это сообщество?
- **Для кого?** Кто целевая аудитория?
- **Что предлагаете?** Какую ценность получат участники?
- **Чем отличаетесь?** Почему это сообщество, а не существующее?

Плохие причины создавать сообщество:

- ✗ “Все делают, и мы сделаем”
- ✗ “Нужно собрать базу контактов для найма”
- ✗ “CEO сказал сделать”

Хорошие причины:

- ✓ “Нет сообщества Go разработчиков в Екатеринбурге, а нас 200+ человек”
- ✓ “Хотим создать пространство для обмена опытом по ML в production”
- ✓ “Нужно сообщество пользователей нашего API для поддержки друг друга”

Шаг 2: Выберите платформу

Для русскоязычной аудитории:

Telegram — самый популярный выбор

Плюсы:

- Все уже используют
- Просто создать и управлять
- Встроенные функции (polls, threads в супергруппах)
- Bot API для автоматизации

Минусы:

- Сложно структурировать информацию (все в ленте)
- Поиск по истории ограничен
- Модерация требует времени

Discord — для более структурированного общения

Плюсы:

- Каналы по темам (можно разделить backend/frontend/devops)
- Голосовые каналы (для voice chat)
- Интеграции и боты
- Роли и права доступа

Минусы:

- Порог входа выше (нужно устанавливать)
- Меньше популярен в РФ чем Telegram

Совет:

Начните с Telegram чата (низкий порог входа). Если вырастет и нужна структура → добавьте Discord или Slack.

Шаг 3: Создайте ядро (Core Team)

Не запускайте сообщество один. Burnout гарантирован.

Минимальная команда:

- 2-3 модератора (админы, отвечают за правила и порядок)
- 3-5 активистов (генерят контент, вовлекают других)

Где найти:

- Коллеги из компании (кто интересуется темой)
- Знакомые из индустрии
- Активисты других сообществ

Что делает ядро:

- Генерят первые обсуждения (seeding)
- Приглашают первых 50-100 участников
- Отвечают на вопросы новичков
- Поддерживают атмосферу

Шаг 4: Напишите правила

Даже если сообщество "свободное", правила нужны.

Минимальный набор:

1. **Тема сообщества:** О чем можно, о чем нельзя
2. **Уважение:** Никакого троллинга, хейта, дискриминации
3. **Реклама и найм:** Правила промо (часто: "нельзя совсем" или "только в специальном канале")
4. **Язык:** Русский/английский/оба?
5. **Модерация:** Что будет за нарушения (warn → mute → ban)

Примеры правил:

Moscow Python:

1. Обсуждаем Python и смежные технологии
2. Уважаем друг друга (no flame wars)
3. Вакансии и реклама - только в @python_jobs
4. Помогаем новичкам
5. Модераторы могут удалить сообщение без объяснений

Важно:

Правила должны быть видны (pinned message, ссылка в описании).

Шаг 5: Первые 100 участников (самое сложное)

Холодный старт — это больно.

Стратегии:

1. Personal invites

- Пригласите лично 20-30 знакомых (кто точно активный)
- Попросите их пригласить своих знакомых
- Стартовое ядро 50 человек

2. Анонс в смежных сообществах

- Попросите админов других чатов анонсировать ваше
- Кросс-промо (вы им, они вам)

3. Seed discussions

- Ядро команды генерит обсуждения первые 2-4 недели
- Задавайте вопросы, делитесь статьями, просите мнения
- Создавайте иллюзию активности

4. Привяжите к событию

- Запустите сообщество вместе с митапом
- Анонсируйте на конференции
- “Продолжим обсуждение в чате”

Правило:

Первые 100 человек приходят медленно (1-2 месяца). После 100 рост ускоряется (сарафан работает).

Шаг 6: Регулярная активность

Что делать чтобы сообщество не умерло:

Контент (3-5 раз в неделю):

- Новости из индустрии (релизы, конференции)
- Интересные статьи и туториалы
- Вопросы для обсуждения
- Опросы (polls)

События (раз в месяц/квартал):

- Митапы
- Online встречи (Zoom/Discord voice)
- Хакатоны
- Book clubs

Вовлечение участников:

- Highlight полезные комментарии
- Благодарите за помощь
- Давайте роли активным (модератор, expert)

Важно:

Если сообщество тихое 2-3 дня — норм. Если неделя+ — нужна активация.

7.3. Вовлечение и удержание участников

Проблема: 90-9-1 правило

В любом online сообществе:

- 90% — lurkers (молча читают, не пишут)
- 9% — occasional contributors (иногда комментируют)
- 1% — active creators (генерят большую часть контента)

Это нормально и неизбежно.

Задача:

Не пытаться сделать всех active creators, а:

1. Увеличить 1% до 2-3% (в 2-3х больше активных)
2. Перевести часть 90% в 9% (хотя бы комментируют)

3. Удержать lurkers (они тоже ценность — читают, учатся, рекомендуют)

Техники вовлечения

1. Снизить порог входа

Проблема:

Новичок молчит, боится выглядеть глупо.

Решения:

- Welcome message для каждого нового участника
- “Глупых вопросов не бывает” культура
- Dedicated канал #newbie-questions
- Поощряйте первый вопрос/комментарий (даже если simple)

2. Создавать поводы для участия

Простые форматы:

- **Weekly thread:** “Что выучили на этой неделе?”
- **Friday fun:** “Поделитесь любимым code snippet”
- **Monday motivation:** “Над чем работаете?”
- **Polls:** “Какой фреймворк предпочитаете?”

Почему работает:

Низкий effort для участия. Можно ответить одним предложением.

3. Highlight и благодарности

Что делать:

- Когда кто-то дает особенно полезный ответ → поблагодарите публично
- Создайте роль “Helpful member” для активных
- Периодический “Community spotlight” (рассказ про активного участника)

Эффект:

Люди видят, что их ценят → мотивация помогать другим.

4. Gamification (легкая)

Примеры:

- Karma points (как на Stack Overflow)
- Badges за активность (10 ответов, 50 ответов, etc.)
- Leaderboard месяца (кто больше помог)

Осторожно:

Не превращайте в соревнование ради соревнования. Фокус на помощи, не на points.

5. IRL встречи

Мощный инструмент:

Организуйте онлайн митап раз в квартал/полгода.

Эффект:

- Люди знакомятся лично → связи крепче
- После онлайна в online чате активность вырастает в 2-3x
- Появляются микро-группы друзей внутри сообщества

Lifecycle участника

Понимание пути помогает вовлекать правильно:

Стадия 1: Discovery (узнал о сообществе)

- Что делать: Четкое value proposition, легкий вход

Стадия 2: Joining (вступил в чат)

- Что делать: Welcome message, onboarding, правила

Стадия 3: Lurking (читает, молчит)

- Что делать: Создавать обсуждения, задавать вопросы, polls

Стадия 4: First participation (первый комментарий/вопрос)

- Что делать: Positive feedback, быстрый ответ, поддержка

Стадия 5: Regular contributor (регулярно участвует)

- Что делать: Благодарности, роли, вовлечение в организацию

Стадия 6: Core member (часть ядра)

- Что делать: Со-модератор, со-организатор, speaker

Стадия 7: Churn (ушел из сообщества)

- Что делать: Exit interview (почему ушел?), win-back campaigns

Метрики:

- Сколько переходят со стадии 3 на 4? (activation rate)
- Сколько остаются активными 3+ месяца? (retention)

- Почему уходят? (churn reasons)
-

7.4. Модерация и управление конфликтами

Реальность: конфликты неизбежны

Даже в самых дружественных сообществах будут:

- Flame wars (споры "tabs vs spaces", "PHP мертв или нет?")
- Токсичные участники
- Спам и реклама
- Политика и оффтоп
- Личные обиды

Задача модератора: Не устраниить конфликты (невозможно), а **управлять ими**.

Правила модерации

Принцип 1: Четкие правила + enforcement

Что работает:

- Правила прописаны и видны
- Применяются ко всем одинаково (no favorites)
- Прозрачность (почему удалили/забанили)

Что не работает:

- Правила "по ситуации"
- Избирательное применение
- Молчаливое удаление без объяснений

Принцип 2: Предупреждение → Действие

Градация:

1. **Warning** (предупреждение)
 - Первое нарушение → личное сообщение
 - "Привет, твое сообщение нарушает правило X. Удалили. Пожалуйста, не повторяй."
2. **Mute** (временный бан)
 - Второе нарушение или серьезное первое

- 1-7 дней в зависимости от тяжести
- Объяснение в личку

3. **Ban** (постоянный бан)

- Третье нарушение или очень серьезное (троллинг, хейт, спам)
- Объяснение публично (для других)

Важно:

Документируйте (кто, когда, за что). Helps в спорных случаях.

Принцип 3: Деэскалация > Бан

Когда началась flame war:

Плохо:

- Модератор: "Хватит спорить! Баню обоих!"

Хорошо:

- Модератор: "Друзья, вижу что тема вызывает эмоции. Давайте переведем в конструктивное русло: какие конкретные проблемы/решения?"
- Или: "Предлагаю продолжить в отдельном треде/канале, чтобы не захламлять общий чат"
- Или (если совсем плохо): "Прошу остановить. Продолжение в личке, если хотите."

Техники:

- Переключить на факты (вместо эмоций)
- Попросить аргументы / ссылки
- Разделить (отдельный тред)
- Дать остыть (пауза в discussion)

Типы сложных ситуаций

Ситуация 1: "Помогите, не работает!" (без деталей)

Новички часто не умеют задавать вопросы.

Плохо:

"Сам разбирайся, мы не телепаты!"

Хорошо:

- "Привет! Чтобы помочь, нужно больше информации:"
 - Какая ошибка (текст, screenshot)?

- Что пробовали?
- Какое окружение (OS, версия, etc.)?
- Или: "Вот шаблон для вопросов: [ссылка]"

Со временем качество вопросов улучшится.

Ситуация 2: Спам / реклама

Виды:

- Откровенный спам (рассылки, боты)
- "Мягкая" реклама ("Наш сервис решает эту проблему!")
- Найм-спам ("У нас открыта вакансия!")

Как справляться:

- Откровенный → ban immediately, delete
- Мягкая реклама → warning → mute если повторяется
- Найм → зависит от правил:
 - Разрешаете в отдельном канале? → переместите
 - Запрещаете совсем? → удалите

Ситуация 3: Токсичный участник

Признаки:

- Постоянно негативит
- Критикует других (не конструктивно)
- Создает конфликты
- Демотивирует новичков

Как справляться:

1. Личная беседа (может он не понимает как звучит?)
2. Если продолжается → предупреждение
3. Если не меняется → mute или ban

Важно:

Один токсик может разрушить атмосферу для 100 человек. Не бойтесь банить.

Ситуация 4: Оффтоп и политика

Проблема:

Начали обсуждать политику/религию → холивар → разделение сообщества.

Как справляться:

- Правила: "Никакой политики и религии" (если это tech community)
- Или: Отдельный канал #offtopic (если позволяете)
- Модератор останавливает сразу как началось

Исключение:

Иногда события в мире влияют на tech (санкции, блокировки). Разрешите ограниченное обсуждение с четкими правилами.

Burnout модератора

Реальность:

Модерация сообщества 500+ человек — это часы каждый день. Споры про правила. Благодарных мало, критиков много. Желание все бросить и уйти в лес.

Как избежать:

1. Разделяйте модерацию

- Минимум 2-3 модератора (распределите нагрузку)
- Ротация (каждый модерит свою неделю)
- Модератор может взять break (отпуск от модерации)

2. Автоматизация

- Боты для авто-модерации (антиспам)
- Авто-ответы на частые вопросы
- Авто-Welcome для новичков

3. Не пытайтесь контролировать все

- Сообщество должно само-регулироваться
- Community reporting (участники reportят проблемы)
- Trust активным участникам помогать новичкам

4. Помните зачем

- Когда хочется бросить, вспомните зачем начинали
- Посмотрите на positive feedback (сохраняйте его)
- Делегируйте больше (ко-организаторы, со-модераторы)

Совет из практики:

Когда появился второй active admin для DevRel чата на 900 человек — стало в разы легче. Ищите союзников, не тащите один.

Ключевые выводы главы

1. **Community building — это годы.** Не quick win. Нужно терпение и системность.
 2. **Не создавайте сообщество если оно уже есть.** Лучше присоединиться к существующему и помочь расти.
 3. **Ядро команды критично.** Не делайте один — burnout гарантирован. Минимум 2-3 организатора.
 4. **Правила нужны** даже в “свободных” сообществах. Четкие и прозрачные.
 5. **90-9-1 — нормально.** Не все будут активными. Фокус на вовлечении 9% и удержании 90%.
 6. **Модерация — сложная работа.** Конфликты неизбежны. Автоматизация и разделение нагрузки помогают.
 7. **Offline + Online = идеально.** Гибридный формат сочетает reach и depth of connections.
-

Практические задания

Для тех, кто запускает сообщество:

Неделя 1:

- [] Определите цель и аудиторию
- [] Выберите платформу
- [] Напишите правила
- [] Соберите core team (2-3 человека)

Неделя 2-4:

- [] Пригласите первых 50 участников (personal invites)
- [] Seed discussions (ядро генерит контент)
- [] Анонс в смежных сообществах

Месяц 2-3:

- [] Достигните 100 участников
- [] Организуйте первый онлайн митап
- [] Настройте регулярную активность (3-5 постов/неделя)

Месяц 4-6:

- [] Рост до 200-500
- [] Выращивайте активных (дайте роли, responsibilities)
- [] Оцените метрики, скорректируйте стратегию

Для тех, кто в сообществе состоит:

Станьте активнее:

- [] Перестаньте lurker-ить — напишите хоть один комментарий
- [] Помогите новичку с вопросом
- [] Поделитесь интересной статьей
- [] Предложите тему для митапа
- [] Придите на онлайн встречу

Предложите помощь:

- [] "Могу помочь с модерацией"
- [] "Могу со-организовать митап"
- [] "Могу сделать бота для чата"

Эффект:

Активные участники часто становятся со-организаторами, спикерами, и это великолепный карьерный буст.

Что дальше?

В следующей главе разберем **создание контента для разработчиков**: какие типы контента работают, как создавать технические туториалы, как дистрибутировать и измерять эффективность.

А пока:

Если вы организатор:

- Проверьте здоровье вашего сообщества по метрикам из главы
- Попробуйте 1-2 техники вовлечения
- Разделите модерацию (если тащите один)

Если вы участник:

- Станьте активнее (из 90% в 9%, из 9% в 1%)

- Предложите помочь организаторам
- Приведите коллегу в сообщество

Помните: Сильные сообщества строятся участниками, не только организаторами. Вы можете быть частью этого.

Статус главы: Черновик v1.0

Объем: ~2,100 слов (~8-9 страниц)

Следующая редакция: Добавить кейсы успешных российских комьюнити, детальнее про Discord setup, больше про автоматизацию модерации

Глава 8. Контент для разработчиков

Зачем эта глава: Контент — основа DevRel. Но не любой контент работает. Разработчики ненавидят маркетинговую белиберду и любят конкретику. В этой главе разберем какие типы контента работают, как создавать качественный технический контент, и как его распространять.

8.1. Типы контента: что работает для developers

1. Технические статьи и туториалы

Что:

Подробные статьи с кодом, объяснениями, примерами.

