стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Директор 2026: AI ускорил код. Почему компания всё равно тормозит

Управление 2026
Александр Апазиди
24.04.2026

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Представлюсь!

Александр Апазиди

CTO / CIO · C-level опыт · тренер, ментор

- Более 30 лет в сфере информационных технологий, прошел путь от разработчика и руководителя проектов до ИТ-директора
- Schlumberger, IBM, Nestle, InBev, Volvo, Metro C&C, СИБУР
- Преподаватель в Стратоплан, Яндекс Практикум
- Блог про ИТ менеджмент

https://t.me/apazidi_IT

@apazidi

О чем поговорим

- Почему разработчик может ускориться в 3–5 раз, а компания — только на 10%
- Где AI реально усиливает delivery, а где лишь раздувает типовые метрики
- Какие 6 типов организационных задержек чаще всего съедают выигрыш от AI
- Что директор может изменить, чтобы ускорить не только создание кода, а весь поток ценности

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

AI ускоряет или тормозит?

Код пишется быстрее. Компания быстрее не становится

Инженеры генерируют больше результата

- Больше кода
- Больше PR
- Быстрее тесты
- Быстрее документация
- Быстрее прототипы

→

Поток

- Lead time почему-то не падает
- Очереди согласований и ревью увеличиваются
- Количество стыков растет

→

Бизнес-эффект

- Time-to-market не удваивается
- Ценность создается медленнее, чем ожидали
- Руководство видит разрыв между “локально быстрее” и “системно быстрее”

Что я слышу в разговорах с разработчиками и руководителями

Что я слышу в разговорах с разработчиками и руководителями

Разработчики

≈ 50%: “ускоряет в разы”

≈ 50%: “скорее тормозит”

Сильная поляризация на уровне individual contributor

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Что я слышу в разговорах с разработчиками и руководителями

Разработчики

≈ 50%: “ускоряет в разы”

≈ 50%: “скорее тормозит”

Сильная поляризация на уровне individual contributor

Руководители / большие организации

Малая команда

x2–x3

Big Tech / enterprise

~10% или “не уверены”

Ощущение

wishful thinking

Чем больше организация, тем слабее локальный AI-эффект похож на ускорение всей системы

Один и тот же AI даёт три разных эффекта

Разработчик (IC)

“я стал в разы быстрее”

x3–x5

Команда 5–6 человек

ускорились, но меньше чем IC

x2–x3

Организация 200+

куда-то ускорение делось

~10%

Один и тот же AI даёт три разных эффекта

Разработчик (IC)

“я стал в разы быстрее”

x3–x5

→

Команда 5–6 человек

ускорились, но меньше чем IC

x2–x3

→

Организация 200+

куда-то ускорение делось

~10%

Масштаб “съедает” локальный выигрыш через согласования, стыки, зависимости между людьми и отделами

В большинстве средних и небольших компаний AI становится новой нормой

Использование

90%
используют AI на работе

Время

≈ 2 часа
в рабочий день

Задачи

код, ревью, документация, анализ, тесты

В большинстве средних и небольших компаний AI становится новой нормой

Использование

90%
используют AI на работе

Время

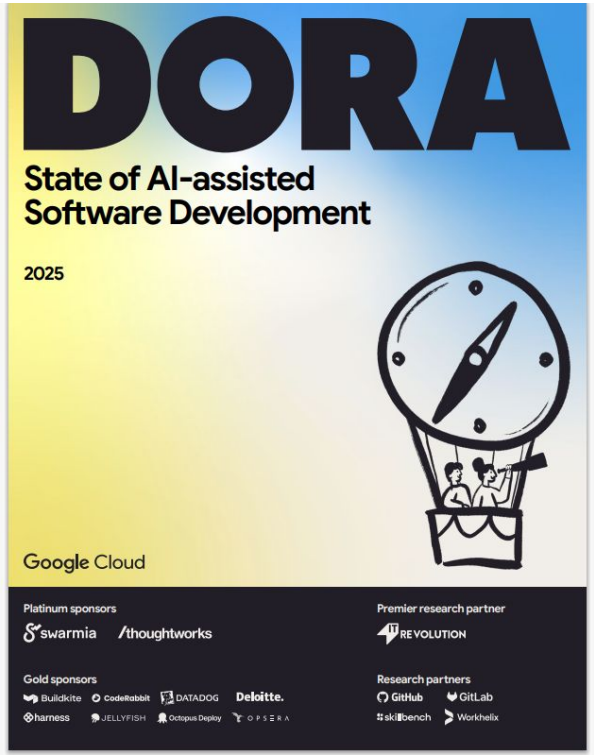
≈ 2 часа
в рабочий день

Задачи

код, ревью, документация, анализ, тесты

- Вопрос “внедрять или нет” уже не стоит
- Главный вопрос: как извлечь ценность для всей компании, а не только локальный выигрыш

