стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Как строить устойчивые ИТ команды

Stratoplan B2B Conf
Александр Апазиди
30.01.2026

Представлюсь!

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
- Блог про ИТ менеджмент

  https://t.me/apazidi_IT

@apazidi

О чем поговорим

- Мы привыкли мерить latency, но не измеряем скорость коммуникаций между командами
- Досконально считаем RPS, но редко измеряем когнитивную нагрузку на команду
- Знаем, сколько было 5xx ошибок, но не сбоев в бизнес-процессах
- Максимизируем uptime и мечтаем о 99,99%, но не знаем, как измерить вовлеченность

Цель доклада

Цель доклада

- Оргструктура — это система, а не просто оргчарт. В любой системе важно взаимодействие. А зачастую над этим не думают, пока не станет больно.

Что получите

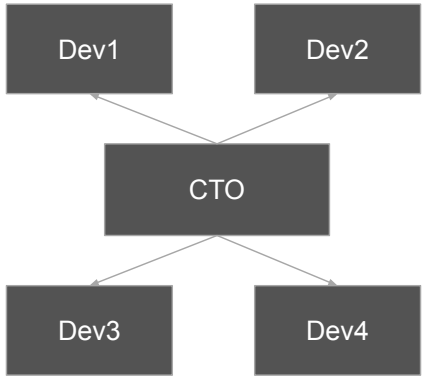
- Идеи и инструменты, как понять, где теряется эффективность и как это исправить (на базе фреймворка Team Topologies).

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

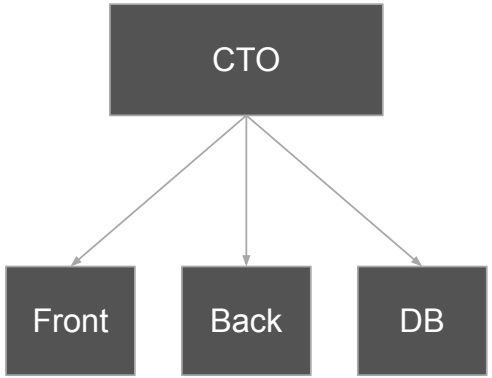
Типовые орг. структуры и какие типовые боли с ними связаны

Типовые орг. структуры в ИТ

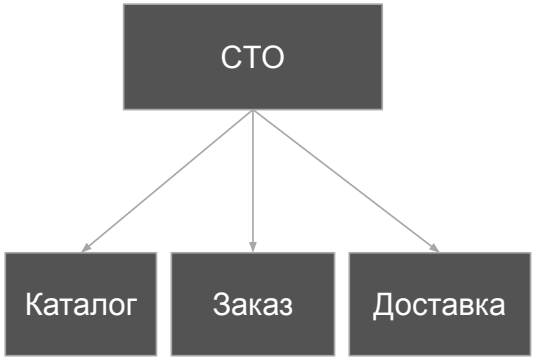
- Обычно оргструктуры в ИТ изменяются вместе с нагрузкой на команду - от плоской, к специализированной (по компетенции) или по функции