Форматы:

- **How-to guides:** "Как настроить микросервисы на Kubernetes"
- **Tutorials:** Пошаговые инструкции с кодом
- **Deep dives:** Детальный разбор технологии/подхода
- **Comparisons:** "React vs Vue: что выбрать в 2025"
- **Best practices:** "10 правил написания чистого Go кода"

Объем: 1,500-3,000 слов (medium-sized статьи работают лучше)

Временные затраты:

- Автор-эксперт: 10-20 часов
- DevRel поддержка: 4-6 встреч
- Редактор: 3-5 часов
- **Total: 2 месяца от идеи до публикации** (с учетом прокрастинации)

Что делает статью хорошей:

- Решает реальную проблему
- Примеры кода (working code, можно скопировать и запустить)
- Объяснения "почему", не только "как"
- Ссылки на доп. ресурсы
- Четкая структура (введение, problem, solution, conclusion)

Что убивает статью:

- Продающий тон ("Наш продукт лучший!")
- Отсутствие кода (только теория)
- Устаревший код (не работает)
- Вода и общие слова

2. Доклады и презентации

Типы докладов:

Experience Report:

- "Как мы мигрировали с монолита на микросервисы"
- "Как мы масштабировали систему до 100K RPS"
- Real story с цифрами, проблемами, решениями

Tech Deep Dive:

- "Внутреннее устройство нашего ML pipeline"
- "Архитектура высоконагруженной системы"
- Technical детали для экспертов

Best Practices:

- "5 уроков из 2 лет работы с Kubernetes"
- "Что мы узнали за год разработки на Rust"
- Lessons learned, применимо для других

Workshop/Hands-on:

- Участники что-то делают руками
- Live coding
- Интерактивность

Что делает доклад хорошим:

- Четкий takeaway (что participants узнают)
- Структура (problem → solution → results → lessons)
- Демо или примеры (не только слайды)
- Честность (что пошло не так тоже рассказываем)

Временные затраты:

- Новичок: 40 часов + 8 встреч с DevRel
- Опытный: 20-30 часов + 6 встреч
- Профи: 5-6 часов + финальная репетиция

3. Видео и стримы

Форматы:

Короткие видео (5-10 минут):

- Quick tips
- Tool reviews
- News и обновления

Средние (15-30 минут):

- Туториалы
- Code walkthroughs
- Tech talks записи

Длинные (1+ час):

- Workshops
- Livestreams (coding sessions)
- Подробные курсы

Площадки:

- YouTube (основная)
- Twitch (для livestreams)
- VK Видео (если РФ аудитория)

Преимущества видео:

- Показывает процесс (не только результат)
- Личный контакт (люди видят лицо, слышат голос)
- Легче воспринимается чем текст (для некоторых)

Недостатки:

- Дороже в production (видео, монтаж, звук)
- Сложнее обновлять (текст можно поправить, видео нужно переснимать)
- Требует навыков (камера, свет, монтаж)

4. Подкасты

Особенности:

- Audio-only
- Conversational format
- Легче создавать чем видео

Форматы:

- Интервью с экспертами
- Обсуждение tech news
- Deep dives в технологии

Преимущества:

- Minimum preparation (можно импровизировать)
- Идеи, которые не дооформились в статью, можно обсудить
- Аудитория слушает in commute / на пробежке

Недостатки:

- Сложнее находить аудиторию (меньше discovery)
- Требует регулярности (редкие подкасты не взлетают)

5. Документация

Типы:

- API documentation
- Getting Started guides
- Code examples
- FAQ и troubleshooting

Почему критично для B2D:

Плохая документация = разработчики не используют ваш продукт.

Best practices:

- Начните с Quick Start (5-10 минут до first success)
- Примеры кода (copy-paste ready)
- Понятная структура
- Search
- Актуальность (регулярно обновляйте)

6. Интерактивный контент

Форматы:

- Coding challenges
- Quizzes
- Interactive tutorials (CodeSandbox, StackBlitz embedded)
- Playgrounds

Преимущества:

- Высокий engagement
- Learn by doing
- Запоминается лучше

Недостатки:

- Сложнее создавать
 - Требует технической реализации
-

8.2. Создание качественного технического контента

Процесс создания статьи (детально)

Этап 1: Идея (1-3 дня)

Откуда брать идеи:

- Проблемы, которые решали в работе
- Частые вопросы в community
- Новые технологии и инструменты
- Lessons learned из проектов

Валидация идеи:

- Гуглится ли эта проблема? (keyword research)
- Есть ли уже статьи? Можно ли сделать лучше?
- Кому это интересно? (размер аудитории)

Этап 2: Структура (1-2 дня)

Outline статьи:

```
# Заголовок (проблема или обещание)
```

```
## Введение
```

- В чем проблема
- Кому актуально
- Что узнаете из статьи

```
## Контекст
```

- Background информация
- Почему проблема важна

```
## Решение
```

- Пошаговое объяснение
- Примеры кода
- Пояснения

```
## Альтернативы
```

- Другие подходы
- Сравнение (когда что использовать)

```
## Заключение
```

- Резюме
- Дальнейшие шаги
- Ссылки на ресурсы

Этап 3: Черновик (5-10 часов)

Советы:

- Пишите как объясняли бы коллеге (разговорный тон OK)
- Код первым (напишите working example, потом текст вокруг)
- Не стремитесь к совершенству (сначала done, потом perfect)

Этап 4: Редактура (3-5 часов)

Checklist:

- [] Код работает (проверьте!)
- [] Нет опечаток и грамматических ошибок
- [] Структура логичная
- [] Картинки и схемы (если нужны)
- [] Ссылки рабочие

- [] Заголовок цепляющий

Этап 5: Публикация

Pre-publication:

- Выберите теги (relevant и популярные)
- Подберите картинку для превью
- Напишите engaging intro (первый абзац решает)

Post-publication:

- Анонс в соцсетях и каналах
- Ответы на комментарии (первые часы критичны!)
- Share в relevant сообществах (но не спам!)

Как упростить процесс создания контента

Проблема:

Писать сложно. Инженеры откладывают. 2 месяца от идеи до публикации.

Решения:

1. Расшифровки докладов

- Записали доклад → расшифровали (1-2К/час) → причесали в статью
- Экономия времени автора: с 20 часов до 5

2. Интервью-формат

- DevRel берет интервью у эксперта (1 час разговора)
- Оформляет в статью
- Эксперт только комментирует и апрувует

3. Совместное написание

- Инженер пишет technical part
- DevRel/редактор пишет intro, transitions, conclusion
- Вместе редактируют

4. Контент-студии (аутсорс)

- Отдаете brief
- Студия пишет
- Вы редактируете и апрувите

- **Стоимость:** 50-100К за статью (дорого, но быстро)
-

8.3. Дистрибуция контента

Проблема: "Написали статью, а никто не читает"

Реальность:

Создание контента = 20% работы. Дистрибуция = 80%.

Правило:

Тратьте на промо столько же времени сколько на создание.

Каналы дистрибуции

1. Собственные каналы

Telegram:

- Анонс с коротким summary (3-5 предложений)
- Цепляющий первый абзац
- Call to action ("Читайте полностью по ссылке")

Email newsletter:

- Если есть подписчики
- Digest формат (топ статей месяца)

Корпоративный блог:

- Репост с Habr или наоборот

2. Профессиональные сообщества

Как делать правильно:

- Не просто кидайте ссылку ("Написал статью, читайте")
- Дайте context: "Недавно решал проблему X, написал как. Может кому полезно"
- Будьте готовы обсуждать (отвечайте на вопросы и комментарии)

Где:

- Telegram чаты по технологиям
- Discord серверы
- Reddit (r/programming, r/golang, etc. — но осторожно с self-promotion)

- LinkedIn groups

3. Social media

LinkedIn:

- Перепост с коротким summary
- Tag relevant людей и компании
- Используйте hashtags (#DevRel, #Python, etc.)

Twitter:

- Thread формат (разбейте статью на 5-7 твитов)
- Картинки и code snippets
- Mention автора и площадку

4. Aggregators

- Hacker News (если англоязычная статья и очень качественная)
- Lobsters
- [Dev.to](#)

Осторожно:

Self-promotion может быть downvote. Лучше попросите кого-то поделиться.

SEO для технических статей

Базовые принципы:

1. Keyword research

- Что гуглят разработчики?
- Google Suggest, Answer The Public
- Long-tail keywords лучше ("как настроить SSL в nginx" лучше чем "nginx SSL")

2. Title optimization

- Включайте keywords
- Обещайте конкретный результат
- Плохо: "Про микросервисы"
- Хорошо: "Миграция монолита на микросервисы: пошаговый гайд"

3. On-page SEO

- H1, H2, H3 структура

- Meta description (summary для Google)
- Alt tags для картинок
- Internal links (если на вашем блоге)

4. Evergreen контент

- Пишите не про сиюминутное ("React 18.2 released")
 - А про фундаментальное ("React hooks: complete guide")
 - Evergreen статьи работают годами
-

8.4. Метрики контента

Что измерять

Reach:

- Views (просмотры)
- Unique visitors
- Impressions (сколько раз показали)

Engagement:

- Time on page (сколько читали)
- Scroll depth (дочитали ли до конца)
- Bounce rate (ушли сразу или остались)
- Comments, likes, shares

Conversion:

- CTR (click-through rate на ваш сайт/продукт)
- Signups (регистрации)
- Trial starts (начали использовать продукт)

Quality:

- Who engaged (junior vs senior vs leads из target компаний)
- Sentiment в комментариях (positive vs negative)
- Backlinks (кто ссылается на статью)

Top performing content: что работает лучше

По данным Habr и DevRel сообщества:

Форматы-победители:

1. **Experience reports с цифрами** ("Как мы выдержали 1 млн RPS")
2. **Сравнения технологий** ("Postgres vs MySQL для high-load")
3. **Debugging stories** ("Как мы искали баг 3 дня")
4. **Architecture deep dives** ("Как устроен наш ML pipeline")
5. **Mistakes и lessons learned** ("5 ошибок при использовании Redis")

Почему работают:

- Конкретика (цифры, кейсы)
- Честность (включая failures)
- Применимость (читатель может использовать)

Форматы-аутсайдеры:

- "Мы наняли 10 человек" (никому не интересно)
 - "Наш продукт обновился" (если не показать как использовать)
 - Общие рассуждения без деталей
-

Ключевые выводы главы

1. **Контент — основа DevRel.** Без контента нет visibility и credibility.
 2. **Качество >> Количество.** Лучше 5 отличных статей чем 20 посредственных.
 3. **Разработчики любят конкретику.** Код, цифры, реальные кейсы. Не любят маркетинговую воду.
 4. **Дистрибуция = 80% работы.** Создать контент легко, донести до аудитории — сложно.
 5. **Упрощайте процесс для авторов.** Расшифровки докладов, интервью-формат, помочь с редактурой.
 6. **Измеряйте не только views.** Engagement, conversion, quality of audience важнее.
 7. **Evergreen > News.** Фундаментальный контент работает годами.
-

Статус главы: Черновик v1.0

Объем: ~1,400 слов (~6 страниц)

Следующая редакция: Добавить примеры успешных статей, детальнее про видео-контент, больше про content planning

Глава 9. Developer Advocacy: голос разработчиков

Зачем эта глава: Developer Advocacy — это не только “продвигать компанию в сообществе”, но и обратное: **представлять интересы разработчиков внутри компании.** Это двусторонняя роль, и часто вторая часть важнее. Разберем как быть адвокатом разработчиков.

9.1. Что такое Developer Advocacy

Две стороны роли

Внешняя (традиционное понимание):

- Представлять компанию/продукт в сообществе
- Выступать на конференциях
- Писать статьи
- Помогать разработчикам использовать ваш продукт

Внутренняя (часто забывают):

- Собирать feedback от разработчиков (external или internal)
- Доносить этот feedback до продуктовой/инженерной команды
- Представлять интересы developers в продуктовых решениях
- Быть “голосом пользователя” в компании

Почему обе важны:

Если вы только “продвигаете” но не слушаете → вас воспринимают как маркетолога.

Если вы слушаете и влияете на продукт → доверие и credibility в сообществе.

Чем отличается от Product Manager

Product Manager:

- Решает ЧТО строить (roadmap, priorities)
- Балансирует разные интересы (бизнес, пользователи, команда)
- Метрики: adoption, revenue, retention

Developer Advocate:

- Представляет КАК разработчики это воспринимают
- Фокус на Developer Experience (DX)
- Метрики: developer satisfaction, ease of use, community sentiment

Работают вместе:

PM определяет что строить → DA дает feedback как сделать это developer-friendly

9.2. Сбор обратной связи от разработчиков

Источники feedback

1. Прямые разговоры

Где:

- After-party на конференциях
- Митапы и встречи
- Coffee chats (1-on-1)
- Community events

Что спрашивать:

- "Какие проблемы решаете?"
- "Что неудобно в наших инструментах?"
- "Что хотели бы видеть?"
- "Почему используете конкурента, а не нас?"

Важно:

- Слушайте больше чем говорите
- Записывайте (после разговора, не во время — неловко)
- Уточняйте детали ("А можешь показать пример?")

2. Community channels

Telegram чаты, Discord, форумы:

- Мониторьте что обсуждают
- Замечайте паттерны (одна и та же жалоба от разных людей = системная проблема)
- Собирайте screenshots/quotes (с разрешения!)

Инструменты:

- Social listening tools (Mention, Brand24)
- Или manual monitoring (читайте каждый день)

3. Support tickets и GitHub issues

Что смотреть:

- Частые вопросы (значит плохая документация)
- Частые баги (значит проблема в продукте)
- Feature requests (что хотят)

Задача DA:

Не просто "это баг, отдал в support", а "это паттерн, нужно изменить подход".

4. Surveys и исследования

Типы:

- NPS опросы ("Порекомендуете ли продукт коллеге?")
- Feature voting ("Что важнее: A, B или C?")
- User research (глубокие интервью с 5-10 пользователями)

Frequency:

Квартально или раз в полгода (не чаще, люди устают от опросов).

9.3. Донос feedback до команды

Проблема: "Собрал feedback, а что дальше?"

Типичные ошибки:

✗ Dump всего feedback сразу:

"Вот 100 комментариев из чата, почитайте" → Никто не читает.

✗ Нет приоритизации:

"Люди жалуются на A, B, C, D, E..." → Команда не знает что важнее.

✗ Нет контекста:

"Пользователь X написал что API неудобный" → А что конкретно? Какой use case?

Как делать правильно

Шаг 1: Структурируйте feedback

Категоризация:

- Bugs (что сломано)
- UX issues (что неудобно)
- Feature requests (чего не хватает)
- Confusion (что непонятно → плохая документация)

Шаг 2: Приоритизируйте

Матрица Impact × Frequency:

	Часто упоминают	Редко
High impact	🔥 Fix ASAP	🟡 Consider
Low impact	🟢 Nice to have	⚪ Ignore

Шаг 3: Добавьте контекст

Для каждого feedback:

- **Что:** Конкретная проблема/запрос
- **Кто:** Кто говорит (junior vs senior, какой use case)
- **Сколько:** Как часто упоминается
- **Impact:** Как это влияет на adoption/satisfaction

Пример хорошего feedback report:

```
## Issue: SDK сложный для onboarding
```

****Проблема:**** Новые разработчики тратят 2-4 часа на первый API call

****Frequency:**** 15 упоминаний за месяц (из 50 новых разработчиков)

****Quotes:****

- "Потратил 3 часа чтобы понять как сделать простой GET request"
- "Документация запутанная, примеров нет"
- "Почему нет Quick Start guide?"

****Impact:****

- Drop-off: 30% разработчиков уходят не сделав первый call
- NPS: -20 для onboarding experience

****Recommendation:****

- Создать Quick Start Guide (10-minute to first success)
- Добавить code examples для top-3 use cases

- Упростить auth flow

****Priority:** 🔥 High (влияет на 30% новых пользователей)**

Шаг 4: Презентуйте команде

Формат:

- Регулярные синки (раз в 1-2 недели)
- Или async (Confluence page, обновляете раз в неделю)

На синках:

- Топ-3 feedback items этой недели
- Что изменилось (progress tracking)
- Новые инсайты

Как влиять на решения

Реальность:

DA не принимает финальные решения. Это делает PM/Engineering.

Роль DA:

- Предоставить data (что говорят разработчики)
- Объяснить impact (почему это важно)
- Предложить решения (как можно исправить)
- Адвокировать (убедить приоритизировать)

Когда DA эффективен:

- У вас есть credibility (команда доверяет вашим insights)
- Вы приносите structured data (не просто "мне кажется")
- Вы persistent (напоминаете, следите за прогрессом)

Когда DA бессилен:

- Компания не слушает (тогда это не advocacy, а имитация)
- Feedback противоречит бизнес-приоритетам
- Нет ресурсов на изменения

9.4. Баланс между компанией и сообществом

Этическая дилемма DA

Ситуация:

Разработчики жалуются на проблему X. Вы донесли до команды. РМ сказал: "Да, знаем, но это не приоритет на ближайший год".

Что делать:

Вариант 1: Транслировать позицию компании

"Команда в курсе, но сейчас приоритет на фичу Y, проблема X в roadmap на следующий год."

Вариант 2: Пушить внутри

Собрать больше data, показать impact, убедить изменить приоритеты.

Вариант 3: Открыто признать

"Да, это проблема. Мы работаем над решением, но нужно время. Вот workaround пока."

Баланс:

- С одной стороны, вы представитель компании
- С другой, вы должны быть honest с сообществом
- Доверие разрушается мгновенно если люди чувствуют что вы врете

Принцип:

Будьте честны. Лучше сказать "Мы в курсе, но пока нет ресурсов" чем "Все отлично, никаких проблем нет".

Когда пора уйти

Признаки что DA роль не работает:

- Компания не слушает feedback (ваши insights игнорируются)
- Вас просят "продавать" плохой продукт (этически неприемлемо)
- Нет влияния на решения (вы просто рупор, не адвокат)
- Внутренние проблемы (HR, культура) разрушают внешний бренд

Что делать:

1. Попытаться изменить (эскалация к СТО/CEO)
 2. Если не меняется → ищите другую компанию (где DA ценят)
-

Ключевые выводы главы

1. **Developer Advocacy — двусторонняя роль.** Не только "говорим о компании", но и "слушаем разработчиков".

- 2. Слушание важнее говорения.** Собирайте feedback постоянно, структурируйте, приоритизируйте.
- 3. Донос feedback — навык.** Не dump данных, а structured insights с приоритетами и рекомендациями.
- 4. Влияние требует credibility.** Команда должна доверять вашим insights.
- 5. Баланс между компанией и сообществом.** Будьте честны. Доверие — ваш главный актив.
- 6. Если компания не слушает — уходите.** DA без влияния = бессмысленная роль.