DORA 2024 vs DORA 2025



https://dora.dev

DORA 2024: неожиданный сюрприз

Парадокс 2024

- рост использования AI сопровождался ухудшением **производительности (throughput)**
- и ростом **нестабильности в поставке**

Что мы ждали

AI → быстрее код → быстрее поставка

Что показал DORA 2024

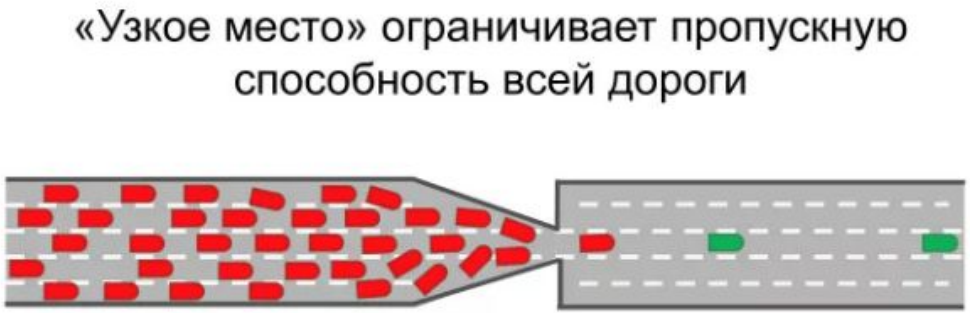
AI дает локальный выигрыш, но пропускная способность падает, а нестабильность растет

DORA 2025: сюрпризы продолжаются

DORA 2025: сюрпризы продолжаются

- команды учатся использовать AI эффективнее
- инструменты становятся более продвинутыми (появились агенты)
- но безопасно переваривать AI-ускорение система ещё не научилась

Это не парадокс - это классика



"Час, сэкономленный не в узком месте — это мираж. Час, сэкономленный в узком месте — это час, сэкономленный всей системой."

Проверим на цифрах

Разработчик (IC)

