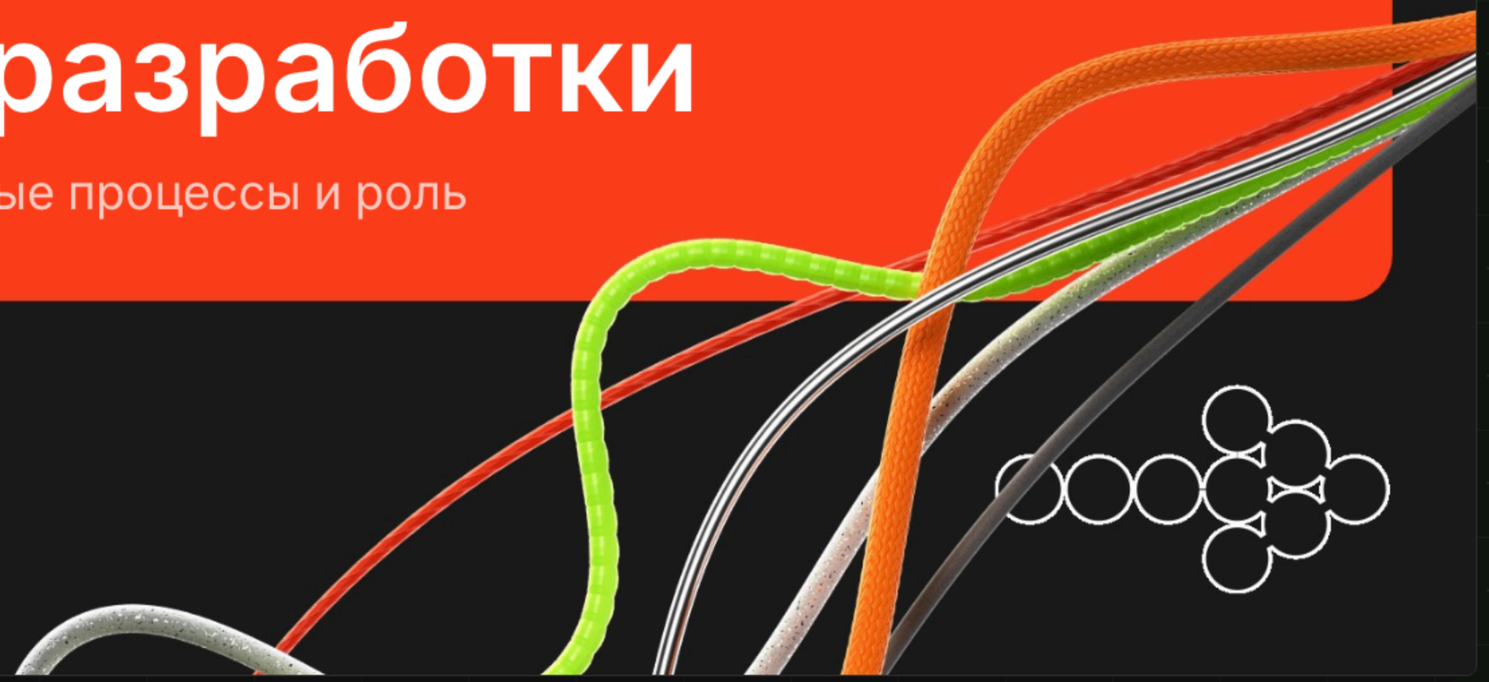
Яндекс

dream → teamlead · 28 марта 2026

AI меняет культуру разработки

Исследование, инженерные процессы и роль тимлида



Часть 1

01

AI4SDLC Research 2025

Как AI влияет на разработку ПО — метаанализ и опрос инженеров

Исследование Дизайн

Как мы смотрели на AI4SDLC

Практики, культура, доверие

- Где AI уже стал ежедневным.
- Где внедрение остаётся редким.
- Как связаны скорость, качество и доверие.
- Что превращает личный выигрыш в командный.

Исследование Метаанализ

Метаисследование: источники данных

50+ исследований, четыре типа источников

- Массовые опросы разработчиков и технических лидеров.
- Телеметрия и продуктовые/инженерные логи.
- Полевые эксперименты и контролируемые исследования задач.
- Внутренние корпоративные исследования и интервью.

AI4SDLC (2025)

Исследование Опрос

Опрос инженеров и техлидов

- Самооценка эффекта на продуктивность, качество и доверие.
- Частота использования AI по типам задач SDLC.
- Переменный размер выборки по вопросам.
- Описательный анализ без причинно-следственных выводов.

Ограничения исследования

Как читать результаты

- Выборка отражает контекст.
- Процессы тоже влияют на продуктивность.
- Индустрия меряет продуктивность по-разному.
- Новизна и адаптация искажают динамику.