Статус главы: Черновик v1.0

Объем: ~1,500 слов (~6 страниц)

Следующая редакция: Добавить больше примеров feedback reports, кейсы успешного advocacy, framework для prioritization

ЧАСТЬ 4. МЕРОПРИЯТИЯ

Глава 10. Типы DevRel мероприятий

Зачем эта глава: Мероприятия — ключевой инструмент DevRel. Но митап, конференция, хакатон, воркшоп — это разные форматы с разными целями, бюджетами и эффектами. Разберем когда что использовать.

10.1. Митапы

Формат:

- 30-150 участников
- 2-4 часа
- 2-3 доклада или 1 воркшоп
- Неформальный нетворкинг

Цели:

- Community building
- Regular touchpoint с аудиторией
- Testing ground для спикеров
- Локальная visibility

Бюджет: 30-200К

Организация: 4-8 недель

См. Главу 11 для детального чек-листа

10.2. Конференции

Форматы:

Малые (200-500):

- Региональные или нишевые
- 1-2 дня
- 1-2 трека
- **Стоимость участия для спонсора:** 200-500К

Средние (500-1500):

- Крупные региональные или нишевые федеральные
- 2 дня, 2-3 трека
- **Стоимость:** 500К - 2 млн

Крупные (1500+):

- HighLoad++, Mobius, TeamLead Conf
- 2-3 дня, 4-6 треков
- **Стоимость:** 2-10 млн+

Форматы участия:

1. Спикер (бесплатно, но нужно пройти CFP)

- Visibility
- Credibility
- Нетворкинг

2. Спонсор со стендом

- Дорого
- Высокая visibility
- Прямой контакт с тысячами
- См. Главу 12 про геймификацию стендов

3. Партнер (дешевле чем спонсор)

- Логотип на сайте
- Упоминание
- Иногда небольшой стенд

10.3. Хакатоны

Формат:

- 24-48 часов интенсивной разработки
- Команды создают проекты
- Призы для победителей

Цели:

- Product adoption (если хакатон вокруг вашего API)

- Hiring (находите сильных разработчиков)
- Innovation (участники могут создать что-то полезное)

Стоимость: 300К - 2 млн (зависит от масштаба и призов)

Призы:

- 1 место: 200-500К
 - 2-3 места: 100-200К
 - Спец призы
-

10.4. Воркшопы

Формат:

- Hands-on обучение
- 10-30 участников
- 2-4 часа
- Участники что-то создают руками

Цели:

- Обучить использованию вашего продукта
- Deep engagement (больше чем на докладе)
- Сбор детального feedback

Бюджет: 20-100К

10.5. Онлайн vs офлайн

Онлайн:

- Geographic reach (вся страна/мир)
- Дешевле
- Легче организовать
- Меньше engagement
- Нет личного контакта

Офлайн:

- Сильные связи

- Запоминается
- Networking эффективнее
- Дороже
- Geographic ограничение

Гибридный:

- Оффлайн event + online stream
- Лучшее из обоих миров
- Но сложнее организовать

Статус главы: Черновик v1.0

Объем: ~900 слов (~4 страницы)

Следующая редакция: Расширить каждый раздел, добавить сравнительную таблицу, кейсы разных форматов

Глава 11. Организация митапов: от идеи до follow-up

Зачем эта глава: Митапы — один из самых эффективных инструментов DevRel для employer branding и community building. Но организация даже небольшого митапа требует внимания к десяткам деталей. В этой главе — полный чек-лист и практические советы из реального опыта.

11.1. Планирование митапа: сроки, бюджет, чек-листы

Что такое митап в контексте DevRel?

Митап (meetup) — неформальное мероприятие для профессионального сообщества, обычно:

- 30-150 участников
- 2-4 часа длительность
- 2-3 доклада или воркшоп
- Неформальный нетворкинг (до/после/между докладами)
- Часто бесплатный для участников

Чем митап отличается от конференции:

- Меньше масштаб (conference: 200-2000+)
- Менее формально
- Дешевле организовать
- Проще logistics
- Быстрее организуется (1-2 месяца vs 6-12)

Зачем компании организовывать митапы:

- Positioning как thought leaders
- Employer branding (показать команду и офис)
- Networking с потенциальными кандидатами
- Сбор feedback от сообщества
- Построение комьюнити вокруг технологий
- Testing ground для спикеров перед большими конференциями

Timeline: за сколько начинать подготовку

Минимум: 4-6 недель (если все звезды сошлись)

Комфортно: 8-12 недель (для первого митапа)

Оптимально: 3-4 месяца (для крупного митапа 100+ человек)

Пример timeline (10 недель):

Неделя -10 (за 10 недель до события):

- [] Определить тему и формат
- [] Согласовать с руководством
- [] Оценить бюджет

Неделя -8:

- [] Забронировать площадку
- [] Найти 2-3 спикеров (подтвердить участие)
- [] Создать landing page / регистрацию

Неделя -6:

- [] Анонс в соцсетях и сообществах
- [] Договориться с кейтерингом
- [] Заказать мерч / печать материалов (если нужно)

Неделя -4:

- [] Reminder участникам
- [] Проверить прогресс регистрации
- [] Провести репетицию со спикерами

Неделя -2:

- [] Финальная репетиция
- [] Подготовить реквизит (презентеры, бейджи, печатные материалы)
- [] Подтвердить все договоренности (площадка, кейтеринг, фото/видео)

Неделя -1:

- [] Reminder #2 участникам
- [] Финальный чек-лист организатора
- [] Brief команде (кто за что отвечает)

День события:

- [] Прийти за 2-3 часа до начала
- [] Set up (оборудование, регистрация, кейтеринг)
- [] Провести митап
- [] Clean up

Неделя +1 (после):

- [] Разослать thank you notes
- [] Опубликовать записи / фото / отчет
- [] Собрать feedback
- [] Проанализировать метрики
- [] Follow-up с интересными людьми

Бюджет: сколько стоит митап

Минимальный митап (30 человек):

Статья расхода	Сумма
Площадка	0-20К (может быть бесплатно)
Кейтеринг	15-30К (500-1000Р на человека)
Мерч	10-20К (стикеры, футболки)
Фото/видео	5-10К (если нанимаете)
Печать материалов	2-5К
ИТОГО	30-85К

Средний митап (70 человек):

Статья расхода	Сумма
Площадка	20-50К
Кейтеринг	50-100К
Мерч	30-50К
Фото/видео	15-30К
Продвижение	10-20К
Печать + реквизит	10-15К

Статья расхода	Сумма
ИТОГО	135-265K

Большой митап (150 человек):

Статья расхода	Сумма
Площадка	50-150K
Кейтеринг	150-300K
Мерч	50-100K
Фото/видео	30-50K
Продвижение	20-40K
Оборудование (если нужно)	20-50K
ИТОГО	320-690K

Способы снизить бюджет:

- Площадка: попросите в IT-компаниях (часто дают бесплатно для комьюнити)
 - Кейтеринг: только напитки + пицца (вместо полноценного фуршета)
 - Фото/видео: попросите друзей или сделайте сами
 - Мерч: only stickers (дешево и любят)
-

11.2. Поиск площадки и спикеров

Как найти площадку для митапа

Критерии:

- Вместимость (с запасом 20-30%)
- Локация (метро, парковка, понятный адрес)
- Проектор + экран (или большой ТВ)
- Микрофон / звук (для 50+ человек обязателен)
- Wi-Fi (для IT-аудитории критично!)
- Комфорт (кондиционер, туалеты, места для сидения)
- Кейтеринг зона (место для еды и напитков)

Варианты площадок:

1. Офис IT-компании (часто бесплатно)

- Плюсы: Бесплатно/дешево, готовая инфраструктура, показываете офис
- Минусы: Ограничения по времени (секьюрити), не всегда central location

2. Коворкинги (15-50K)

- Плюсы: Гибкие условия, центр города, готовое оборудование
- Минусы: Платно, может быть шумно

3. Конференц-залы отелей (30-100K)

- Плюсы: Профессионально, можно large events
- Минусы: Дорого, формально

4. Антикафе / лофты (10-30K)

- Плюсы: Атмосферно, необычно
- Минусы: Не всегда подходящее оборудование

Лайфхак:

Договоритесь с 2-3 IT-компаниями об обмене площадками. Вы даете им свой офис для митапов, они вам свой. Win-win.

Как найти спикеров

Проблема #1: "Никто не хочет выступать!"

Реальность:

Многие хотят, но:

- Страх (синдром самозванца: "Мне нечего рассказать")
- Нет времени ("Завалило рабочими проектами")
- Не знают как (никогда не делали доклад)

Решения:

Источник 1: Внутренняя команда

Поиските в своей компании:

- Сеньоров с опытом (проще начать)
- Инженеров, которые активны в чатах (любят объяснять)
- Тех, кто делал internal tech talks

- Новичков с крутыми pet-projects

Как вовлечь:

- Персональное приглашение (не mass-рассылка)
- Помощь с подготовкой ("Я буду с тобой каждый шаг")
- Официальное разрешение (согласование с менеджером)
- Ценность для карьеры ("Это в твоем performance review")

Источник 2: Внешние спикеры

- Активисты сообщества
- Авторы популярных статей на Habr
- Мейнтейнеры open-source проектов
- Преподаватели курсов
- DevRel-ы из других компаний (обмен спикерами)

Как привлечь:

- Понятная Value proposition ("Аудитория 80 человек, запишем и выложим на YouTube")
- Компенсация (если external): 10-30K за доклад или бартер
- Quality event (хорошая организация, теплая атмосфера)

Источник 3: CFP (Call for Papers)

Открытый набор докладов:

- Создаете форму с описанием митапа
- Даёте 2-3 недели на подачу заявок
- Отбираете лучшие

Плюсы:

- Находите новых спикеров
- Разнообразие тем
- Прозрачность (community ценит)

Минусы:

- Может прийти мало заявок
- Нужно время на отбор и коммуникацию

Подготовка спикеров

Для начинающих спикеров (подробности в Главе 2):

8 встреч в процессе подготовки:

1. Про идею (brainstorm темы) 2-3. Про тезисы и анонс (для CFP)
2. Про структуру доклада 5-6. Про слайды и черновые прогоны
3. Публичное выступление для своих (внутренний тест)
4. Финальный прогон за пару дней до выступления

3 месяца — оптимальный срок подготовки:

- Меньше → не успеет морально созреть
- Дольше → заколебется быть в напряжении

Для опытных:

- 6 встреч
 - 2 месяца
 - Финальная репетиция обязательна
-

11.3. Продвижение и регистрация

Как привлечь участников

Каналы продвижения:

1. Собственные каналы

- Telegram канал компании
- Email рассылка (если есть база)
- Корпоративные соцсети (LinkedIn, VK)
- Корпоративный блог

2. Профессиональные сообщества

- Telegram чаты разработчиков (Moscow Python, Spb Frontend, etc.)
- Профильные каналы (DevRel, specific tech)
- Slack/Discord сообщества

3. Площадки для митапов

- meetup.com
- timepad.ru

- leader-id.ru

4. Партнерские анонсы

- Попросите спикеров анонсировать в своих каналах
- Обменяйтесь анонсами с другими организаторами
- Попросите комьюнити-лидеров поделиться

Timing анонсов:

- **-6 недель:** Первый анонс ("Save the date, детали скоро")
- **-4 недели:** Полный анонс (спикеры, темы, регистрация)
- **-2 недели:** Reminder ("Осталось 2 недели, места заканчиваются")
- **-3 дня:** Финальный reminder ("Завтра встреча, ждем!")
- **-1 день:** Практическая информация (адрес, как пройти, что принести)

Регистрация: как правильно

Зачем вообще регистрация:

- Понимать сколько человек придет (для кейтеринга и площадки)
- Собрать контакты (email для follow-up)
- Отсеять нерелевантных (если нужно)
- Создать sense of commitment

Что спросить при регистрации:

Минимум:

- [] Имя
- [] Email
- [] Компания (опционально, но полезно)
- [] Согласие на обработку данных

Полезно добавить:

- [] Уровень (junior / middle / senior)
- [] Роль (backend / frontend / devops / etc.)
- [] "Откуда узнали про митап?" (трекинг каналов)
- [] "Что хотели бы услышать?" (сбор тем для будущего)

Платформы:

- Google Forms (бесплатно, просто)
- Timepad (популярно в РФ, готовая аудитория)
- [meetup.com](https://www.meetup.com) (международная аудитория)
- Custom форма на сайте (если есть ресурсы)

Overbooking: регистрируй с запасом

Проблема:

Регистрируется 100 человек, приходит 60. Пустой зал, еда остается.

Причины:

- Люди забывают
- Планы меняются
- Регистрируются "на всякий случай"

Типичный show-up rate:

- Бесплатные митапы: 50-70%
- Платные: 80-90%
- Корпоративные (для сотрудников): 60-70%

Решение: Overbooking

Если площадка на 80 человек → регистрируйте 110-120.

Как контролировать:

- Закрывайте регистрацию за 3-5 дней
- Шлите reminders ("Подтвердите участие")
- Создайте waitlist (если переборщили)

Формула:

Целевое количество участников / Ожидаемый show-up rate = Регистраций

Пример: 80 человек / 0.65 = 123 регистрации

11.4. День события: чек-лист организатора

За 2-3 часа до начала

Setup:

- [] Пришла команда (минимум 2-3 человека)
- [] Протестирували проектор и презентации
- [] Микрофон работает (протестируйте звук!)
- [] Wi-Fi работает (пароль на слайде/флип-чарте)
- [] Регистрационная зона готова (бейджи, списки)
- [] Кейтеринг приехал и расставлен
- [] Указатели / навигация (если большое здание)
- [] Фотограф на месте (или камера настроена)

Презентеры/кликеры:

- [] Протестирували заранее
- [] Знаете какие кнопки гасят экран (часто бывает!)
- [] Запасные батарейки

Для спикеров:

- [] Зона для подготовки (тихое место)
- [] Вода на сцене
- [] Проверили слайды на проекторе
- [] Знают тайминг (сколько минут)

Во время митапа

Регистрация (первые 30 минут):

- [] Дружелюбный reception
- [] Быстрая выдача бейджей
- [] Указание где кейтеринг / туалеты / зал

Открытие:

- [] Welcome speech (3-5 минут)
 - Спасибо что пришли
 - Что будет сегодня (программа)
 - Housekeeping (Wi-Fi, туалеты, fire exit)
 - Представление спонсоров (если есть)

Во время докладов:

- [] Следите за таймингом (дайте знать спикеру за 5 и 2 минуты)
- [] Модерируйте Q&A (если нужно)
- [] Записывайте видео / делайте фото
- [] Кто-то из команды available для решения проблем

Между докладами:

- [] Анонсируйте перерыв
- [] Напомните где кейтеринг
- [] Encourage networking
- [] Пополняйте еду/напитки

Closing:

- [] Спасибо спикерам и участникам
- [] Анонс следующего митапа (если планируете)
- [] Призыв к действию ("Подпишитесь на канал", "Заполните feedback")

После митапа

В тот же вечер:

- [] Выложите фотки в соцсети ("Спасибо всем кто пришел!")
- [] Thank you в чате/канале

На следующий день:

- [] Отправьте email всем зарегистрированным
 - Спасибо участникам
 - Ссылки на презентации спикеров
 - Ссылка на запись (если есть)
 - Форма feedback
 - Анонс следующего

Через неделю:

- [] Опубликуйте запись на YouTube
- [] Напишите краткий отчет (с фото и highlights)
- [] Соберите и проанализируйте feedback
- [] Follow-up с интересными людьми (потенциальные кандидаты, партнеры, следующие спикеры)

11.5. Метрики успеха митапа

Количественные метрики

До события:

- Регистрации (сколько, динамика)
- Источники (откуда узнали)
- Conversion rate (landing page views → регистрации)

Во время:

- Show-up rate (пришло / зарегистрировалось)
- Attendance retention (сколько ушли до конца)
- Networking activity (много ли общались)

После:

- Просмотры записи на YouTube
- Engagement в соцсетях (лайки, комментарии, репосты)
- Подписки на канал/рассылку

Качественные метрики

NPS митапа:

"Оцените от 0 до 10: насколько вероятно, что вы порекомендуете наши митапы коллеге?"

Цель: NPS > 50 (отлично), > 30 (хорошо), < 10 (есть проблемы)

Feedback questions:

- Что понравилось больше всего?
- Что можно улучшить?
- Какие темы хотели бы услышать в будущем?
- Будете ли приходить снова?

ROI митапа

Входящие затраты:

Например: 150K (бюджет) + 60 часов DevRel времени

Выходящие результаты:

- 70 участников, из них:
 - 20 потенциальных кандидатов (качественные контакты)
 - 5 в pipeline найма
 - 1 нанят за 3 месяца
- Видео на YouTube: 2000 просмотров за год
- +150 подписчиков в Telegram канал
- 3 статьи в блог (репосты презентаций спикеров)

ROI расчет:

Ценность = (1 найм × 500К стоимость) + (2000 просмотров × CPM) + (Brand lift)

Затраты = 150К + (60 часов × стоимость часа DevRel)

ROI = Ценность / Затраты

Даже консервативная оценка:

Если митап привел к найму 1 человека за полгода → ROI 2-3x.