Кодирование 80%, остальное 20%

Если кодирование ускоряем в x5, то в итоге и получаем 80% → 16%

x3–x5

Команда 5–6 человек

Кодирование 60%, остальное 40%

Если кодирование ускоряем в x5, то в итоге и получаем 60% → 12%

x2–x3

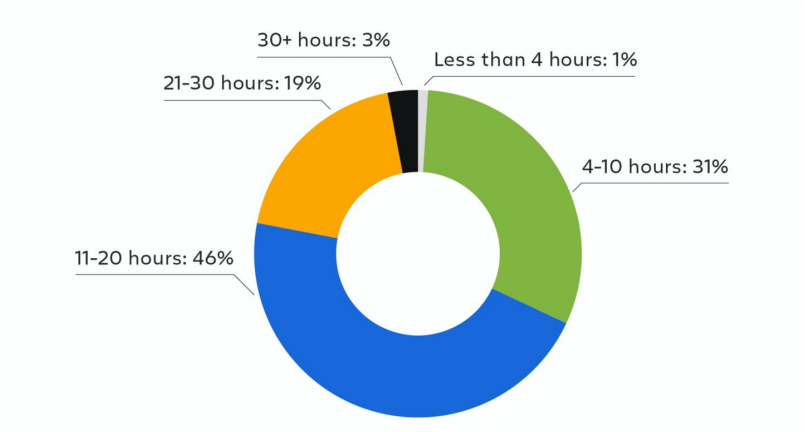
Организация 200+

Кодирование 15%, остальное 85%

Если кодирование ускоряем в x5, то в итоге и получаем 15% → 3%

~10%

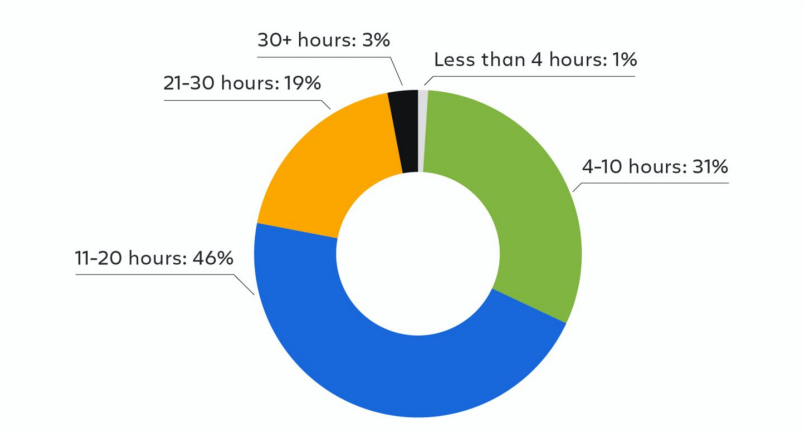
Пример: отчет Atlassian DevEx 2025

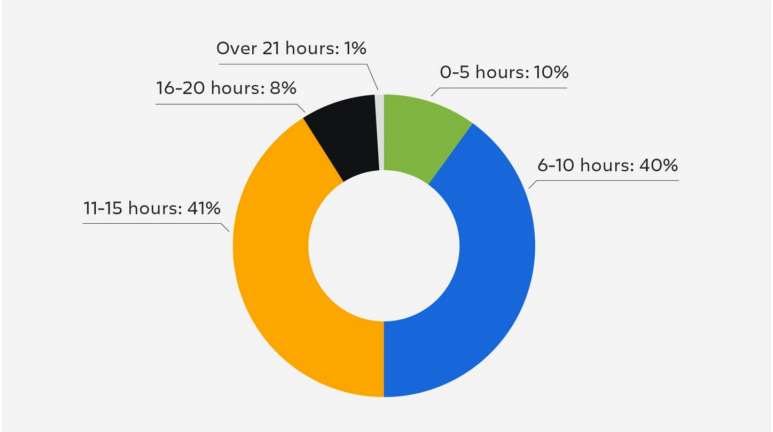


Исследование показывает, что 68% разработчиков с помощью AI высвобождают более 10 часов в неделю!

Ссылка на исследование

Пример: отчет Atlassian DevEx 2025



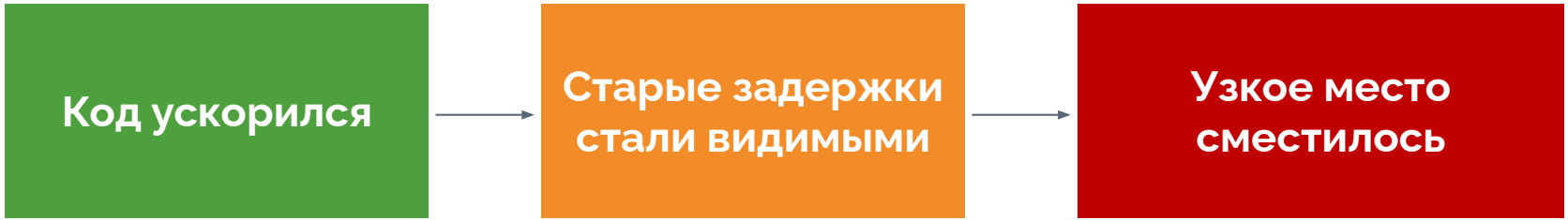