Что видно по метаисследованию

Шесть повторяющихся паттернов в исследованиях 2023–2025

Default, но неравномерно

AI вошёл в daily toolkit, но эффект зависит от задачи.

Coding быстрее поставки изменений

Узкие места уходят в review, тесты, интеграцию.

ROI через процессы

Доверие и роль инженера зависят от зрелости SDLC-контура.

Что видно по опросу инженеров

Шесть top-line сигналов из AI4SDLC Survey 2025

AI стал повседневным инструментом

Кодогенерация и автодополнение вошли в daily practice.

Локальный эффект сильнее

Продуктивность ощущается сильнее качества и пропускная способность команды.

Доверие и ожидания расходятся

Доверие к AI-коду низкое, но бизнес ждёт эффекта.

AI уже рядом, но не везде

High-risk задачи пока требуют человека

Массовые сценарии

- 58% часто используют генерацию.
- Кодинг и отладка — ядро.

Редкие сценарии

- 24% применяют AI для ревью кода регулярно.
- 18% используют AI для legacy-задач.

Результаты Продуктивность

Личная скорость не равна поставке изменений

Ускорение кодинга ≠ ускорение поставки ценности

- 64% видят рост продуктивности.
- Командный эффект ограничен процессами.
- DORA 2024: AI-внедрение снижало пропускную способность разработки.
- DORA 2025: пропускная способность разработки лучше, instability осталась.

Результаты Качество

Доверие требует системной проверки

Verify, don’t trust как инженерная норма

Качество

- 32% отмечают улучшение качества кода.
- 14% фиксируют ухудшение.
- Контекст и сложность определяют результат.

Доверие

- 49% не доверяют AI-коду.
- Доверяют лишь 11%.
- Культура проверки — ключевое условие эффекта.

Узкое место ушёл в поставке изменений

Главный вывод для тимлида

- Commit → prod всё ещё долгий.
- Review, тесты и релизы ограничивают.
- Документация становится context quality.
- Побеждает перестройка системы работы.

Часть 2

02

От классического PDLC к AI-native-разработке

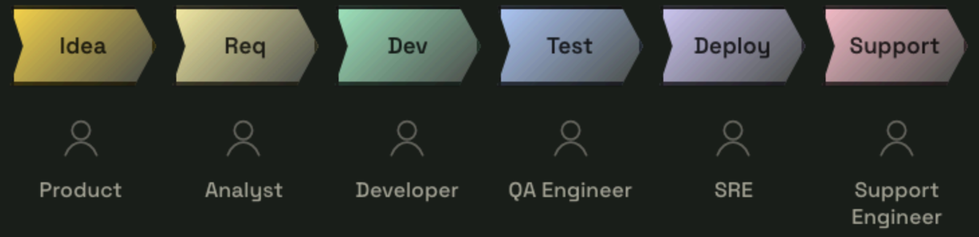
SE 1.0 → SE 2.0: не смена инструмента, а перестройка системы

Два режима

SE 1.0 и SE 2.0

Role-based SDLC vs agent-based SDLC: как меняется сам контур работы

Software Engineering 1.0 (Role-based SDLC)



Потери на передачах работы

Сделали часть AI сценариев внутри ролей

Локальные оптимизации внутри

На brownfield проектах обкатываем AI сценарии по ролям

Software Engineering 2.0 (Agent-based SDLC)