Плоская



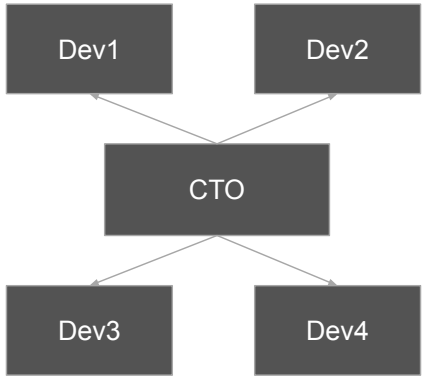
По компетенции



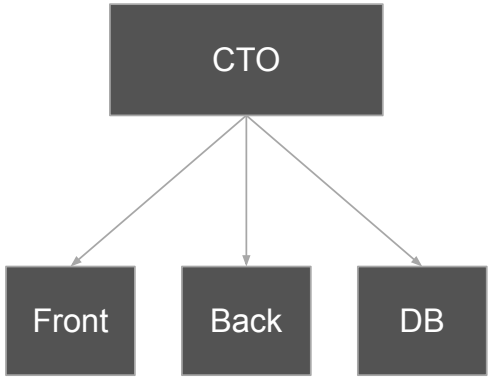
По домену

Типовые орг. структуры в ИТ

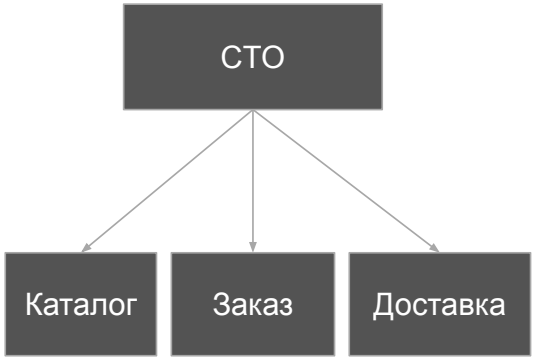
- Обычно оргструктуры в ИТ изменяются вместе с нагрузкой на команду - от плоской, к специализированной (по компетенции) или по функции
- Следующий логичный шаг - матрица