Исследование показывает, что 68% разработчиков с помощью AI высвобождают более 10 часов в неделю!

.. И тратят более 10+ часов в неделю на согласования и non-coding задачи

Ссылка на исследование

AI смещает узкое место с разработки на скорость принятия решений



Теория ограничений

1. Определи ограничение системы
2. Максимизируй его отдачу
3. Подчини все остальное этому решению
4. Расширь ограничение
5. Повторить

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

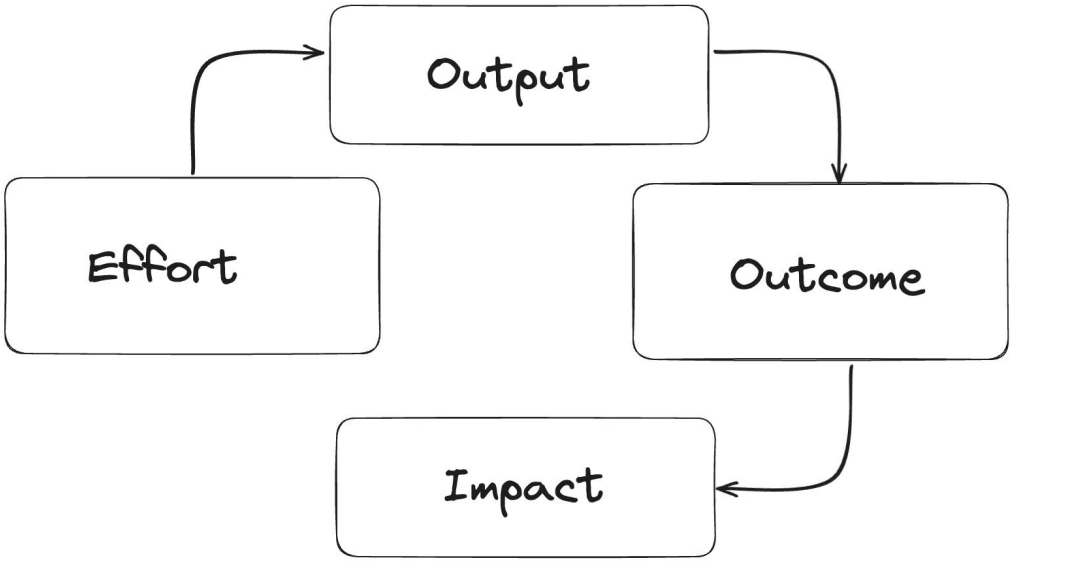
Работа Директора 2026

1. Определи ограничение системы
2. Максимизируй его отдачу
3. Подчини все остальное этому решению
4. **Расширь ограничение**
5. **Повторить**