Меньше потерь на передачах работы

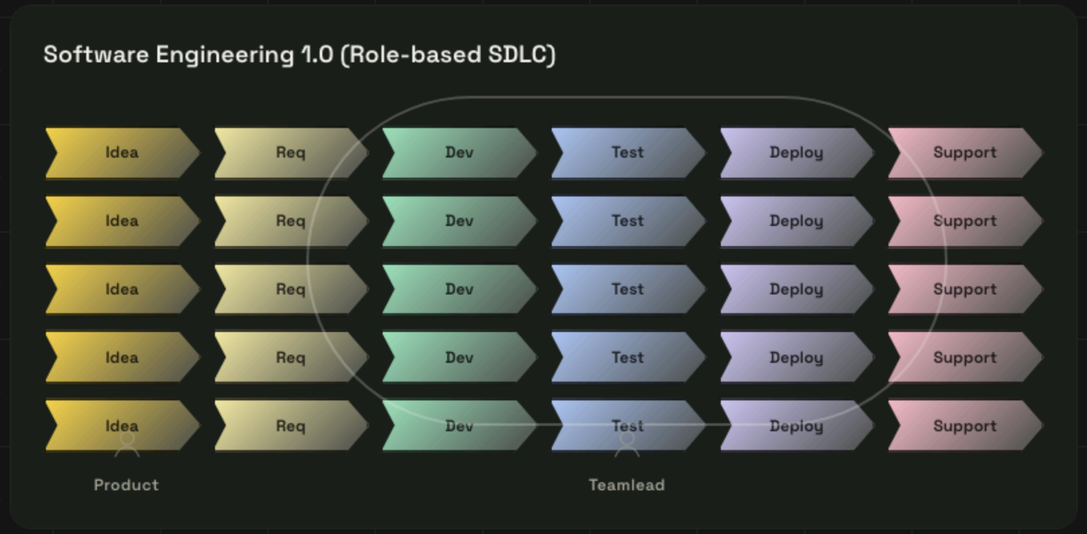
Быстрее e2e сценарии

На greenfield проектах пробуем agent-based разработку

Забираем наработки по сценариям

Забираем наработки по сценариям

SE 1.0 и SE 2.0 и роль тимлида



На brownfield проектах обкатываем AI сценарии по ролям

На greenfield проектах пробуем agent-based разработку

Индустрия 2024-2026

Лидеры уже перестраивают контур

Google, Meta, AWS, DORA

- Google: AI внутри engineering loop.
- Meta: JIT-тесты под конкретный diff.
- AWS: роли сжимаются в один цикл.
- DORA: payoff зависит от платформы.

SE 2.0 ≠ бесплатное ускорение

Почему ускорение кодинга может замедлить поставку

- Слабый version control → rework.
- Шумные тесты → нестабильность.
- Агент быстрее открывает PR.
- Основа: branch protection, CI, signing.

SE 1.0 ↔ SE 2.0: обмен практиками

Не победа одного режима, а системный перенос практик

SE 2.0 → фронтир

Полигон для orchestration, JIT-тестов и автодокументации.

SE 1.0 → фундамент

Зрелые контрольные точки, developer platform, безопасная поставка, операционная надёжность.

Перенос практик

1.0 даёт защитные ограничения; 2.0 возвращает автоматизацию.

Часть 3

03

От AI-native-разработки к AI-native-организации

Следующий уровень сжатия: меняется не только SDLC, но и сама модель компании

Сдвиг Org Design

Почему следующий шаг — AI-native-организация

Код ускорился, тормозят решения и handoffs

Сжимаются слои

После ролей внутри SDLC сжимаются оргпрослойки.

Decision latency

Согласования и context handoffs становятся узкое место.

Leverage вместо headcount

Фокус на output per employee и платформе.

Сигналы Big Tech

Org-сигналы у лидеров рынка

Шире span, меньше уровней, сильнее платформа

Шире span

- Meta: AI org ratio до 1:50.
- Nvidia: ~50 reports у CEO.
- Ставка на сильных IC.

Меньше уровней

- Amazon: +15% IC/manager.
- Microsoft и Google: AI-first tooling.
- Intel: faster execution, fewer layers.

Flattening не даёт бесплатной скорости

Без платформы плоскость превращается в хаос

- Нужны golden paths, policy, observability.
- Иначе менеджеры тонут в исключениях.
- Ratios 1:50 нельзя копировать слепо.
- Менеджер проектирует human-agent контур.

Часть 4

04

Как меняется роль тимлида

Новые задачи лидера в мире AI-native-процессов

Тимлид проектирует систему работы

Не только распределяет задачи

Раньше

- Распределял задачи.
- Контролировал сроки и качество.
- Нанимал под роли.

Теперь

- Проектирует human-agent loop.
- Настраивает quality gates.
- Растит диагностику и ревью.

Новая работа тимлида

Пять зон ответственности

- Задаёт автономность AI по риску.
- Вшивает verification в definition of done.
- Управляет контекстом: ADR, runbooks, docs.
- Перестраивает метрики, policy и обучение.

Три главных вызова для тимлида

Что может пойти не так при переходе к AI-native-everything

Эрозия навыков

Код писать легче, глубокие навыки могут деградировать.

Иллюзия скорости

Локальная скорость растёт, поставка может стоять.

Доверие команды

Verify должен стать нормой, а не недоверием.

Ключевые тезисы

Что забрать с собой

- AI ускоряет артефакты, а не поставку.
- SE 2.0 усиливает инженерную дисциплину.
- Тимлид проектирует human-agent систему.
- Измеряйте скорость поставки ценности.

- ↗ ai4sdlc-research.space · ↗ tellmeabout.tech

Ссылки и материалы

Источники References

AI adoption, DevEx, culture

- AI4SDLC Research 2025
- DORA AI + State of DevOps
- GitHub Octoverse + Copilot docs
- Stack Overflow, JetBrains, Atlassian

Фидбек

Как вам доклад?

Поделитесь, пожалуйста, фидбеком —
это поможет нам стать еще лучше

Q&A

Ждём ваши вопросы

Задать в чате
Yandex for Teamleads

Задать в форме

Яндекс

Спасибо!

Если хотите изучить упомянутые в докладе материалы, то переходите в канал ”Книжный куб”, где есть все ссылки на них

Александр Поломодов

Technical Director & Fellow,
T-Технологии

@Book_Cube