Но главное — не прямой ROI:

- Reputation и brand awareness
- Укрепление комьюнити
- Практика для internal спикеров
- Feedback от разработчиков

11.6. Частые ошибки и как их избежать

Ошибка 1: Плохой Wi-Fi

Почему критично:

IT-аудитория будет гуглить во время докладов, заливать фотки, проверять почту. Плохой Wi-Fi = недовольство.

Как избежать:

- Протестируйте заранее (попросите 5-10 человек одновременно нагрузить)
- Узнайте пароль и пропишите на видном месте
- Если Wi-Fi слабый — предупредите заранее

Ошибка 2: Нет тайм-менеджмента

Проблема:

Спикер говорит 50 минут вместо 30. Митап затягивается на час. Люди уходят.

Как избежать:

- Четкий тайминг для спикеров (30 минут доклад + 10 Q&A)
- Сигналы: за 5 минут, за 2 минуты, за 0
- Модератор с правом прервать (вежливо)

Ошибка 3: Забыли про newcomers

Проблема:

Постоянные участники общаются между собой. Новички стоят в углу, не знают кого.

Как избежать:

- Бейджи с именами (крупный шрифт!)
- Icebreaker activity в начале
- Назначьте 2-3 человека из команды: "Ваша задача — познакомить новичков с другими"
- Networking bingo (см. Главу 12)

Ошибка 4: Плохой кейтеринг

Проблемы:

- Недостаточно еды (кончилась за 10 минут)
- Неподходящая еда (вегетарианцы, аллергики)
- Неудобно есть (нет вилок, тарелок)

Как избежать:

- Заказывайте с запасом 20%
- Уточните dietary restrictions при регистрации
- Варианты: вегетарианские, без глютена
- Простая еда: пицца, сэндвичи, фрукты (не нужны вилки)
- Напитки: вода, соки, кофе/чай

Ошибка 5: Нет follow-up

Проблема:

Митап прошел классно, но потом — тишина. Люди забыли, momentum потерян.

Как избежать:

- Email next day (спасибо + материалы)
 - Публикация видео через 3-5 дней
 - Анонс следующего митапа через неделю
 - Personal follow-up с ключевыми людьми
-

Ключевые выводы главы

1. **Начните заранее.** Минимум 6-8 недель для комфортной подготовки первого митапа.
 2. **Бюджет можно оптимизировать.** Даже 30-50К достаточно для качественного митапа на 30 человек.
 3. **Площадка и спикеры — две критичные точки.** Забронируйте площадку рано, начните искать спикеров за 2-3 месяца.
 4. **Overbooking — нормально.** Регистрируйте на 30-50% больше чем вмещает площадка.
 5. **Детали решают.** Wi-Fi, тайминг, кейтеринг, работа с новичками — мелочи создают общее впечатление.
 6. **Follow-up критичен.** Митап не заканчивается когда люди разошлись. Email, видео, анонс следующего — это часть опыта.
 7. **ROI митапов — долгосрочный.** Не ждите немедленного найма. Это про репутацию, отношения, community.
-

Чек-лист организатора митапа (полный)

За 8-12 недель:

- [] Определить тему, формат, дату
- [] Согласовать с руководством и получить бюджет
- [] Забронировать площадку
- [] Найти 2-3 спикеров

За 6 недель:

- [] Создать landing page / регистрацию
- [] Первый анонс

- [] Начать подготовку спикеров

За 4 недели:

- [] Полный анонс (программа, спикеры)
- [] Договориться с кейтерингом
- [] Заказать мерч / печать (если нужно)

За 2 недели:

- [] Reminder участникам
- [] Репетиция со спикерами
- [] Подтвердить площадку и кейтеринг

За 1 неделю:

- [] Финальная репетиция
- [] Подготовить реквизит
- [] Brief команде

За 1 день:

- [] Reminder с практической информацией
- [] Проверить что все готово

День события:

- [] Прийти за 2-3 часа
- [] Setup
- [] Провести митап по плану
- [] Clean up

После:

- [] Email участникам (следующий день)
 - [] Публикация видео (3-5 дней)
 - [] Отчет и анализ метрик (неделя)
 - [] Follow-up с ключевыми людьми
-

Что дальше?

В следующей главе мы вернемся к геймификации, которую уже рассматривали в Главе 12, но теперь в контексте **всех типов DevRel мероприятий**, не только стендов.

А пока — попробуйте:

Если вы еще не организовывали митапы:

1. Начните с internal tech talk в компании (проще logistics)
2. Со-организуйте с кем-то (разделите ответственность)
3. Начните с малого (30 человек, простой формат)

Если уже организовываете:

1. Пройдите чек-лист — что можно улучшить?
2. Добавьте 1-2 новых элемента (игрофикация, новый формат)
3. Систематизируйте (регулярность: раз в месяц/квартал)

Помните: Первый митап всегда сложнее. Второй уже проще. К пятому у вас будет отлаженная машина.

Статус главы: Черновик v1.0

Объем: ~2,000 слов (~8 страниц)

Следующая редакция: Добавить более детальный бюджет, примеры программ митапов, шаблоны анонсов и писем (в приложение)

Глава 12. Геймификация и активации на IT-мероприятиях

Ключевая идея главы: Простая раздача сувениров и сбор контактов на стенах показывает низкую эффективность. Настоящую ценность создают интерактивные форматы, где реальные инженеры компании увлеченно общаются с посетителями о технологиях и проектах. Такой подход формирует репутацию и дает материал для контента на месяцы вперед.

Введение: Почему геймификация работает на IT-мероприятиях

Представьте типичный IT-митап или конференцию:

- Стенды компаний с логотипами и флаерами
- Промоутеры, раздающие ручки с названием компании
- QR-коды для подписки на вакансии
- Люди проходят мимо, не задерживаясь

А теперь представьте другую картину:

- У стенда очередь
- Люди смеются, активно общаются с командой
- Фотографируются, делятся в соцсетях
- Возвращаются к стенду несколько раз за вечер
- После мероприятия вспоминают именно вашу компанию

В чем разница? Геймификация и интерактивные активации.

Что такое геймификация в контексте DevRel?

Геймификация — это применение игровых механик и элементов в неигровых контекстах для повышения вовлеченности, мотивации и создания позитивного опыта.

В контексте DevRel мероприятий это означает:

- Превратить посещение стенда в увлекательный опыт
- Создать запоминающиеся взаимодействия с брендом
- Дать людям повод вернуться и пообщаться с вашей командой
- Получить органическое продвижение через UGC (user-generated content)

Важно понимать: Геймификация на IT-событиях — это не детские развлечения. Это продуманная система вовлечения профессиональной аудитории через игровые механики, которые:

- Соответствуют интересам разработчиков
 - Не отвлекают от главного (общение, нетворкинг)
 - Создают повод для разговора о технологиях компании
 - Помогают собрать качественные контакты и обратную связь
-

12.1. Почему геймификация работает: научное обоснование

Прежде чем переходить к практическим механикам, важно понять **почему** игровые элементы так эффективны. Это не магия и не манипуляция — это использование естественных свойств человеческого мозга.

Октализ: 8 ключевых мотиваторов

Ю-Кай Чоу в книге "Actionable Gamification" описал систему **Octalysis** — 8 основных факторов мотивации, на которых строится любая успешная геймификация:

1. Epic Meaning & Calling (Миссия и призвание)

Людям важно чувствовать, что они часть чего-то большего.

Применение на IT-мероприятиях:

- "Помоги выбрать следующую open-source фичу нашей библиотеки"
- "Стань частью исследования: как разработчики решают проблему X"
- "Твой голос влияет на roadmap нашего продукта"

2. Development & Accomplishment (Достижения и прогресс)

Мозг любит видеть прогресс и отмечать достижения.

Применение:

- Чек-листы активностей ("Собери 5 стикеров с разных стендов")
- Уровни участия ("Новичок → Эксперт → Мастер")
- Видимый прогресс ("Пройди 3 из 5 станций квеста")

Научное обоснование:

Исследования показывают, что разбиение большой задачи на маленькие подзадачи с видимым прогрессом **повышает мотивацию и снижает когнитивную нагрузку**. Когда

человек ставит галочку или получает наклейку, его мозг воспринимает это как завершение цикла и вознаграждает выбросом дофамина.

Механизм работы: Физическое действие по завершению задачи (клик, галочка, получение стикера) служит для мозга сигналом о закрытии когнитивного цикла. Это помогает восстанавливать ментальную энергию и создает ощущение достижения, что мотивирует продолжать активность.

3. Empowerment of Creativity & Feedback (Креативность и фидбек)

Людям нравится создавать что-то свое и получать обратную связь.

Применение:

- "Создай свой мем про баги" (печать на футболках/стикерах)
- "Напиши код, который решает эту задачу необычным способом"
- "Собери схему архитектуры из кубиков Lego"

4. Ownership & Possession (Владение и коллекционирование)

Люди ценят то, что "принадлежит" им или что они собирали.

Применение:

- Коллекционные стикеры/значки (разные на разных событиях)
- "Собери полную коллекцию стикеров нашего стека технологий"
- Персонализированный мерч ("Твоя любимая IDE" на футболке)

5. Social Influence & Relatedness (Соц. давление и связи)

Мы социальные существа — нам важно мнение окружающих.

Применение:

- Лидерборды ("Топ-10 участников квиза")
- Командные активности (соревнование между столами/компаниями)
- Фотозоны и share-able контент
- "Приведи друга — получи бонус"

6. Scarcity & Impatience (Дефицит и нетерпение)

Ограниченност ресурсов повышает ценность.

Применение:

- "Только 50 первых участников получат limited edition мерч"
- "Секретный приз разыгрывается в 19:00 — не пропусти"

- "Exclusive доступ к закрытой бете продукта"

7. Unpredictability & Curiosity (Непредсказуемость и любопытство)

Мозг обожает сюрпризы и тайны.

Применение:

- Лотереи и случайные призы
- "Угадай, что в черном ящике?"
- Секретные задания ("Найди 5 спрятанных QR-кодов на стенде")
- Easter eggs в демо продукта

8. Loss & Avoidance (Избегание потерь)

Страх упустить возможность (FOMO) — мощный мотиватор.

Применение:

- "Акция действует только сегодня"
- "Последние 5 мест на закрытую встречу с СТО"
- "Не пропусти розыгрыш MacBook в конце дня"

Этические рамки: При использовании этих мотиваторов важно помнить о балансе. Цель геймификации — создать позитивный и запоминающийся опыт взаимодействия с брендом, а не манипулировать поведением людей. Все механики должны быть прозрачными, честными и давать реальную ценность участникам.

Как мозг реагирует на игровые механики

Недавние исследования показывают:

1. Чек-листы снижают ошибки и повышают вовлеченность

Доказано в авиации и медицине — работает и на мероприятиях.

2. Разбиение на подзадачи повышает мотивацию

Особенно на начальных этапах, когда конечная цель кажется далекой.

3. Физическое действие усиливает эффект

Клик, галочка, наклейка — все это создает ощущение завершения цикла.

4. Однозадачность эффективнее многозадачности

Люди лучше вовлекаются, когда у них одна четкая активность, а не 10 параллельных.

Практический вывод:

Создавайте активности с четкими, небольшими шагами, видимым прогрессом и физическим подтверждением выполнения.

12.2. Форматы игровых активностей на стендах

Теперь перейдем к конкретным форматам, которые работают на IT-мероприятиях.

Формат 1: Технический квиз

Суть: Викторина на знание технологий, best practices, истории IT.

Механика:

- Планшет/экран с вопросами на стенде
- 10-15 вопросов разной сложности
- Мгновенный результат + место в лидерборде
- Призы топ-3 участников в конце дня

Примеры вопросов:

- "Какая команда Git отменяет последний коммит?"
- "В каком году вышел первый iPhone?"
- "Что означает аббревиатура SOLID?"
- "Сколько HTTP статус-кодов существует?"

Почему работает:

- Разработчики любят проверить свои знания
- Элемент соревнования (лидерборд)
- Быстро (5-7 минут)
- Можно сделать вопросы про вашу компанию/продукт (1-2 из 15)

Реквизит:

- Планшет/экран с формой (Google Forms, Typeform, custom)
- Монитор для отображения лидерборда
- Призы для победителей

Бюджет: 10-30К (разработка формы + призы)

Формат 2: Кодинг-челлендж

Суть: Решить алгоритмическую задачу прямо на стенде.

Механика:

- Экран с IDE (VSCode/PyCharm/online editor)

- 3-5 задач разной сложности
- Можно выбрать язык программирования
- Тайминг: 10-20 минут
- Засчитываются только работающие решения

Примеры задач:

- Легкая: "Напиши функцию FizzBuzz"
- Средняя: "Найди все анаграммы в массиве строк"
- Сложная: "Реализуй LRU Cache"

Варианты усложнения:

- Ограничение по времени
- Ограничение по количеству строк кода
- "Code golf" — кто короче напишет
- Pair programming (команда из 2 человек)

Почему работает:

- Показывает, что ваша компания ценит технические скиллы
- Сразу видно сильных кандидатов
- Естественный повод начать разговор о технологиях
- Можно снимать на видео для соцсетей (с разрешения)

Реквизит:

- Ноутбук/планшет с IDE
- Монитор для наблюдателей
- Таймер
- Призы

Бюджет: 5-15К (настройка окружения + призы)

Лайфхак: Используйте вашу собственную open-source библиотеку в задачах — покажете продукт в деле.

Формат 3: "Угадай технологию" (слепой тест)

Суть: Определить технологию/фреймворк/инструмент по коду или описанию.

Механика:

- Показываем snippet кода БЕЗ подсветки синтаксиса и названия языка
- Участник угадывает: язык, фреймворк, или даже конкретную библиотеку
- 5-10 раундов
- Подсчет правильных ответов

Примеры:

```
@app.route('/hello')
def hello():
    return 'Hello World!'
```

Ответ: *Flask (Python)*

```
const [count, setCount] = useState(0);
```

Ответ: *React Hooks (JavaScript)*

Варианты:

- Угадать язык программирования
- Угадать фреймворк
- Угадать компанию по стеку технологий
- Угадать продукт по архитектуре

Почему работает:

- Быстро и весело
- Не требует реального кодинга (подходит для большего числа людей)
- Можно играть группами
- Генерирует дискуссии ("Это точно Vue, а не React!")

Реквизит:

- Экран с слайдами/форма
- Кнопки для ответов
- Или просто карточки с вариантами

Бюджет: 2-5К (подготовка контента)

Формат 4: Геймифицированный стенд с квестом

Суть: Посетитель проходит 3-5 "станций" на стенде, выполняя задания.

Пример квеста "Путь разработчика":

Станция 1: Онбординг

- Задание: "Пройди быстрый тест на знание нашей компании (3 вопроса)"
- Награда: Стикер "Junior Developer"

Станция 2: Code Review

- Задание: "Найди 3 бага в этом коде"
- Награда: Стикер "Reviewer"

Станция 3: Деплой

- Задание: "Собери правильную последовательность CI/CD пайплайна"
- Награда: Стикер "DevOps"

Станция 4: Production

- Задание: "Сыграй в мини-игру: отрази DDoS атаку (кликалка)"
- Награда: Стикер "SRE"

Станция 5: Бонус

- Задание: "Пообщайся с нашим инженером 5 минут, узнай про проект"
- Награда: Стикер "Team Player"

Финал:

Собрал все 5 стикеров → получаешь финальный приз (футболка/мерч/доступ к продукту)

Почему работает:

- Люди проводят у стенда 15-30 минут (вместо 30 секунд)
- Проходят через все key messages компании
- Общаются с разными членами команды
- Видимый прогресс (коллекция стикеров)
- Эффект завершения ("Я начал — нужно закончить")

Реквизит:

- Карта квеста (бумажная или digital)
- 5 станций на стенде
- Стикеры/штампы для отметок

- Финальный приз

Бюджет: 30-100К (зависит от размера стенда и призов)

Лайфхак: Добавьте "secret level" — спрятанное задание для самых внимательных.

Формат 5: Лотерея/розыгрыш (но правильный)

Классика: "Оставь визитку — участвуй в розыгрыше iPhone"

Проблема: Люди бросают визитки не глядя, не вовлекаются, 95% контактов "холодные".

Улучшенная версия:

Механика "Заработай билеты":

- Каждая активность на стенде = 1 лотерейный билет
- Прошел квиз → 1 билет
- Решил задачу → 2 билета
- Пообщался с инженером → 3 билета
- Подписался на блог → 1 билет
- Привел друга → 2 билета

Финал:

В конце дня розыгрыш. Чем больше билетов — тем выше шанс.

Почему лучше:

- Люди вовлечены, а не просто бросили визитку
- У вас уже есть контакт + context (чем человек интересовался)
- Ощущение fairness — кто больше участвовал, у того выше шансы
- Люди возвращаются к стенду ("А если еще заданий пройду?")