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

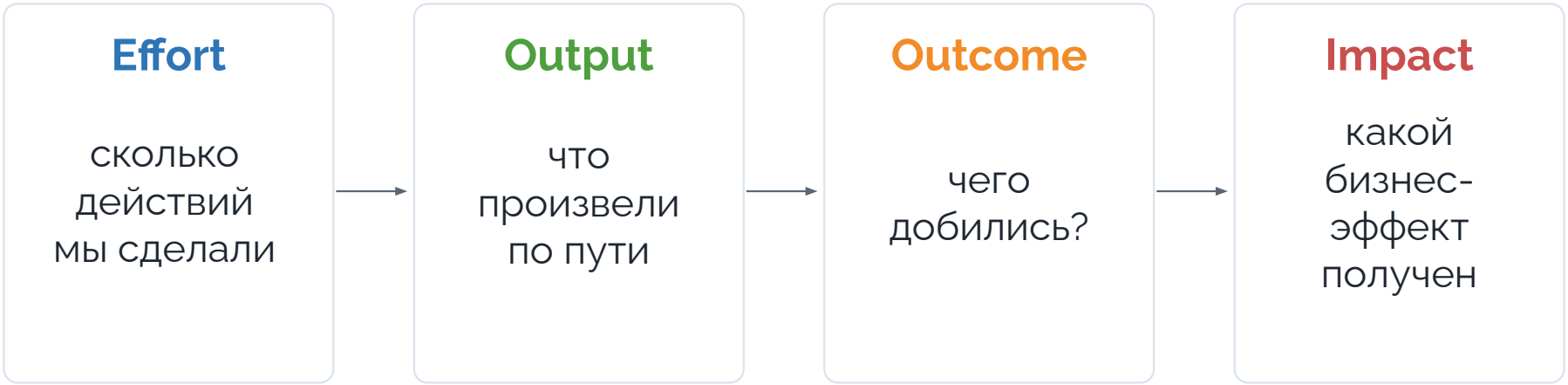
А как искать эти самые узкие места?

Effort → Output → Outcome → Impact

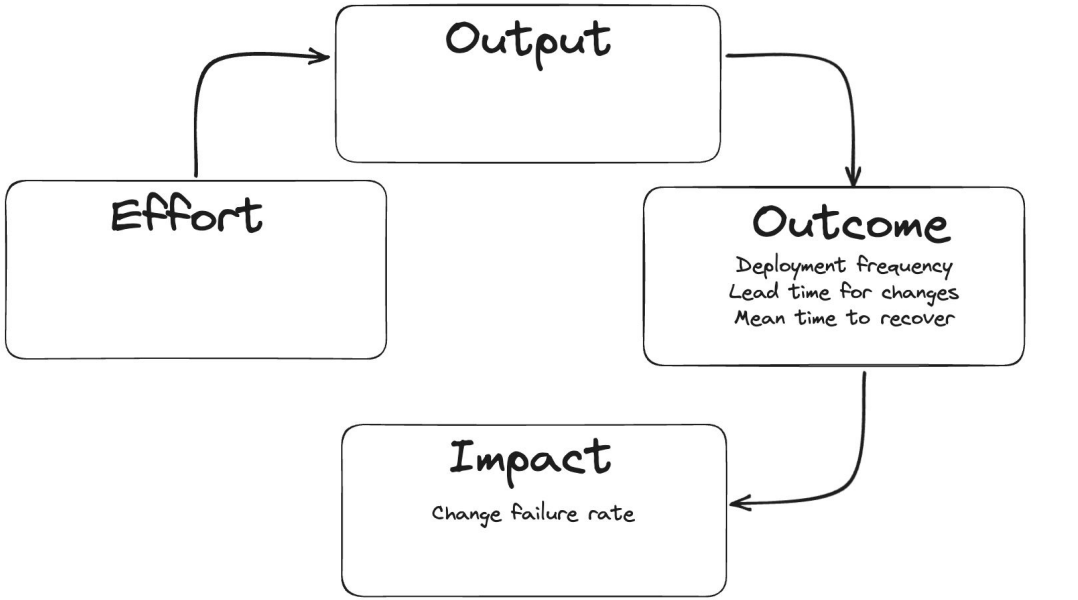


Measuring developer productivity? A response to McKinsey (2023, Gergely Orosz and Kent Beck)

Effort → Output → Outcome → Impact



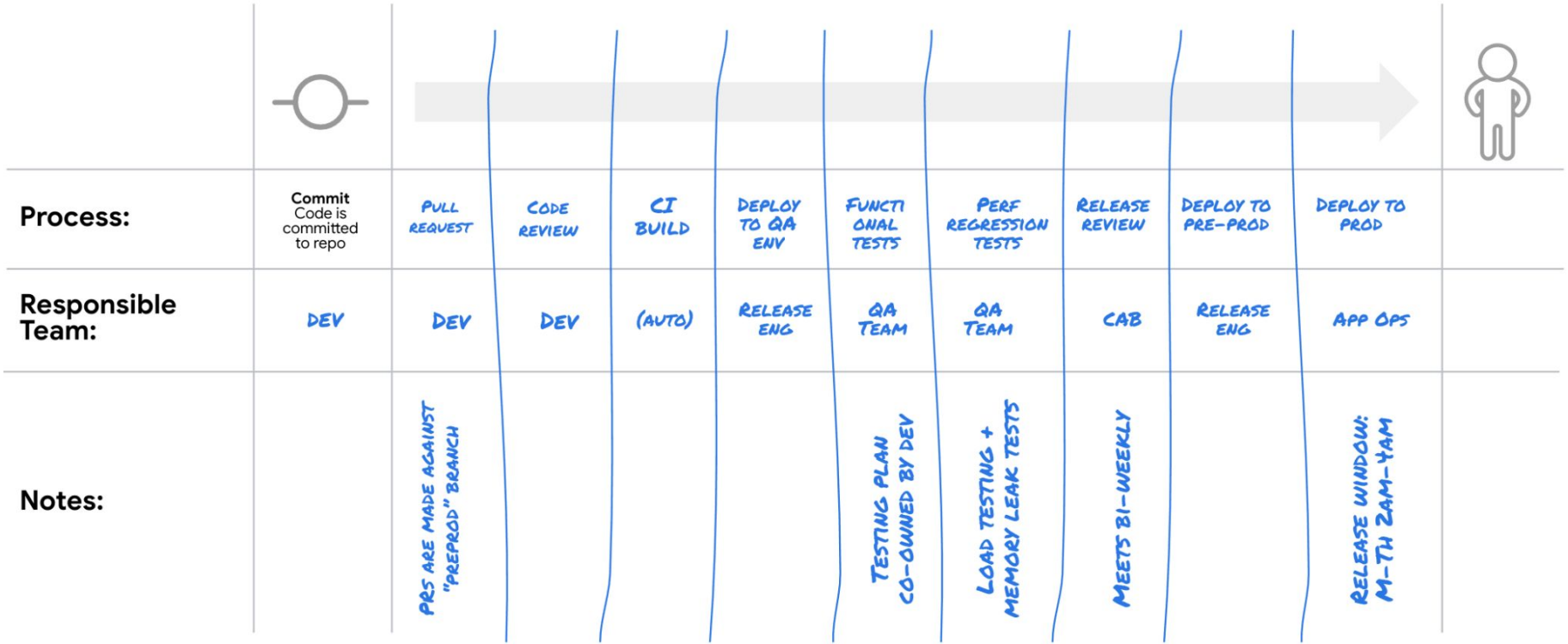
Effort → Output → Outcome → Impact



Kent Beck / Software Design: Tidy First? and pragmaticengineer.com

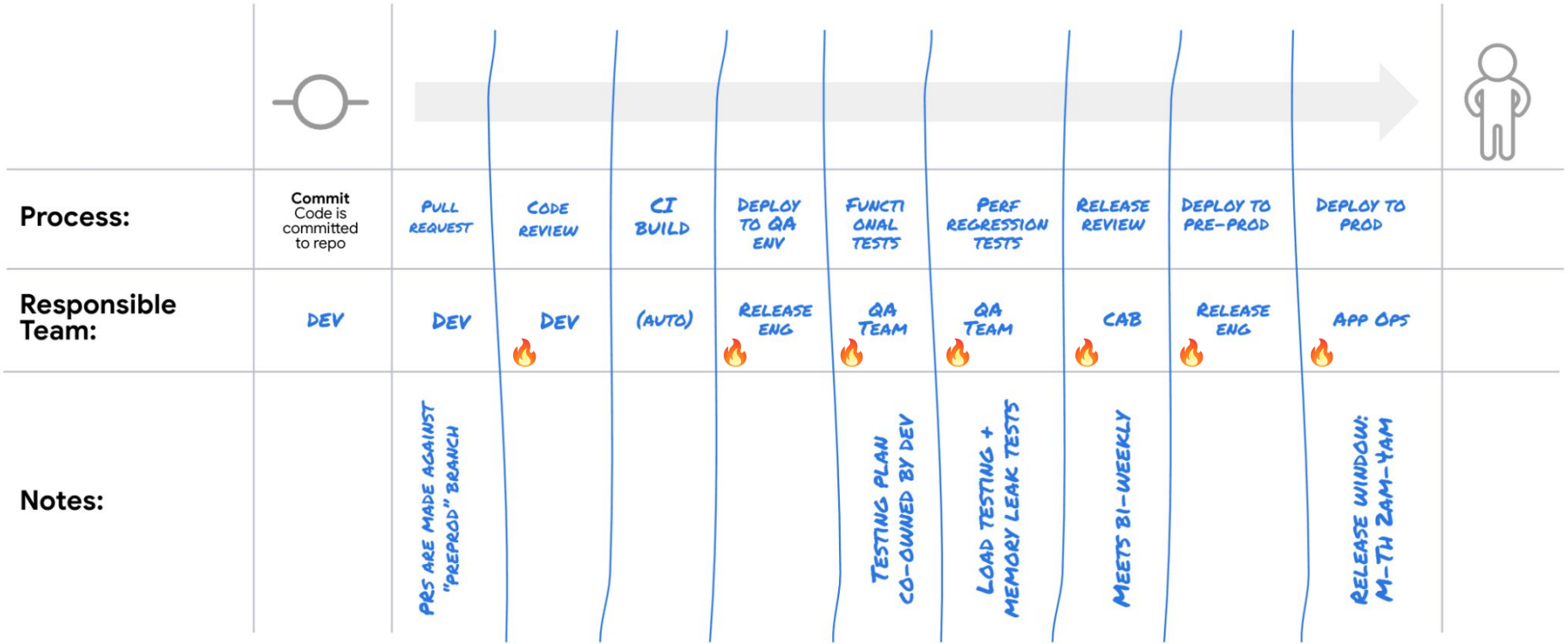
[Measuring developer productivity? A response to McKinsey (2023)]