Важно:

Разыгрывайте что-то ценное для разработчиков:

- MacBook / мощный ноутбук / мощная клавиатура
 - Годовая подписка на JetBrains / GitHub Copilot
 - Билет на крупную конференцию
 - НЕ ручки, блокноты, носки
-

Формат 6: Фотозона с AR/VR элементами

Суть: Создать Instagram-worthy experience.

Варианты:

Классика:

- Брендированный баннер
- Props (очки, усы, таблички с code-шутками)
- Хорошее освещение

Продвинутый уровень:

- AR фильтр (через смартфон)
- Green screen с подстановкой фона (офис компании, космос, матрица)
- VR демо вашего продукта
- Роботизированная рука, которая фотографирует

Механика вовлечения:

- Сфотографался → отметь нас в Instagram → участвуй в конкурсе
- Лучшее фото дня → приз
- Репост → дополнительный шанс в розыгрыше

Почему работает:

- Organic reach (люди сами постят в соцсети)
- Контент для вашего SMM
- Люди приводят друзей ("Пойдем сфоткаемся!")
- Запоминается

Бюджет: 15-150K (от простого баннера до AR-фильтра)

Лайфхак: Сделайте несколько вариантов фона/фильтров, чтобы люди возвращались.

Формат 7: "Собери архитектуру" (Lego/магниты)

Суть: Физическая активность — построить что-то из блоков.

Варианты заданий:

- Собери микросервисную архитектуру из кубиков Lego
- Построй схему базы данных из магнитов на доске
- Создай CI/CD пайплайн из карточек

Механика:

1. Участник получает задание ("Построй отказоустойчивую систему")
2. Даётся набор блоков (каждый блок = компонент: сервер, БД, load balancer, etc.)
3. Время: 5-10 минут
4. Ваш инженер оценивает решение и обсуждает подход

Почему работает:

- Тактильное взаимодействие (запоминается лучше)
- Показывает как человек мыслит
- Генерит дискуссию ("А почему ты не добавил кэш?")
- Фоткается для портфолио и соцсетей
- Можно делать в командах

Реквизит:

- Наборы Lego / магниты / карточки
- Большая доска или поверхность
- Карточки с заданиями

Бюджет: 10-30K (Lego, магниты, печать карточек)

12.3. Кейсы использования геймификации

Теперь посмотрим на реальные примеры (анонимизированные или из открытых источников).

Кейс 1: Яндекс на конференциях

Задача:

Привлечь внимание на крупной конференции, где 100+ стендов.

Решение:

Квест "Путешествие по технологиям Яндекса"

Механика:

- Участник получает карту-буллет с 6 станциями
- Каждая станция = одна технология Яндекса (Поиск, Карты, Такси, Облако, etc.)
- На каждой станции — мини-активность (квиз, демо, игра)
- За прохождение — наклейка на карту
- Собрал все 6 → обменял на эксклюзивный мерч

Результаты:

- 500+ участников квеста (из 2000 посетителей конференции)
- Среднее время у стенда: 25 минут
- 200+ постов в соцсетях с хэштегом кампании
- 150+ качественных лидов (кто дошел до конца и оставил контакт)

Почему сработало:

- Видимый прогресс (карта с наклейками)
 - Ценный финальный приз
 - Каждая станция = отдельная беседа с инженером
 - Gamification loop: начал → увлекся → дошел до конца
-

Кейс 2: Стартап на митапе (бюджет: 15K)

Задача:

Небольшой стартап (команда 15 человек) на локальном митапе. Бюджет минимальный, нужно выделиться.

Решение:

"Bug Bounty на стенде"

Механика:

- На большом мониторе — демо продукта, но в нем специально оставлены баги (UI, логика, опечатки)
- Задание: найди максимум багов за 5 минут
- Чем больше нашел — тем лучше приз
- После — обсуждение с разработчиком: "Как бы ты это исправил?"

Реквизит:

- Ноутбук с демо
- Блокнот для записи багов
- Призы: стикеры, футболки (бюджет 10K)
- Печать флаера с правилами (2K)

Результаты:

- 30+ участников
- Обнаружили 5 реальных багов (бонус!)

- 💬 Глубокие техдискуссии → 8 кандидатов в pipeline
- 📊 Feedback по UX продукта

Почему сработало:

- Минимальный бюджет
- Показывает продукт в реальности
- Разработчики обожают находить баги
- Двойная польза: и маркетинг, и QA

Лайфхак: Оставьте 1-2 очень хитрых бага, чтобы даже сеньоры попотели.

Кейс 3: Корпорация на DevRelCon (бюджет: 500K)

Задача:

Крупная корпорация хочет показать экспертизу на международной DevRel конференции.

Решение:

"DevRel Escape Room"

Механика:

- Мини-квест-комната на стенде (3x3 метра)
- Легенда: "Вы — DevRel команда, нужно запустить продукт за 15 минут"
- 5 этапов: написать код, пройти code review, деплой, инцидент в production, communication с community
- Каждый этап = головоломка/задание
- Команды 3-4 человека
- Записываем на видео → контент для соцсетей (с разрешения)

Реквизит:

- Конструкция квест-комнаты (аренда/постройка: 150K)
- Реквизит для заданий: ноутбуки, планшеты, props (100K)
- Видеооператор (50K)
- Призы (100K)
- Дизайн и продакшн (100K)

Результаты:

- 50 команд (200 участников)
- 15 видео-роликов для соцсетей (виральных: 3)

- 📊 Упоминания в отчетах о конференции
- 💼 Контакты всех участников + данные о прохождении (кто как решал задачи)

Почему сработало:

- Уникальный опыт (никто другой так не делал)
- Командная активность → естественный нетворкинг
- Видео-контент на год вперед
- Показывает экспертизу компании в DevRel

Минус: Высокий бюджет, сложная логистика.

12.4. Метрики эффективности геймификации

Как понять, сработала ли ваша активность?

Метрики вовлечения

Количественные:

- 📈 **Participation rate** = Участники / Посетители стенда
 - Хорошо: > 30%
 - Отлично: > 50%
- ⏳ **Среднее время у стенда**
 - Без геймификации: 1-3 минуты
 - С геймификацией: 10-30 минут
 - Цель: увеличить минимум в 5x
- ⏱ **Repeat visits** = Сколько раз один человек вернулся
 - Хорошо: 20%+ возвращаются
 - Отлично: 40%+ возвращаются

Качественные:

- 💬 **Глубина разговоров**
 - Собирали просто визитки → 🚫
 - Поверхностный small talk → 🟡
 - Обсуждение технологий, проектов → 🟢

- **Качество лидов**

- Relevance: подходят ли под ваши вакансии
- Intent: проявляют ли интерес к компании
- Context: что узнали про человека

Метрики узнаваемости

Social Media:

- Количество UGC (user-generated content): фото, посты, stories
- Mentions и хэштеги
- Reach (органический охват через репосты)

Цель: Минимум 30% участников постят в соцсети

Запоминаемость:

- Brand recall: через неделю после конференции спросите участников "Какие стенды запомнились?"
- Цель: попасть в топ-3 упоминаемых

ROI метрики

Краткосрочные:

- Лиды в pipeline (кол-во + качество)
- Конверсия в созвоны/интервью
- Подписки на блог/рассылку

Долгосрочные:

- Найм (сколько участников стали сотрудниками)
- Партнерства (кто из участников стал партнером/клиентом)
- Рост бренда (изменение awareness метрик)

Формула ROI геймификации:

$$\text{ROI} = (\text{Ценность лидов} + \text{Ценность контента} + \text{Brand lift}) / \text{Затраты}$$

Где:

- Ценность лидов = Количество × Стоимость найма / 10-20 (конверсия)

- Ценность контента = Стоимость создания эквивалентного контента
- Brand lift = Изменение awareness × Средний LTV сотрудника

Пример расчета:

- Затраты: 100К (разработка + реквизит + призы)
- Лиды: 50 качественных контактов
- Контент: 20 видео/фото для соцсетей (эквивалент 50К на создание)
- Нанято: 2 человека за полгода после события

$$\text{ROI} = (50 \times 500\text{K} \text{ (средняя стоимость найма)} / 15 + 50\text{K} + \text{оценка brand lift}) / 100\text{K}$$
$$\text{ROI} \approx 3.2x \text{ (за полгода)}$$

Вывод: Даже консервативная оценка показывает 3х возврат.

12.5. Частые ошибки и как их избежать

Ошибка 1: Слишком сложная механика

Проблема:

Квест из 15 шагов с запутанными правилами. Люди не понимают что делать → уходят.

Как избежать:

- Правила должны объясняться за 30 секунд
- Первый шаг должен быть очевидным
- Визуальные подсказки (стрелки, номера, цвета)
- Протестируйте на 3-5 коллегах перед событием

Правило: Если бабушка не поймет механику — упрощайте.

Ошибка 2: Геймификация ради геймификации

Проблема:

Крутая игра, но она никак не связана с вашей компанией/продуктом. Люди поиграли, получили приз, ушли — и ничего не запомнили.

Как избежать:

- Каждая механика должна коммуницировать что-то о компании

- Призы должны быть брендированными (не просто iPhone, а iPhone + онбординг в вашу платформу)
- В процессе игры — разговор с инженерами, демо продукта

Правило: Геймификация = средство, не цель.

Ошибка 3: Слишком детские механики

Проблема:

Воздушные шарики, раскраски, конкурс на самый смешной мем. Аудитория — сеньор-разработчики. Они чувствуют себя неловко.

Как избежать:

- Помните свою аудиторию: это профессионалы
- Механики должны быть интеллектуальными или техническими
- Если юмор — то профессиональный (code jokes, memes про баги)
- Не “играть в детей”, а “интеллектуальный вызов”

Исключение: Иногда “глупые игры” работают как icebreaker, но они должны быть самоироничными.

Ошибка 4: Перегруженный стенд

Проблема:

10 разных активностей, шум, хаос. Люди не понимают с чего начать → уходят.

Как избежать:

- 1-3 активности максимум
 - Четкое зонирование пространства
 - “Гид” на входе: “Привет, начни с квиза вон там”
 - Управляемый поток (не пускать всех одновременно)
-

Ошибка 5: Забыли про призы

Проблема:

Крутая игра, но приз — ручка с логотипом. Разочарование.

Как избежать:

- Призы должны быть реально ценными

- Лучше мало, но круто, чем много и дешево
- Tier призы: малый (всем участникам), средний (топ-10), большой (победитель)

Что ценят разработчики:

- Годовые подписки на инструменты (JetBrains, GitHub Pro)
 - Качественные периферийные устройства (клавиатуры, наушники)
 - Книги (технические или бизнес)
 - Эксклюзивный доступ к вашему продукту
 - Билеты на другие конференции
-

Ошибка 6: Нет метрик

Проблема:

Вы провели активность, все было весело, но вы не знаете: сработало ли это?

Как избежать:

- Определите метрики до события
- Соберите данные: кто участвовал, сколько времени, что понравилось
- Post-event опрос (короткий, 3-5 вопросов)
- Отслеживайте соцсети (хэштеги, mentions)
- Посчитайте ROI

Инструменты:

- Google Forms для сбора feedback
 - QR-коды для трекинга участия
 - Уникальные промокоды/ссылки для каждой активности
 - Social listening tools (Mention, Brand24)
-

Ключевые выводы главы

1. **Геймификация — не развлечение ради развлечения.** Это инструмент для глубокого вовлечения профессиональной аудитории.
2. **Работает благодаря мозгу.** Используем естественные свойства: любовь к прогрессу, коллекционированию, соревнованию, закрытию циклов.
3. **8 мотиваторов Oktylysis** — ваш framework для дизайна активностей.

4. **Форматы разнообразны:** от простых квизов (бюджет 5K) до escape rooms (бюджет 500K). Выбирайте по возможностям.
 5. **Метрики критичны:** Participation rate, время у стенда, quality of leads, social reach, ROI.
 6. **Частые ошибки:** Сложность, оторванность от бренда, детскость, хаос, плохие призы, отсутствие метрик.
 7. **Главное правило:** Геймификация должна создавать повод для разговора о вашей компании и технологиях, а не заменять его.
-

Чек-лист для внедрения геймификации

Перед мероприятием:

- [] Определена цель (awareness? leads? hiring?)
- [] Выбрана механика под целевую аудиторию
- [] Правила можно объяснить за 30 секунд
- [] Механика связана с брендом/продуктом
- [] Призы реально ценные для аудитории
- [] Протестировано на коллегах
- [] Подготовлены метрики для оценки
- [] Реквизит заказан и доставлен
- [] Команда знает свои роли

Во время:

- [] "Гид" встречает и объясняет правила
- [] Отслеживаем количество участников
- [] Собираем feedback в реальном времени
- [] Следим за соцсетями (хэштеги)
- [] Фиксируем проблемы для improvement

После:

- [] Подсчитаны метрики
 - [] Собрана обратная связь
 - [] Проанализирован ROI
 - [] Составлен отчет с выводами
 - [] Контент из события использован в соцсетях
-

Что дальше?

В следующей главе мы разберем **инструменты DevRel**: от аналитики до автоматизации, включая AI-инструменты, которые меняют профессию.

А пока — начните с малого:

1. Выберите одну механику для следующего митапа/конференции
2. Определите бюджет (даже 5-10К достаточно для старта)
3. Протестируйте на коллегах
4. Измерьте результаты
5. Итерируйте!

Помните: Лучшая геймификация — та, которую участники вспоминают через месяц и рассказывают друзьям.

Статус главы: Черновик v1.0

Объем: ~2,200 слов

Следующая редакция: Добавить больше фото примеров, схемы механик, шаблоны для организаторов

ЧАСТЬ 5. ИНСТРУМЕНТЫ И МЕТРИКИ

Глава 13. Инструменты DevRel

Зачем эта глава: DevRel работает с десятками задач одновременно: контент, события, community, аналитика. Правильные инструменты могут в разы повысить эффективность. Разберем must-have инструменты для каждого направления.

13.1. Аналитика и метрики

Google Analytics (бесплатно)

Для чего:

- Трафик на блог/сайт
- Source tracking (откуда приходят)
- Поведение (что читают, куда кликают)

Setup: 1-2 часа

Social Media Analytics

Встроенные:

- Telegram Stats (для каналов)
- LinkedIn Analytics
- YouTube Analytics

External tools:

- Brand24 (от 50\$/месяц) - мониторинг упоминаний
 - Mention - social listening
-

13.2. Коммуникационные платформы

Для сообществ

Telegram:

- Бесплатно
- Bot API для автоматизации

- Интеграции

Discord:

- Бесплатно
- Боты и интеграции
- Структурированность

Slack:

- Бесплатный tier (ограничения)
- Платный: \$8/user/месяц
- Много интеграций

Для контента

Notion / Confluence:

- База знаний
- Контент-планы
- Collaboration

Google Docs:

- Бесплатно
 - Простая коллаборация
 - Comments и suggestions
-

13.3. Автоматизация

n8n / Zapier

Что автоматизировать:

- Публикация (новая статья на Habr → пост в Telegram)
- Сбор feedback (форма → Google Sheets → Slack notification)
- Social media (кросс-постинг)

Стоимость:

- Zapier: от \$20/месяц
- n8n: бесплатно (self-hosted) или от \$20/месяц

Боты для Telegram

Use cases:

- Welcome message для новых участников
- Авто-модерация (антиспам)
- FAQ бот
- Reminder для events

Создание:

- Bot API (бесплатно)
 - Разработка: 20-100 часов в зависимости от сложности
-

13.4. CRM и управление лидами

Для B2D компаний:

- HubSpot (от \$50/месяц)
- Salesforce (дорого, для крупных)

Для employer branding:

- Google Sheets (простой tracking)
 - Airtable (более структурированно)
 - Notion (all-in-one)
-

Ключевые выводы

- 1. Начните с бесплатных.** Google Analytics, Telegram, Notion бесплатны и покрывают 80% нужд.
 - 2. Автоматизируйте рутину.** Кросс-постинг, уведомления, сбор данных — это для инструментов, не для вас.
 - 3. Не переплачивайте.** Enterprise tools нужны только крупным компаниям.
-

Статус главы: Черновик v1.0

Объем: ~800 слов (~3 страницы)

Следующая редакция: Расширить каждый раздел, добавить сравнительные таблицы, больше конкретных инструментов с ссылками

Глава 14. AI-инструменты в DevRel

Зачем эта глава: AI радикально меняет DevRel. Задачи, которые раньше занимали часы, теперь выполняются за минуты. Но есть подводные камни. Разберем как использовать AI эффективно и этично.

Disclaimer: Эта глава может быстро устареть. AI развивается стремительно. Фокусируемся на принципах, не на конкретных инструментах.

14.1. Генерация контента

ChatGPT, Claude, Ollama для текстов

Use cases:

1. Brainstorming идей

- “Дай 10 идей для статей про Kubernetes”
- “Какие проблемы решают разработчики с микросервисами?”