Value Stream Management



https://dora.dev/guides/value-stream-management/

Value Stream Management

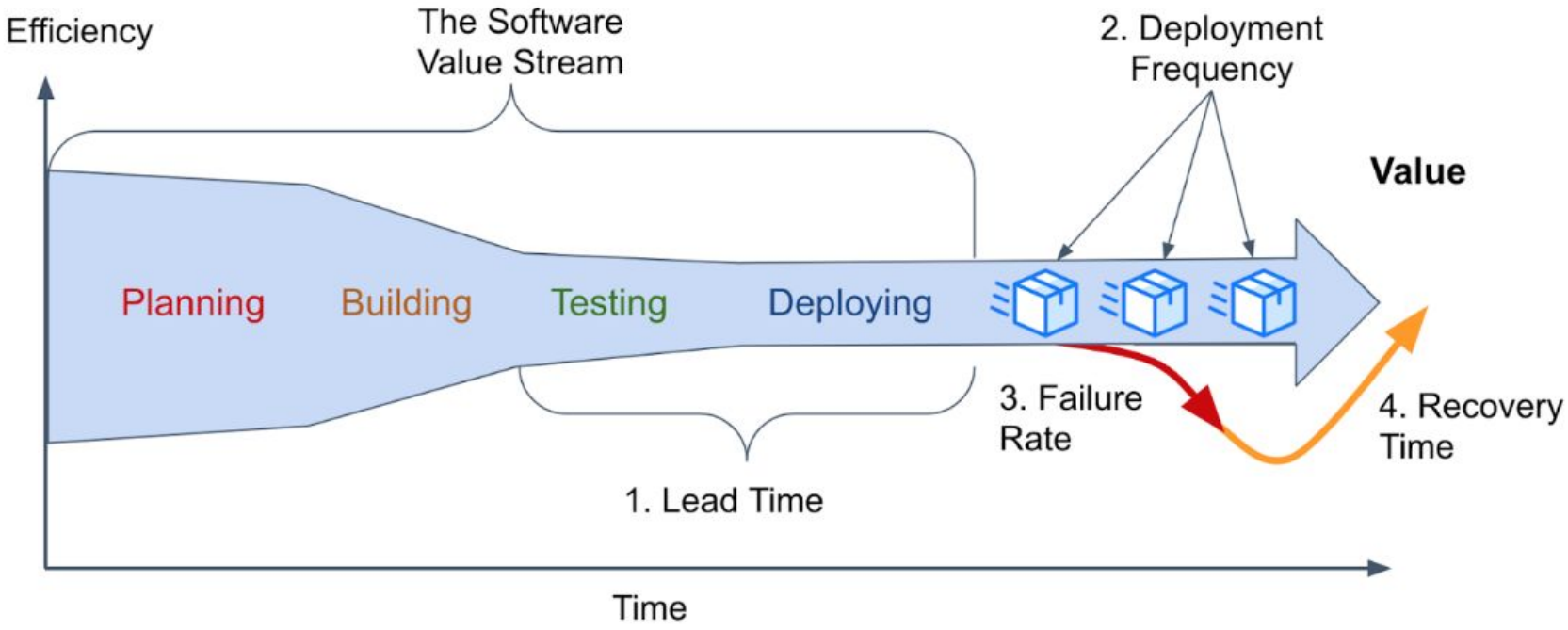


https://dora.dev/guides/value-stream-management/

Value Stream Management

https://dora.dev/guides/value-stream-management/

Value Stream Management



Шесть мест, где AI-выигрыш теряется

1. Время на решения

2. Зависимости

3. Контроль и регулирование

4. WIP (задачи в работе)

5. Переделки

6. Платформы

Шесть мест, где AI-выигрыш теряется

1. Время на решения
2. Зависимости
3. Контроль и регулирование
4. WIP (задачи в работе)
5. Переделки
6. Платформы

Напишите в чат, если какие-то из этих мест “болят” у вас в организации

1. Время на решения (decision latency)

- Что важнее “быстро” или “качественно” или “безопасно”?
- Неясно, кто принимает решение, всё эскалируется наверх
- Споры на арх. советах. Нет единого владельца или он top-level
- Спор идет дольше, чем написать и переделать несколько раз

Главная метрика

Затраты времени (p90) на шаги с принятием решений в VSM

2. Зависимости - ждем другую команду

Главная метрика

Blocked time / lead time

3. Контроль и регулирование (IT Governance)

Арх. совет | ИБ | Compliance | Ручной approval

- AI быстрее генерирует изменения (код готов)
- но скорость review и проверок не растёт с той же скоростью
- в результате получаем новое узкое место - очередь на согласование

Метрики:

1) **Среднее время review**
2) **Кол-во review / количество исключений из правил**

4. WIP - параллельная работа

Много начатого
Мало завершённого

Главная метрика

WIP per team

- AI помогает открыть больше работы
- Вспоминаем Kanban мантру “Stop starting, start finishing”

5. Rework, переделка съедает ускорение

Главные метрики

Rework rate
Change failure rate

Если AI ускорил первое прохождение, но увеличил переделку — бизнес-выигрыш может быть нулевым

6. Платформенная команда - ускоритель или тормоз?

AI требует не меньше платформы, а больше

CI/CD | envs | secrets | guardrails | self-service

Главные метрики

Platform SLA
Self-service ratio

- слабая platform team становится заметной мгновенно
- ticket-очередь не успевает за AI-ускорением

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Пять типовых “обезболивающих таблеток” директора

Полномочия на решения

DACI / владельцы/ SLA

Маленькие команды

ближе к outcome / impact

WIP limits

ограничить портфель

Platform as Product

сервисы само-обслуживания

Value stream mapping

увидеть узкое место

Маленькие и автономные команды

Небольшие команды с “loosely coupled architecture”

- Выигрыш в скорости в 20-30% (измеряли velocity, PR)
- И увеличили свой “happy time” на 50%

Небольшие команды с “tightly coupled with ERP (SAP)”

- Ускорения от использования AI не смогли обнаружить / реализовать

Маленькие и автономные команды

Небольшие команды с “loosely coupled architecture”

- Выигрыш в скорости в 20-30% (измеряли velocity, PR)
- И увеличили свой “happy time” на 50%

Небольшие команды с “tightly coupled with ERP (SAP)”

- Ускорения от использования AI не смогли обнаружить / реализовать

Вывод: маленькая команда может генерировать сразу outcome / impact. Там где раньше нужно было несколько команд, теперь справится одна, но ей нужно дать полномочия и автономность

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Так какие конкретные есть рецепты для директора?

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Рецепт зависит от компании

Тип организации	Куда целиться / узкое место	Как влияет AI
IT подразделение в не-IT компании (например производство)	**Спрос**: запросов от бизнеса всегда больше, чем доступных ресурсов	Только усугубит: больше мощности → больше проектов и запросов от бизнеса
Большой Enterprise	**Координация**: стыки, регуляторика, очереди, владельцы решений	AI подсветит узкие места, но не расширит их
IT Product	**Темп**: Больше фич → выше конкурентоспособность	AI помогает в гипотезах и прототипах, но потом опять упираемся в координацию
IT Outsource	**Бизнес-модель**: Ценность “мы умеем программировать” умирает	AI как угроза: обесценивает часы, атакует конкурентное преимущество

ИТ в не-ИТ компании

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

КОНТЕКСТ

десятки проектов на десятки разработчиков. Legacy, SAP/1C/CRM/MES/WMS, небольшая кастом разработка

УЗКОЕ МЕСТО

Спрос превышает ёмкость, независимо от скорости.