2. Структурирование

- “Создай outline для статьи ‘Миграция на Go’”
- “Какие разделы должны быть в титуле?”

3. Черновик

- “Напиши введение для статьи про...”
- “Объясни концепцию X простым языком”

4. Редактура

- “Улучши этот параграф”
- “Сделай текст более engaging”

Что AI делает хорошо:

- Генерация идей (brainstorming)
- Структурирование мыслей
- Редактура и улучшение формулировок
- Перевод и адаптация

Что AI делает плохо:

- ✗ Глубокая техническая экспертиза (галлюцинации)
- ✗ Уникальные инсайты (выдает "average" мнение)
- ✗ Актуальность (данные могут быть устаревшими)
- ✗ Код для production (работает, но не best practices)

Правило:

AI = ассистент, не replacement. Генерирует черновик, вы дорабатываете и проверяете.

Примеры использования в этой книге

Кейс: Создание книги "База DevRel"

Что делал AI:

- Структурирование контента из Telegram каналов
- Фильтрация релевантности
- Генерация черновиков глав
- Редактура и форматирование

Что делал человек:

- Определение структуры книги
- Финальная редактура
- Добавление личного опыта
- Проверка фактов

Экономия времени: ~60-70%

Качество: Сопоставимо с написанием вручную (после редактуры)

14.2. Анализ аудитории и sentiment

AI для анализа community

Use cases:

1. Sentiment analysis

- Анализ комментариев (positive vs negative)
- Tracking настроений в сообществе
- Early warning (негатив растет)

2. Topic extraction

- О чем говорят в чате/форуме
- Trending topics
- Emerging issues

3. User segmentation

- Кто активные участники
- Кто новички
- Кто эксперты

Инструменты:

- Ollama (бесплатно, локально) + custom скрипты
- ChatGPT API (платно, но проще)
- Специализированные (Brandwatch, Sprinklr - дорого)

Практический пример из этого проекта:

Использовали Ollama qwen2.5:7b для анализа 2,876 сообщений из DevRel каналов:

- Извлекли 107 советов
- Категоризировали по темам
- Отфильтровали релевантные для книги

Экономия: недели ручного анализа → часы с AI

14.3. Автоматизация рутинь

Email и reports

Что автоматизировать:

- Thank you emails после митапов
- Недельные/месячные отчеты (draft)
- Follow-up sequences

Инструменты:

- ChatGPT для генерации писем
- Zapier/n8n для автоматизации отправки

Social media

Content repurposing:

- Статья → Twitter thread (AI генерирует)
- Доклад → LinkedIn пост
- Long-form → Short-form

Scheduling:

- Buffer, Hootsuite для планирования
 - AI для генерации постов из контента
-

14.4. Этика использования AI в DevRel

Что допустимо

Использовать AI как ассистента:

- Черновики
- Brainstorming
- Редактура
- Анализ данных

С прозрачностью:

- Указывайте если контент AI-generated (в каких-то контекстах)
- Особенно для кода (люди должны знать)

Что недопустимо

Выдавать AI контент за свой без проверки:

- AI может галлюцинировать факты
- Код может не работать или быть небезопасным
- Устаревшая информация

Спамить AI-generated контент:

- 10 посредственных AI статей хуже чем 1 качественная human
- Аудитория чувствует разницу

Использовать для манипуляции:

- Fake reviews
- Astroturfing (искусственное создание buzz)

Принцип

AI должен усиливать человека, не заменять.

DevRel про отношения и trust. Это невозможно автоматизировать полностью.

Ключевые выводы

1. **AI = game changer для DevRel.** Экономия 60-70% времени на рутинных задачах.
2. **Лучше всего для:** Brainstorming, черновики, редактура, анализ данных.
3. **Хуже всего для:** Глубокая экспертиза, уникальные инсайты, relationship building.
4. **Этика критична.** Будьте прозрачны, проверяйте факты, не спамьте.
5. **Эта глава устареет.** AI меняется быстро. Следите за новыми инструментами.

Статус главы: Черновик v1.0

Объем: ~1,100 слов (~4-5 страниц)

Следующая редакция: Добавить больше конкретных инструментов, промпты для типичных задач, кейсы применения AI

Глава 15. Метрики эффективности DevRel

Зачем эта глава: “Покажи мне ROI от DevRel” — один из самых частых вопросов от руководства. В этой главе разберем как измерять эффективность DevRel, какие метрики действительно важны, как доказать ценность работы, и как не утонуть в ванильных метриках.

15.1. Проблема измерения DevRel

Почему сложно измерять DevRel?

Вызов 1: Отложенный эффект

DevRel активности дают результат не сразу:

- Доклад на конференции → человек узнал о компании → через 6 месяцев откликнулся на вакансию → через год нанят
- Статья на Habr → читатель начал следить за компанией → через год стал клиентом

Как доказать attribution?

Что именно доклад привел к найму, а не реклама или сарафан?

Вызов 2: Качество > Количество

Легко измерить:

- Сколько докладов сделали
- Сколько статей написали
- Сколько просмотров набрали

Сложно измерить:

- Насколько изменилось восприятие бренда?
- Насколько выросло доверие?
- Насколько улучшилось качество кандидатов?

Вызов 3: Много факторов влияния

На найм/retention влияют:

- DevRel (awareness, бренд)
- HR (процессы, культура)
- Product (успех продукта)

- Market (состояние рынка труда)
- Зарплаты (конкурентность)

Как выделить чистый эффект DevRel? **Невозможно через A/B тест** (нет контрольной группы).

Что работает: multi-level метрики

Вместо попытки измерить "чистый ROI DevRel", измеряйте на **разных уровнях**:

Level 1: Activity metrics (что делаем)

Level 2: Engagement metrics (как реагируют)

Level 3: Perception metrics (как воспринимают)

Level 4: Business metrics (как влияет на бизнес)

Связь между уровнями не 1-to-1, но есть корреляция.

15.2. Level 1: Activity Metrics (что делаем)

Это самые простые метрики — **считаем активности**.

Контент

Что считать:

- Количество докладов (поданных / принятых / сделанных)
- Количество статей (написанных / опубликованных)
- Количество видео / туториалов
- Количество podcast эпизодов

Почему важно:

Baseline для планирования. Если в прошлом году сделали 10 докладов, в этом можем запланировать 15.

Почему недостаточно:

10 плохих докладов хуже 5 отличных. Количество ничего не говорит о качестве и impact.

События

Что считать:

- Количество митапов (организованных)
- Количество конференций (с участием)
- Количество воркшопов / хакатонов

Разбивка:

- Собственные события (вы организатор)
- Участие (вы спикер/спонсор)

Сообщество

Что считать:

- Размер сообщества (участники в чате/канале)
 - Рост (новые участники в месяц)
 - Retention (сколько остаются активными 3+ месяца)
-

15.3. Level 2: Engagement Metrics (как реагируют)

Теперь смотрим **как аудитория реагирует** на активности.

Метрики контента

Для статей:

- **Просмотры** (views)
 - Unique visitors (уникальные)
 - Total views (с повторными)
- **Читаемость** (scroll depth, time on page)
 - Цель: > 60% дочитывают до конца
- **Engagement** (likes, comments, shares)
 - Engagement rate = (Likes + Comments + Shares) / Views
 - Хорошо: > 5%, Отлично: > 10%
- **Трафик** на сайт с статьи
 - Сколько перешли на ваш сайт/продукт

Для докладов:

- **Attendance** (сколько пришли на доклад)
 - Full room vs half-empty
- **Questions** (сколько вопросов после доклада)

- Много вопросов = интересно + понятно объяснили
- **Feedback** от организаторов
 - Оценка доклада (1-5 или 1-10)
 - Комментарии участников
- **Видео** (если выкладывают)
 - Просмотры на YouTube
 - Comments, likes

Метрики событий

Для митапов:

- **Registration → Show-up rate**
 - Зарегистрировалось 100 → Пришло 65 → SR = 65%
 - Хорошо: > 60%, Отлично: > 70%
- **NPS события**
 - “Порекомендуете ли митап коллеге?” (0-10)
 - Цель: NPS > 30 (хорошо), > 50 (отлично)
- **Repeat attendance**
 - Сколько участников возвращаются на следующие митапы
 - Цель: > 30% regulars
- **Quality of networking**
 - Субъективно, но важно
 - Люди активно общались? Или сидели по углам?

Метрики сообщества

Активность:

- **Daily Active Users (DAU) или Monthly Active Users (MAU)**
 - Сколько пишут хотя бы раз в день/месяц
- **Messages per day**
 - Среднее количество сообщений
 - Тренд (растет? падает?)

- **90-9-1 ratio**

- Идеально: 85-12-3 (больше активных)

Качество:

- **Average response time** на вопросы

- Как быстро community помогает новичкам
 - Цель: < 30 минут

- **Question resolution rate**

- Сколько вопросов получают ответ
 - Цель: > 80%

- **Sentiment** (позитив vs негатив)

- Можно через AI анализировать (как мы делали в этом проекте!)
 - Цель: > 70% позитива
-

15.4. Level 3: Perception Metrics (как воспринимают)

Это сложнее измерить, но **критично важно**.

Brand Awareness (осведомленность)

Как измерять:

1. Surveys (опросы целевой аудитории)

Вопросы:

- "Назовите 3 IT-компании в России, которые вы знаете" (top-of-mind)
- "Слышали ли вы о компании X?" (awareness)
- "Что вы знаете о компании X?" (depth of awareness)

Метрики:

- **Top-of-mind awareness** (% кто назвал вас первыми)
- **Aided awareness** (% кто знает после подсказки)
- **Brand attributes** (с чем ассоциируют: "innovative", "good employer", etc.)

2. Search volume

- Google Trends: "Название компании" + "вакансии"

- Яндекс.Wordstat: рост запросов

3. Social listening

- Упоминания бренда (с sentiment analysis)
- Share of voice (% упоминаний vs конкуренты)
- Hashtag performance

Инструменты:

- Brand24, Mention (платные)
- Google Alerts (бесплатно, но basic)
- IQBuzz, YouScan (для РФ рынка)

Brand Consideration (рассматривают ли как employer)

Как измерять:

1. Application source tracking

При отклике спрашивайте: "Откуда узнали о вакансии?"

Опции:

- Habr
- LinkedIn / HeadHunter
- Рекомендация знакомого
- Доклад на конференции
- Статья в блоге
- Telegram канал
- Другое

Что смотреть:

- Organic applications (без платной рекламы)
 - Рост organic = работает DevRel
- Source attribution
 - "Статья в блоге" растет → контент работает

2. Employer brand surveys

Вопросы для кандидатов:

- “Почему рассматриваете нашу компанию?”
- “Что знаете о нашей технической культуре?”
- “Где слышали о нас?”

Качественная метрика:

Глубина знания. Если кандидат говорит “Читал вашу статью про микросервисы, впечатлило” — DevRel сработал.

Reputation metrics

Внешние рейтинги:

- Glassdoor rating (company rating, CEO approval)
- Хабр Карьера (Best IT Employers)
- [vc.ru](#) рейтинги
- LinkedIn company rating

Цель:

Положительная динамика (было 3.5 → стало 4.2 за год).

Internal NPS (eNPS):

- “Порекомендуешь ли компанию другу как место работы?” (0-10)
- Формула: % Promoters (9-10) - % Detractors (0-6) = eNPS
- Хорошо: > 20, Отлично: > 50

Корреляция:

eNPS часто коррелирует с external reputation. Если внутри недовольны → внешний бренд тоже пострадает.

15.5. Level 4: Business Metrics (impact на бизнес)

Это то, что волнует руководство: **как DevRel влияет на бизнес-результаты?**

Hiring Metrics

Primary:

- **Time to hire** (дни от открытия вакансии до оффера)
 - Сильный бренд → быстрее найм
 - Baseline → Target (было 60 дней → стало 45)
- **Cost per hire** (стоимость найма одного человека)

- Organic leads дешевле чем платная реклама
- Baseline → Target (было 500K → стало 350K)
- **Quality of hire** (сложно измерить, но критично)
 - Offer acceptance rate (сколько принимают offers)
 - Onboarding success (как быстро выходят на productivity)
 - Retention (остаются ли долго)
- **Organic applications** (% откликов без платной рекламы)
 - Рост organic = эффект DevRel
 - Baseline → Target (было 20% → стало 40%)

Secondary:

- **Referrals** (рекомендации от сотрудников)
 - Растут когда люди гордятся компанией
 - % от общего найма
- **Passive candidate activation**
 - Сколько "не ищут работу" но откликаются на ваши вакансии
 - Сильный бренд → люди сами хотят к вам

Retention Metrics

Primary:

- **Employee retention rate**
 - % сотрудников, остающихся 1+ год, 2+ года
 - Сильная культура → выше retention
- **Voluntary turnover rate**
 - % уходящих по собственному желанию
 - Цель: снижать

Корреляция с DevRel:

DevRel не прямо влияет на retention, но:

- Сильный external бренд → люди гордятся компанией → меньше уходят
- Internal community (DevRel тоже организует) → крепче связи → меньше turnover

Product Metrics (для B2D компаний)

Если ваш продукт = для разработчиков:

Adoption:

- **Signups** (регистрации)
 - Источник: органика vs платная реклама
 - DevRel attribution (UTM-метки, promo codes)
- **Active users** (DAU, MAU)
 - Рост активных пользователей
 - Retention cohorts

Engagement:

- **API calls / usage**
- **Feature adoption** (какие фичи используют)
- **Time to first API call** (как быстро начинают использовать после регистрации)

Community contribution:

- **GitHub stars / forks** (для open-source)
 - **Contributors** (сколько контрибьютят)
 - **Community support** (сколько вопросов отвечают сами пользователи)
-

15.6. Отчетность: как показывать результаты

Что волнует разных стейкхолдеров

CEO/Founder:

- Brand awareness (знают ли нас?)
- Talent acquisition (легче ли нанимать?)
- Market positioning (лидеры мнений или нет?)

СТО:

- Качество кандидатов (приходят ли strong?)
- Technical reputation (считают ли нас экспертами?)
- Community feedback (что говорят о наших технологиях?)

CMO:

- Reach и engagement
- Content performance
- Social media growth
- Share of voice vs competitors

CFO:

- ROI (прямой или косвенный)
- Cost per hire (сравнение с другими каналами)
- Budget efficiency

Формат отчета

Frequency: Квартальные отчеты (детальные) + monthly updates (краткие)

Структура квартального отчета:

```
# DevRel Q1 2025: Отчет

## Executive Summary
- 2-3 ключевых достижения
- 2-3 ключевые метрики
- 1 главный вызов

## Activities (что сделали)
- 12 докладов (список с конференциями)
- 8 статей (ссылки, просмотры)
- 3 митапа (участники, feedback)
- Рост Telegram канала: +500 подписчиков

## Engagement (как реагировали)
- Всего охват: 50,000 просмотров/участников
- Engagement rate: 7.5% (выше baseline 5%)
- NPS митапов: 52 (отлично)

## Perception (как воспринимают)
- Top-of-mind: вошли в топ-10 (было топ-15)
- eNPS: вырос с 25 до 32
- Glassdoor: 4.2 (было 3.9)

## Business Impact (влияние на бизнес)
```

- Organic applications: +40% (было 20%, стало 28%)
- Time to hire: -10 дней (было 55, стало 45)
- Cost per hire: -15% (экономия 2 млн на 20 наймов)
- Hired: 3 человека прямо attributed DevRel активностям

Budget

- Потрачено: 1.2 млн (план: 1.5 млн)
- ROI: ~2.5x (консервативная оценка)

Next Quarter Plans

- Запланировано 15 докладов
- Запуск podcast series
- 2 крупных митапа

Ключевые принципы:

- Начинайте с главного (executive summary)
- Цифры > слов (quantify everything)
- Тренды > абсолютные значения (рост +40% лучше чем "28% organic")
- Attribution где возможно (3 найма из DevRel leads)
- Честно про challenges (не только success story)

Dashboard для DevRel

Создайте дашборд который обновляется автоматически.

Инструменты:

- Google Data Studio (бесплатно, интеграции)
- Metabase (open-source, self-hosted)
- Tableau (если есть корпоративная лицензия)

Что показывать:

Блок 1: Активности (month-to-date)

- Докладов: 3 / 4 planned
- Статей: 2 / 3 planned
- Митапов: 1 / 1 planned

Блок 2: Reach & Engagement

- Контент: 15,000 просмотров ($\uparrow 25\%$ vs last month)

- Engagement rate: 8.2% (\uparrow 1.5% vs last month)
- Telegram: 2,500 подписчиков (\uparrow 150 vs last month)

Блок 3: Business Impact

- Organic applications: 35% (\uparrow 7% vs last quarter)
- Time to hire: 42 дня (\downarrow 8 дней vs last quarter)
- eNPS: 34 (\uparrow 2 vs last quarter)

Обновление: Weekly или monthly.

Доступ: Для руководства и DevRel команды.

15.7. ROI DevRel: как посчитать

Формула ROI (упрощенная)

$$\text{ROI} = (\text{Ценность результатов} - \text{Затраты}) / \text{Затраты} \times 100\%$$

Где:

Ценность = Экономия на найме + Brand lift + Product adoption (если B2D)

Затраты = Зарплаты + Бюджет на активности

Пример расчета (годовой)

Затраты:

- DevRel зарплата: 3 млн/год (1 FTE)
- Бюджет на активности: 2 млн
- Время инженеров: 500 часов \times 5K/час = 2.5 млн
- **Итого: 7.5 млн/год**

Результаты:

Экономия на найме:

- Нанято 20 человек за год
- Organic applications: 40% (8 человек пришли через DevRel)
- Cost per hire через рекрутинг: 500K
- Cost per hire через DevRel: 300K (cheaper)
- Экономия: $8 \times (500K - 300K) = \mathbf{1.6 \text{ млн}}$

Brand lift (оценочно):

- Выросли в рейтинге с топ-15 до топ-10
- eNPS +10 пунктов
- Оценка влияния на retention: 2% меньше turnover
- 200 человек \times 2% \times 500K (cost of replacement) = **2 млн**

Контент (долгосрочная ценность):

- 20 статей на Habr (эквивалент 50K каждая на создание) = 1 млн
- 15 видео докладов (evergreen content) = 0.5 млн
- **Итого: 1.5 млн**

ROI calculation:

Ценность = 1.6 + 2 + 1.5 = 5.1 млн

Затраты = 7.5 млн

ROI = (5.1 - 7.5) / 7.5 = -32%

Стоп! Отрицательный ROI?

Не совсем. Мы не учли:

- Long-term эффект (контент работает годами)
- Compound effect (бренд растет экспоненциально, не линейно)
- Indirect benefits (partnerships, media coverage, thought leadership)

Более честный подход:

ROI DevRel — это **инвестиция в долгосрочный рост**, как R&D. Окупаемость 2-3 года, не 1 год.

Консервативная модель ROI

Год 1: ROI может быть отрицательным или около нуля (строим фундамент)

Год 2: ROI 1-2x (начинают работать compound effects)

Год 3+: ROI 3-5x (экспоненциальный рост)

Аналогия:

Инвестиции в бренд как инвестиции в R&D — не ждите немедленной отдачи, но долгосрочный эффект огромен.

15.8. OKR для DevRel: как ставить цели

OKR (Objectives and Key Results) — framework для постановки целей.

Структура

Objective: Амбициозная качественная цель

Key Results: 3-5 измеримых результатов, которые показывают достижение цели

Примеры OKR для DevRel

Objective 1: Стать топ-10 IT employer в России

KR1: Top-of-mind awareness 15% → 25% (опрос разработчиков)

KR2: eNPS 25 → 40

KR3: Попасть в топ-10 рейтинга Хабр Карьера

KR4: 100+ докладов/статей за год от команды

Objective 2: Построить крупнейшее Go-сообщество в России

KR1: 1,000 участников в Telegram чате (сейчас 200)

KR2: 12 митапов за год (раз в месяц)

KR3: 30% repeat attendance на митапах

KR4: 20+ активных контрибьюторов в core team

Objective 3: Developer adoption нашего API

KR1: 10,000 регистраций разработчиков (сейчас 2,000)

KR2: 30% activation rate (делают хотя бы 1 API call)

KR3: 500+ monthly active developers

KR4: Community NPS 40+

Как ставить хорошие OKR

Objective должен быть:

- Амбициозным (stretch goal, а не safe)
- Качественным (inspiring)
- Aligned с бизнес-целями компании

Key Results должны быть:

- Измеримыми (numbers!)
- Достижимыми (challenging but realistic)
- Time-bound (квартал или год)
- Outcome-based (результат, не активность)

Плохой KR:

“Сделать 20 докладов” (это активность, не результат)

Хороший KR:

“Достичь 50,000 просмотров докладов и 500 новых подписчиков из аудитории” (результат от докладов)

Ключевые выводы главы

1. **DevRel сложно измерять напрямую.** Но можно через 4 уровня: Activities → Engagement → Perception → Business Impact.
 2. **Не зацикливайтесь на ванильных метриках.** “10 докладов” ничего не значит без контекста engagement и impact.
 3. **Важны тренды, не абсолютные значения.** Рост +30% лучше чем “30% в абсолюте”.
 4. **ROI DevRel – долгосрочный.** Год 1 может быть нулевым или отрицательным. Год 3+ дает 3-5x.
 5. **Attribution сложен.** Невозможно доказать что “именно этот доклад привел к найму”. Но корреляцию видно.
 6. **Используйте OKR.** Ambitious objectives + measurable key results = clarity для всех.
 7. **Дашборд > Презентации.** Автоматизируйте отчетность, чтобы метрики всегда были актуальны.
-

Практическое задание

Для DevRel-ов:

Создайте свой measurement framework:

1. Выберите 2-3 метрики с каждого уровня:

- Level 1 (Activity): _____
- Level 2 (Engagement): _____
- Level 3 (Perception): _____
- Level 4 (Business): _____

2. Определите baseline (где сейчас):

- Соберите данные за последние 3 месяца
- Это ваша starting point

3. Поставьте targets (куда хотите прийти):

- Realistic (через 3 месяца)
- Ambitious (через год)

4. Настройте tracking:

- Где данные? (Google Analytics, CRM, опросы?)
- Как часто обновляете? (weekly, monthly?)
- Кто отвечает за сбор?

5. Создайте простой дашборд:

- Google Sheets с графиками (minimum viable)
- Или Data Studio (если есть время)

Для руководителей:

При оценке DevRel:

Не смотрите только на:

- ✗ Количество активностей ("20 докладов!")
- ✗ Ванильные метрики ("100K просмотров!")
- ✗ Краткосрочный ROI (первый год)

Смотрите на:

- ✅ Тренды (растет ли awareness? eNPS? organic applications?)
 - ✅ Quality (глубина engagement, качество лидов)
 - ✅ Долгосрочный impact (year-over-year comparison)
 - ✅ Alignment с бизнес-целями (DevRel работает на приоритеты компании?)
-

Чек-лист главы 15

Метрики настроены:

- [] Выбраны 8-12 ключевых метрик (2-3 с каждого уровня)
- [] Определен baseline (текущее состояние)
- [] Поставлены targets (куда идем)
- [] Настроен tracking (откуда данные, как часто)

Отчетность:

- [] Создан дашборд (автоматический или manual)

- [] Определена frequency (когда репортим)
- [] Понятно кому репортим (stakeholders)
- [] Формат отчета согласован

OKR:

- [] Поставлены квартальные/годовые OKR
 - [] Aligned с бизнес-целями
 - [] Измеримые и ambitious
 - [] Shared с командой и руководством
-

Что дальше?

Мы разобрали основы DevRel (главы 1-2), бренд работодателя (главы 4-5), community и события (главы 7, 11-12), и метрики (глава 15).

В следующих главах разберем **кейсы**: как DevRel работает в реальных российских и международных компаниях, что работает, что нет, и какие уроки можно извлечь.

А пока — настройте метрики:

1. Выберите ваши ключевые метрики
2. Соберите baseline данные
3. Создайте простой tracking (хотя бы в Excel)
4. Начните репортить регулярно

Помните: Что измеряется — то улучшается. Без метрик DevRel — это shot in the dark.

Статус главы: Черновик v1.0

Объем: ~2,000 слов (~8 страниц)

Следующая редакция: Добавить больше примеров дашбордов, детальнее про инструменты аналитики, кейс-стади измерения ROI

ЧАСТЬ 6. ПРАКТИКА И КЕЙСЫ

Глава 16. Кейсы российских компаний

Зачем эта глава: Теория без практики мертва. Посмотрим как DevRel работает в реальных российских IT-компаниях. Что у них получилось, что нет, какие уроки можно извлечь.

Note: Кейсы основаны на публичной информации и открытых источниках.

16.1. Яндекс: DevRel в крупной корпорации

Масштаб

- 20,000+ сотрудников (из них ~15,000 инженеров)
- Множество продуктов и технологий
- Активный DevRel с 2010-х годов

Что делают

Конференции:

- Яндекс.Фронтенд (frontend)
- Я.Субботник (backend, инфраструктура)
- Mobius (mobile)
- Регулярность: 2-4 крупных события в год

Митапы:

- Регулярные митапы по технологиям
- В офисах Яндекса (Москва, Питер, региональные)
- Frequency: несколько в месяц

Контент:

- Habr корпблог (регулярные публикации)
- YouTube канал (записи докладов)
- Подкасты
- Технический блог на сайте

Сообщество:

- Telegram каналы

- GitHub (open-source проекты)

Open Source:

- ClickHouse (самый известный)
- YDB
- Множество библиотек и инструментов

Результаты

Positive:

- Один из топ-3 tech employers в России
- Strong tech brand
- Легче нанимать (люди мечтают работать в Яндексе)
- Thought leadership (Яндекс = эксперты в high-load, ML, maps)

Уроки

1. **Масштаб требует структуры.** У Яндекса скорее всего 10-20+ людей в DevRel (formal и informal).
 2. **Open Source = мощный инструмент.** ClickHouse дал огромный boost репутации.
 3. **Регулярность критична.** Не разовые события, а систематическая работа годами.
-

16.2. VK: Экосистема для developers

Подход

Фокус: VK — платформа, нужны external разработчики для создания приложений.

Что делают:

VK Developers Platform:

- API и SDK
- Документация
- Developer portal

События:

- VK Hackathon (регулярные хакатоны)
- Митапы для разработчиков приложений

- Конференции (VK Tech)

Сообщество:

- VK Developers community
- Поддержка в чате
- Forum для вопросов

Результаты

- Тысячи приложений и игр на платформе VK
- Active developer community
- Ecosystem вокруг продуктов

Уроки

1. Для B2D компаний DevRel критичен. Без developer community платформа мертва.
 2. Инвестиции в DX окупаются. Хорошая документация и инструменты → больше разработчиков используют.
 3. Хакатоны работают. Быстрый способ создать buzz и найти таланты.
-

16.3. Другие компании: краткие примеры

JetBrains

Фокус: Инструменты для разработчиков (IDEs).

DevRel подход:

- Конференции (KotlinConf, JetBrains DevDays)
- Strong presence на всех крупных конференциях
- Technical content (blogs, videos)
- Community support (форумы, issue tracking)

Урок:

Когда ваш продукт = для разработчиков, DevRel = ваш главный канал маркетинга.

Ozon

Фокус: E-commerce, employer branding.

Что делают:

- Tech митапы
- Habr блог
- Участие в конференциях
- Open source проекты

Урок:

Даже для B2C компании tech brand важен для привлечения талантов.

16.4. Стартапы: DevRel с нуля

Кейс: Условный стартап на 50 человек

Ситуация:

Стартап, команда 50 разработчиков, хотят строить tech brand.

Подход:

Minimal DevRel:

- 0.5 FTE DevRel (кто-то part-time)
- Telegram канал
- 3-4 статьи на Habr в год
- Участие в митапах (не организация своих)

Бюджет: 500К-1 млн/год

Результаты через год:

- Небольшой рост awareness
- 2-3 найма attributed DevRel
- Foundation для масштабирования

Урок:

Стартапу не нужен full-time DevRel. Начните с minimal и масштабируйте по мере роста.

Ключевые выводы

- 1. Крупные компании инвестируют серьезно.** Яндекс, VK — это миллионы в год и команды 10-20+ человек.

2. Для B2D DevRel критичен. VK, JetBrains — их продукт зависит от developer community.
3. Для employer branding DevRel масштабируется с размером. 50 человек → minimal, 200+ → full program.
4. Open Source дает huge boost. ClickHouse от Яндекса — пример.
5. Регулярность > Разовые акции. Системная работа годами, не one-off events.

Статус главы: Черновик v1.0

Объем: ~1,200 слов (~5 страниц)

Следующая редакция: Нужен Deep Research для детальных кейсов, интервью с DevRel-ами из этих компаний, больше цифр и метрик

Глава 17. Международный опыт DevRel

Зачем эта глава: DevRel зародился на Западе, и там накоплен огромный опыт. Что можно адаптировать для России и СНГ? Что не работает из-за культурных различий? Разберем best practices от лидеров индустрии.

17.1. Google Developer Relations

Масштаб

- Сотни DevRel-специалистов по всему миру
- Бюджеты в десятки миллионов долларов
- Ecosystem подход (Android, Cloud, Web, ML)

Что делают

Developer Programs:

- Google Developer Experts (GDE) — программа амбассадоров
- Google Developer Groups (GDG) — локальные сообщества по всему миру
- Google Summer of Code — программа для студентов

События:

- Google I/O (крупнейшая developer конференция)
- Региональные DevFests
- Study Jams и воркшопы

Контент:

- Codelabs (интерактивные туториалы)
- YouTube каналы (Google Developers)
- Блоги и case studies

Что можно адаптировать для РФ

GDE-подобная программа:

Найдите 10-20 внешних экспертов, дайте им ресурсы (доступ к продуктам, мерч, visibility), они продвигают ваши технологии в сообществе.

Codelabs формат:

Интерактивные туториалы step-by-step. Работает отлично.

Local communities support:

Помогайте локальным сообществам (площадки, спикеры, мерч) вместо того чтобы создавать свои с нуля.

Что не работает в РФ:

Масштаб бюджетов (у нас нет таких ресурсов). Нужно адаптировать под реальность.

17.2. Microsoft Developer Advocates

Подход

Фокус на cloud (Azure) + инструменты (VS Code, GitHub).

Что делают:

- Cloud Advocates (technical evangelists)
- Microsoft Learn (образовательная платформа)
- GitHub Community (после acquisition)

Особенность:

Strong focus на Open Source (изменение стратегии с 2015).

Уроки

1. **От “закрытости” к open source** — Microsoft показал как можно изменить reputation через DevRel и OSS.
 2. **Developer education as business model.** Microsoft Learn — бесплатная обучающая платформа, которая приводит к adoption Azure.
-

17.3. Что адаптировать, что нет?

Культурные различия

Западный DevRel:

- Много travelling (международные конференции)
- English-first
- Strong focus на online (большие расстояния)

Российский DevRel:

- Локально (Москва, Питер, региональные центры)
- Русский язык приоритет
- Offline важнее (любим личные встречи)

Что работает универсально

- Quality content (везде ценят)
- Честность и transparency
- Community building
- Open Source
- Metrics-driven approach

Что адаптировать

Ambassador programs:

Google GDE → адаптировать под РФ реалии (меньше benefits, но local visibility)

Education platforms:

Microsoft Learn → можно создать lite version на русском

Community support:

Google GDG → поддержка локальных tech communities в регионах

Статус главы: Черновик v1.0

Объем: ~900 слов (~4 страницы)

Следующая редакция: Нужен Deep Research для детальных кейсов Google, Microsoft, GitHub, Stripe, Twilio

Глава 18. Частые ошибки начинающих DevRel и как их избежать

Зачем эта глава: Учиться на чужих ошибках дешевле чем на своих. Эта глава — квинтэссенция болезненного опыта DevRel-сообщества. Антипаттерны, провалы, и как не повторить.

18.1. Топ-10 ошибок начинающих DevRel

Ошибка 1: Фокус на количестве, не качестве

Что делают:

“Сделаем 50 докладов в год!”

Проблема:

50 посредственных докладов хуже 10 отличных. Выгорание команды, низкое качество, плохой feedback.

Как избежать:

- Ставьте realistic goals (10-15 качественных докладов)
 - Фокус на impact, не на quantity
 - Лучше меньше, да лучше
-

Ошибка 2: Игнорирование метрик

Что делают:

Делают активности, но не измеряют результаты.

Проблема:

Не знаете что работает, что нет. Не можете доказать value руководству. Бюджет могут урезать.

Как избежать:

- См. Главу 15: настройте метрики с первого дня
 - Измеряйте все (even если кажется сложным)
 - Регулярно репортите
-

Ошибка 3: Продающий тон в контенте

Что делают:

"Наш API самый быстрый! Регистрируйтесь сейчас и получите скидку!"

Проблема:

Разработчики ненавидят продажи. Instant turn-off.

Как избежать:

- Образовательный контент, не продающий
 - Помогайте решать проблемы
 - Упоминайте продукт в контексте, не как главную цель
-

Ошибка 4: Работать в вакууме

Что делают:

DevRel делает активности сам, не координируясь с HR, Product, Marketing.

Проблема:

Дублирование усилий, противоречивые месседжи, конфликты.

Как избежать:

- Регулярные синки с HR, Marketing, Product
 - Aligned goals и KPIs
 - Shared roadmap
-

Ошибка 5: Выгорание амбассадоров

Что делают:

Находят 2-3 active спикеров и нагружают их постоянно.

Проблема:

Амбассадоры выгорают, сливаются, остаются обиды.

Как избежать:

- Выращивайте пул (10-20 человек minimum)
 - Ротация (не одни и те же каждый раз)
 - Официально выделяйте время (в KPI и Jira)
 - Благодарите и reward
-

Ошибка 6: Забыли про internal audience

Что делают:

Весь фокус на external (конференции, статьи, сообщество).