ЧЕГО НЕ ДЕЛАТЬ

Поднимать velocity, чтобы взять 20 проектов вместо 15. Бизнес вернётся с ещё десятью.

ТРИ СОВЕТА

1 **Сократить WIP проектов**

С десятков проектов в параллель, сократить по 2-3. Stop starting, start finishing

2 **Использовать vibe code на пользу бизнеса**

Любой запрос - сначала AI прототип. Возможно этого будет достаточно в большинстве случаев

3 **AI-документация легаси.**

Расшивает узкое горло «проект ждёт Васю, который один знает 1С». Это не про скорость разработки — про то, чтобы она вообще могла начаться.

Большой Enterprise

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

КОНТЕКСТ

сотни–тысячи разработчиков, SAFe, Team Topologies, платформы уже есть.

УЗКОЕ МЕСТО

Координация, стыки, регуляторке, время решений — то, что AI не трогает.

ЧЕГО НЕ ДЕЛАТЬ

Ждать от Cursor'а 30% общего ускорения. Если кодирование — 15% цикла, его 5x даёт ~12% общего. Не тратить ещё квартал на рост AI adoption rate как на KPI.

ТРИ СОВЕТА

1. **Параллельные AI-native команды**
   Не ломать SAFe и старую оргструктуру - это займет годы. Дать возможность командам самим вырастить новый процесс

2. **Перераспределить 20-30% capacity**
   Вместо того, чтоб выигрыш уходил в "happy time", пусть он лучше уходит в то, что раньше хотели сделать, но было невозможно (проекты класса "D")

3. **Дать автономию (где возможно)**
   Senior IC делает работу команды - путь у него будут полномочия тимлида. А у опытного тимлида таких сеньоров - полномочия руководителя отдела

IT Product: не упустите окно возможностей

КОНТЕКСТ

50–200 разработчиков, IT = бизнес (финтех, медтех, *.тех)

УЗКОЕ МЕСТО

Свой Темп vs конкуренты — при размере, команды, где есть реальное ускорение

ЧЕГО НЕ ДЕЛАТЬ

Нанимать ещё 50 инженеров в этом году ради «роста». Массовый найм больше не гарантирует рост скорости

ТРИ СОВЕТА

1 **Делать то, что раньше было невозможно**

Проверить гипотезу за один день? Отлично
А как теперь их встроить в продукт на следующий день?

2 **AI в продукте, не только в разработке**

AI это одна из самых больших угроз для SaaS : зачастую проще навайбкодить, чем покупать подписку. Какой будет ответ вашего продукта на это?

3 **Выделяйте лучших**

Разрыв между средними и лучшими еще больше увеличился. Не потеряйте своих лучших (подход Netflix)

IT Outsource: смена бизнес модели

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

КОНТЕКСТ

Раньше продавали часы разработчиков. Теперь продавать часы с Cursor?

УЗКОЕ МЕСТО

Бизнес модель. Продать “мы создаем лучшие IT решения” уже не так просто

ЧЕГО НЕ ДЕЛАТЬ

Продолжать продавать часы со старыми нормами. Клиент либо уже понял либо скоро поймет, что скорость разработки снизилась

ТРИ СОВЕТА

1 **Outcome ценообразование**

Не часы, а результат = деньги, которые клиент сможет заработать. См “Цель 3” от Голдратта.

2 **Специализация**

AI без контекста и знаний не работает. Domain-специфичный RAG + экспертиза людей — конкурентное преимущество, которое не заменить AI

3 **AI transformation consulting**

Пока другие думают, как встроить AI в свои процессы - вы это уже испытали на себе и вполне можете продавать как новую услугу

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Подведем итоги

Итоги

- AI может ускорять кодирование в разы
- Кодирование перестало быть узким местом в разработке
- Скорее всего, узкое место теперь в согласованиях, стыках, принятии решений.
- Поэтому, чем крупнее организация — тем меньше виден выигрыш от AI
- Главная работа директора — найти узкое в своем потоке ценности (value stream)
    - Если еще не сделали - value stream mapping ваш первый и самый важный шаг
    - Для этого не нужно собирать большую встречу. Возьмите одну фичу и восстановите по шагам сколько заняло время от идеи до релиза, с описанием каждого шага - это можно сделать за пару часов

Главная работа директора 2026 — не внедрить AI, а не потерять его выигрыш по дороге

Q&A

Александр Апазиди

CTO//CIO, тренер, ментор

- Более 30 лет в сфере информационных технологий, прошел путь от разработчика и руководителя проектов до ИТ-директора
- Schlumberger, IBM, Nestle, InBev, Volvo, Metro C&C, СИБУР

@apazidi