Проблема:

Внутренние сотрудники не знают что делает DevRel, не вовлечены, не поддерживают.

Как избежать:

- Internal tech talks и митапы
 - Внутренние сообщества (knowledge sharing)
 - Communication про DevRel успехи внутри
 - Вовлечение инженеров в external активности
-

Ошибка 7: Плохой follow-up

Что делают:

Провели митап, все разошлись, тишина.

Проблема:

Momentum потерян, люди забыли, ROI низкий.

Как избежать:

- Email next day
 - Видео/материалы через 3-5 дней
 - Анонс следующего события
 - Personal follow-up с ключевыми людьми
-

Ошибка 8: Не слушать feedback

Что делают:

Собирают feedback, но не доносят до команды или игнорируют.

Проблема:

Сообщество чувствует что их не слышат → перестают давать feedback → DA теряет credibility.

Как избежать:

- Структурируйте и приоритизируйте feedback
 - Регулярно доносите до Product/Engineering
 - Close the loop (сообщите что изменили на основе feedback)
-

Ошибка 9: Нереалистичные ожидания от руководства

Что делают:

Обещают "100 наймов через DevRel за квартал!"

Проблема:

DevRel — долгосрочная инвестиция. Immediate ROI не будет. Разочарование руководства → урезание бюджета.

Как избежать:

- Устанавливайте realistic expectations (результаты через 6-12 месяцев)
 - Educate руководство (что такое DevRel, как работает)
 - Показывайте leading indicators (awareness, engagement), не только lagging (hires)
-

Ошибка 10: Пытаться делать все

Что делают:

Хотят быть везде: конференции, митапы, блог, Telegram, Discord, подкасты, видео, хакатоны...

Проблема:

Burn out. Все плохо вместо чего-то хорошо.

Как избежать:

- Выберите 2-3 primary каналов
 - Focus > Coverage
 - Делегируйте и автоматизируйте
 - Откажитесь от низкоэффективного
-

18.2. Антипаттерны в community building

Антипаттерн 1: "Построй и они придут"

Что думают:

Создадим чат/канал → люди сами придут и будут активными.

Реальность:

Мертвый чат. Никто не пишет.

Решение:

- Seeding (ядро генерит контент)
- Active invitation (привлекайте лично)

- Regular activity (постоянная активность)

Антипаттерн 2: Zero moderation

Что думают:

“Мы свободное сообщество, никаких правил!”

Реальность:

Хаос, троллинг, токсичность, люди уходят.

Решение:

- Четкие правила (даже если минимальные)
- Активная модерация
- Быстрая реакция на проблемы

Антипаттерн 3: Over-moderation

Что делают:

Удаляют любую критику, банят за малейшее нарушение.

Проблема:

Сообщество чувствует censorship, атмосфера напряженная.

Решение:

- Баланс (модерировать нужно, но не душить)
- Transparency (объясните почему удалили)
- Allow конструктивную критику

18.3. Уроки из неудачных кейсов

Кейс 1: DevRel при токсичной культуре

Ситуация:

Компания с высоким turnover и плохим Glassdoor рейтингом запускает активный DevRel.

Что произошло:

DevRel создал visibility → больше людей узнали → но also больше увидели негативные отзывы
→ backfire эффект.

Урок:

Внутренняя культура first. Нельзя строить внешний бренд при гнилом internal.

Кейс 2: Митап без follow-up

Ситуация:

Провели отличный митап на 80 человек. Не отправили email, не выложили видео, не анонсировали следующий.

Что произошло:

Momentum потерян. Следующий митап → пришло 30 человек. Сообщество не сформировалось.

Урок:

Follow-up критичен. Митап не заканчивается когда люди разошлись.

Кейс 3: Амбассадоры выгорели

Ситуация:

Нашли 3 сильных спикеров. Нагрузили их: каждый делает по 10 докладов в год. Не выделили official время. Через год все трое refuse участвовать.

Что произошло:

Потеря лучших амбассадоров. Нужно искать новых и строить с нуля.

Урок:

Распределите нагрузку. Официально выделяйте время. Благодарите и reward.

18.4. Как не выгореть в DevRel

Признаки выгорания

- Не хочется идти на митап/конференцию
- Раздражают вопросы в community
- Потеря креативности (делаете на автомате)
- Ощущение "никто не ценит"
- Физическое истощение

Профилактика

1. Делегируй

"Делегируй или сдохнешь" — из практики DevRel.

Что делегировать:

- Рутину (автоматизация, боты)

- Модерацию (ко-модераторы)
- Организацию events (ко-организаторы)
- Нижние 80% задач (совет руководителя)

2. Установите границы

- Не отвечайте на сообщения 24/7
- Weekend = отдых (не проверяйте чат)
- After conference = день recovery

3. Делайте то что energizing

Разные форматы для разных людей:

Тип 1: Энерджайзер

Прет от общения. Пишет часто, легко. → Фокус на спикерство, митапы, community

Тип 2: Дисциплинированный

Держит ритм, делает по плану. → Фокус на регулярный контент, systematic approach

Тип 3: Спонтанный

Вдохновение приходит волнами. → Делайте когда есть энергия, не заставляйте себя

Тип 4: Перегруженный

Совмещает DevRel с другой ролью. → Сфокусируйтесь на 1-2 активностях max

Найдите свой формат — то что наполняет энергией, а не истощает.

4. Support network

- Присоединитесь к DevRel community (есть чаты для DevRel-ов)
- Найдите ментора
- Делитесь проблемами (вы не один)

5. Берите breaks

- После крупной конференции — 2-3 дня отдыха
- Vacation (real vacation, не “работаю из другого города”)
- Mental health days (если чувствуете burn out approaching)

Ключевые выводы главы

1. **Ошибки неизбежны.** Важно учиться и корректировать быстро.

2. **Самые частые:** Количество > качество, игнорирование метрик, продающий тон, выгорание амбассадоров.
3. **Внутреннее важнее внешнего.** Токсичная культура внутри разрушит любой external DevRel.
4. **DevRel - это марафон.** Нереалистичные ожидания приводят к разочарованию.
5. **Burnout — главная угроза.** Делегируйте, устанавливайте границы, делайте то что energizing.
6. **Вы не один.** DevRel community поддержит. Просите помощи.

Статус главы: Черновик v1.0

Объем: ~1,500 слов (~6 страниц)

Следующая редакция: Добавить больше кейсов провалов, детальнее про prevention burnout, checklist самодиагностики

Заключение

Что вы узнали

Мы прошли путь от “Что такое DevRel?” до конкретных инструментов, метрик, и кейсов.

Вы теперь знаете:

- Что такое DevRel** и чем он отличается от маркетинга, HR, community management
 - Какие навыки нужны** и как их развивать
 - Как строить бренд работодателя** от аудита до EVP и стратегии
 - Как создавать контент** который работает для developer audience
 - Как строить сообщества** и управлять ими
 - Как организовывать митапы** от идеи до follow-up
 - Как использовать геймификацию** на мероприятиях
 - Как измерять эффективность** и показывать ROI
 - Какие ошибки совершают** и как их избежать
-

Три главных урока этой книги

1. DevRel — это про отношения, не транзакции

Developer Relations = **Relationships**. Это в названии.

Это не “собрать контакты и отправить вакансии”. Это **построение долгосрочных, доверительных отношений** с developer community.

Метафора из книги:

Не никаперство, а отношения по любви. Если ведете себя достойно, честно рассказываете о компании, искренне интересуетесь людьми — они сами начнут искать ваши вакансии и рекомендовать друзьям.

2. Бренд работодателя = марафон, не спринт

Построение сильного employer brand занимает **годы**, не месяцы.

Год 1: Foundation (аудит, стратегия, первые активности)

Год 2: Growth (масштабирование, compound effects)

Год 3+: Leadership (вы уже в топе, люди мечтают у вас работать)

Не ждите immediate ROI. Но долгосрочный эффект огромен.

3. Качество > Количество. Всегда.

Лучше:

- 5 отличных докладов чем 20 посредственных
- 1 крепкое сообщество чем 5 мертвых чатов
- 10 качественных статей чем 50 "для галочки"

Помните:

Разработчики ценят качество. Один плохой опыт может разрушить репутацию которую строили годами.

Будущее DevRel в России

Тренды которые мы видим:

1. Рост спроса

Все больше компаний понимают ценность DevRel. Позиций становится больше.

2. Профессионализация

DevRel перестает быть "непонятной зверушкой" в штате. Появляются стандарты, best practices, образовательные программы.

3. AI-усиление

AI радикально меняет DevRel. Рутиня автоматизируется, фокус смещается на стратегию и отношения.

4. Локализация

Русскоязычное DevRel-сообщество растет. Появляется свой опыт, свои кейсы, не просто копия Запада.

5. Employer branding фокус

В России DevRel чаще про найм чем про product. И это OK — наша специфика.

Ваш следующий шаг

Если вы хотите начать в DevRel:

Прямо сейчас:

1. Предложите internal tech talk в компании
2. Напишите статью на Habr
3. Вступите в DevRel-сообщество в Telegram

Следующие 3 месяца:

- 3 публичных доклада
- 5 технических статей
- Активность в профсообществе

Через год:

- Портфолио готово → ищите DevRel позиции

Если вы DevRel:

1. Проверьте себя по чек-листам из этой книги
2. Выберите 1-2 слабых места
3. Составьте план improvement
4. Попросите feedback у коллег

Если вы руководитель:

1. Оцените: нужен ли вам DevRel? (Глава 1)
 2. Если да — определите goals и budget
 3. Найдите DevRel или выделите кого-то из команды
 4. Дайте time и support
-

Последнее слово

DevRel — это **люди**.

Люди, которые выступают на конференциях.

Люди, которые пишут статьи помогая другим.

Люди, которые строят сообщества.

Люди, которые верят в вашу компанию и делятся этим.

Инвестируйте в людей. Все остальное приложится.

Удачи в вашем DevRel путешествии! 

Рафаэль Миниахметов

Октябрь 2025

P.S. DevRel-сообщество всегда открыто для новых людей. Присоединяйтесь, задавайте вопросы, делитесь опытом. Мы все учимся вместе.

ПРИЛОЖЕНИЯ

Приложение А. Полезные ресурсы для DevRel

Книги

На русском

- Эта книга — База DevRel (вы ее читаете!)

На английском

- “**Developer Marketing and Relations**” by SlashData — теория и практика
 - “**The Business Value of Developer Relations**” by Mary Thengvall — для руководителей
 - “**Developer, Advocate!**” by Geertjan Wielenga — от практика Oracle
-

Сообщества и чаты

Telegram (Россия/СНГ)

- **DevRel русскоязычное сообщество** — основной чат для DevRel-ов
- **Frontend Crew** — frontend разработчики
- **Moscow Python** — Python сообщество
- **DevOps чат** — инфраструктура и DevOps

Discord / Slack

- **DevRel Collective** — международное сообщество
-

Конференции

Россия/СНГ

- **DevRelCon** (если будет в РФ)
- **HighLoad++** — backend и инфраструктура
- **Mobius** — mobile
- **Saint TeamLead Conf** — для лидеров и менеджеров

Международные

- **DevRelCon** (Лондон, SF) — главная DevRel конференция
 - **Developer Week** — San Francisco
 - **API Days** — Париж
-

Онлайн ресурсы

Сайты и блоги

- **DevRel.co** — статьи и гайды
- **Developer Avocados** — блог Mary Thengvall
- **Habr** — русскоязычные tech статьи

YouTube каналы

- **Google Developers** — туториалы и talks
- **Microsoft Developer** — Azure и tools
- **Записи конференций** — HighLoad++, Mobius, etc.

Подкасты

- **DevRel Radio** — интервью с DevRel-ами
 - **Community Pulse** — про community building
-

Инструменты (упомянутые в книге)

Аналитика

- Google Analytics (бесплатно)
- Brand24 (от \$50/мес)

Сообщества

- Telegram (бесплатно)
- Discord (бесплатно)

Автоматизация

- n8n (бесплатно self-hosted)
- Zapier (от \$20/мес)

Контент

- Notion (бесплатно для личного)
- Google Docs (бесплатно)

Примечание: Ссылки и контакты актуальны на октябрь 2025. Проверяйте актуальность.

Приложение Б. Шаблоны и чек-листы

Чек-лист организации митапа

За 8 недель

- [] Определить тему, формат, дату
- [] Согласовать с руководством
- [] Забронировать площадку
- [] Найти 2-3 спикеров

За 4 недели

- [] Создать регистрацию
- [] Анонс в соцсетях
- [] Договор с кейтерингом

За 1 неделю

- [] Reminder участникам
- [] Подготовить реквизит
- [] Репетиция со спикерами

День события

- [] Setup за 2-3 часа до начала
- [] Провести
- [] Clean up

После

- [] Email участникам (next day)
- [] Публикация видео (3-5 дней)
- [] Feedback analysis

Шаблон контент-плана на месяц

Неделя	Формат	Тема	Автор	Дедлайн	Площадка
1	Статья	Kubernetes setup	Иван	5 окт	Habr
2	Видео	Code review	Мария	12 окт	YouTube
3	Доклад	ML pipeline	Петр	19 окт	Митап
4	Статья	Go best practices	Анна	26 окт	Habr

Шаблон отчета по DevRel (квартальный)

```
# DevRel Q1 2025: Отчет
```

Executive Summary

- Ключевое достижение 1
- Ключевое достижение 2
- Главный challenge

Activities

- Докладов: 12
- Статей: 8
- Митапов: 3

Метрики

- Reach: 50,000
- Engagement: 7.5%
- eNPS: +7 пунктов

Business Impact

- Organic applications: +40%
- Hired: 3 attributed DevRel

Next Quarter

- План на Q2

Шаблон EVP для IT-компании

```
# Employee Value Proposition: [Компания]
```

1. Технологии и проекты

Что: [описание]

Impact: [масштаб]

Stack: [технологии]

2. Команда и рост

Кто: [о команде]

Обучение: [что предлагаем]

Карьера: [треки развития]

3. Культура

Ценности: [3-5 ключевых]

Подход: [как работаем]

Balance: [work-life]

4. Компенсация

Зарплата: [уровень рынка]

Бенефиты: [что еще]

Шаблон бюджета мероприятия

Митап на 70 человек

Статья	Сумма
Площадка	30K
Кейтеринг	70K
Мерч	30K
Фото/видео	20K
Промо	15K
Резерв	10K
ИТОГО	175K

Чек-лист качества технической статьи

Перед публикацией:

- [] Код работает (проверено!)
 - [] Примеры complete (можно copy-paste)
 - [] Структура четкая
 - [] Нет опечаток
 - [] Заголовок engaging
 - [] Картинки (если нужны)
 - [] Ссылки рабочие
 - [] Теги релевантные
-

Шаблон Саморепрезентации 1+1+1

1 предложение

[Компания] — [позиция на рынке], которая [ценность] через [подход].

1 абзац

[Расширение каждой части до предложения]

1 страница

[Каждое предложение → абзац]

Примечание: Это базовые шаблоны. Адаптируйте под вашу компанию и задачи.

```
<div style="page-break-after: always;"></div>
```

Приложение В. Глоссарий терминов

A-D

API (Application Programming Interface) – интерфейс для взаимодействия программ

B2D (Business-to-Developer) – бизнес-модель продажи инструментов/сервисов разраб

****CFP (Call for Papers)** – открытый набор докладов на конференцию

****Community Manager** – специалист по управлению сообществами

****CTR (Click-Through Rate)** – процент кликов по ссылке

****Developer Advocate** – представитель интересов разработчиков

****Developer Relations (DevRel)** – построение отношений с разработчиками

****DX (Developer Experience)** – опыт разработчика при использовании продукта

E-H

****EB (Employer Brand)** – бренд работодателя

****eNPS (Employee Net Promoter Score)** – индекс лояльности сотрудников

****EVP (Employee Value Proposition)** – ценностное предложение для сотрудников

****Engagement Rate** – процент вовлеченности аудитории

****HRBP (HR Business Partner)** – HR-партнер для бизнес-подразделения

K-0

****KPI (Key Performance Indicator)** – ключевой показатель эффективности

****NPS (Net Promoter Score)** – индекс потребительской лояльности

****OKR (Objectives and Key Results)** – framework постановки целей

P-T

****ROI (Return on Investment)** – возврат на инвестиции

****SDK (Software Development Kit)** – набор инструментов для разработки

****SMM (Social Media Marketing)** – маркетинг в соцсетях

****TA (Talent Acquisition)** – привлечение талантов (рекрутинг)

****Technical Evangelist**** – технический евангелист (старое название Developer Advocate)

U-Z

****UGC (User-Generated Content)**** – контент созданный пользователями

****Виральный контент**** – контент который распространяется органически (репосты)

****Митап (Meetup)**** – неформальная встреча профессионального сообщества

****Сарафанное радио**** – organic word-of-mouth продвижение

****Холивар (Holy War)**** – бесконечный спор без консенсуса (tabs vs spaces, etc.)

****Примечание:**** В книге объяснены все термины при первом упоминании. Этот глоссарий