Плоская

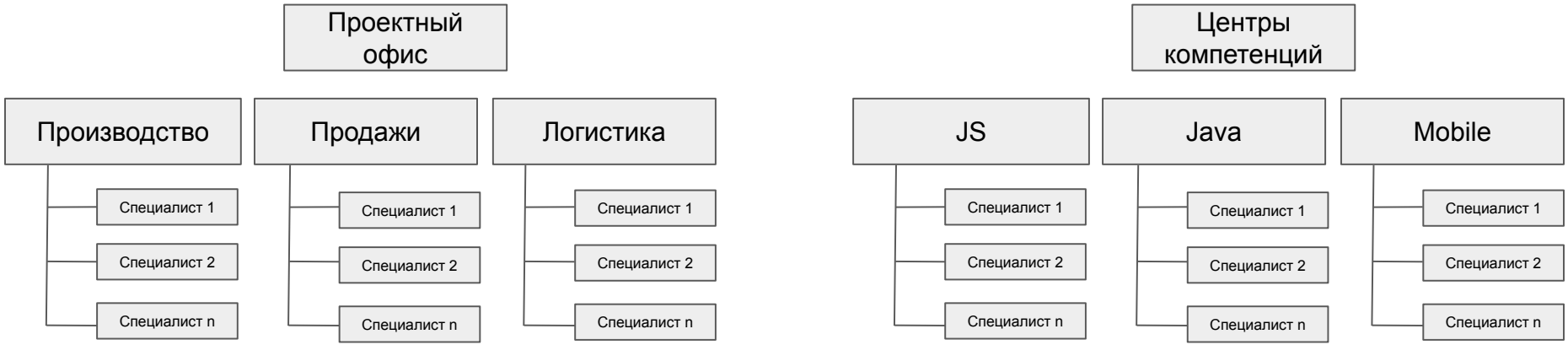


По компетенции

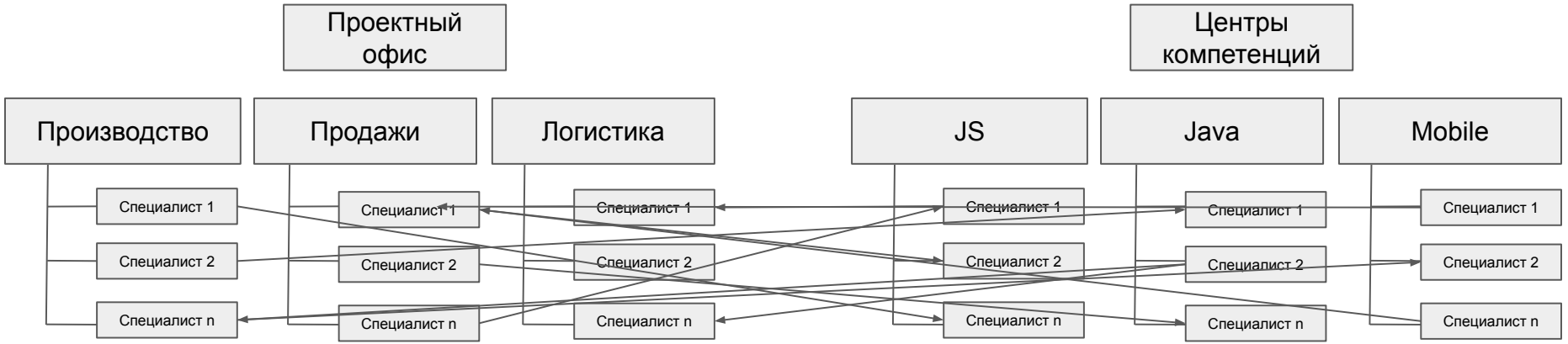


По домену

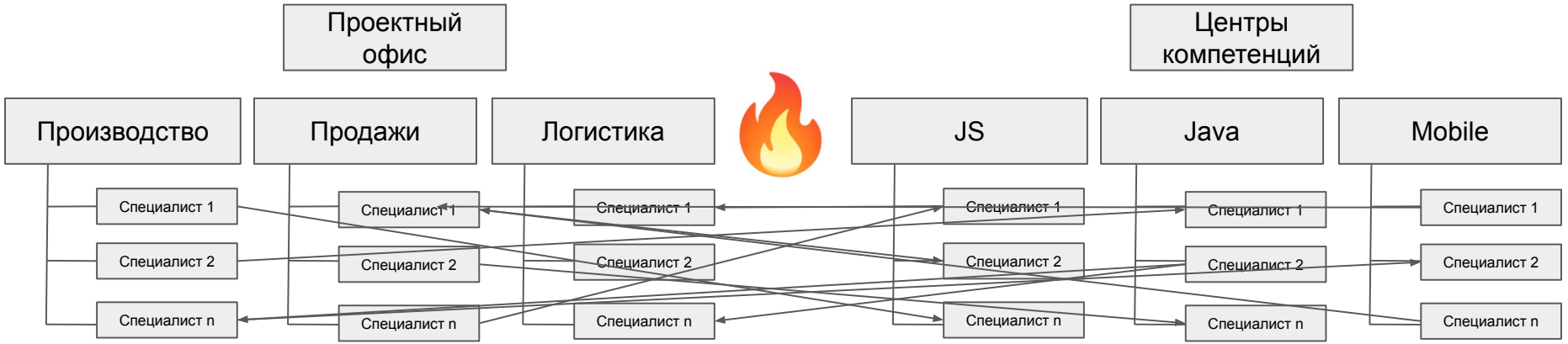
Все - и сразу: матрица



Все - и сразу: матрица



Все - и сразу: матрица



коммуникация many to many =
самый дорогой из возможных способов

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Большинство проблем проектов - не внутри задач, а на стыках между командами

Большинство проблем проектов - не внутри задач, а на стыках между командами

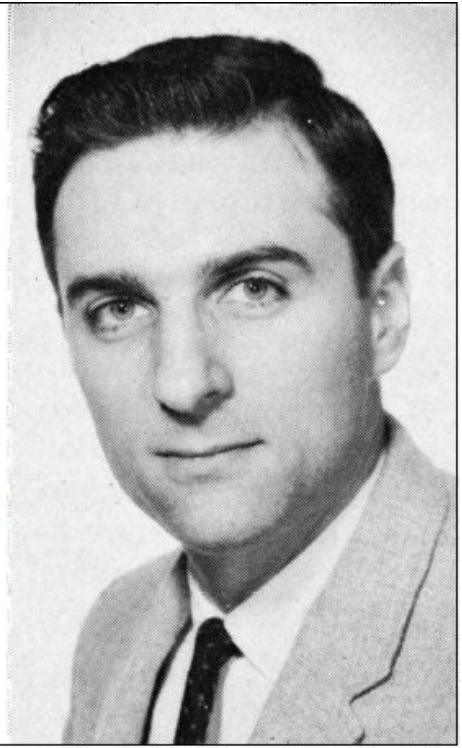
… И это не проблема людей, скорее проблема проектирования

Получается, мы своими руками закладываем проблемы в оргдизайн, которые потом мешают нам самим работать?

Закон Конвея

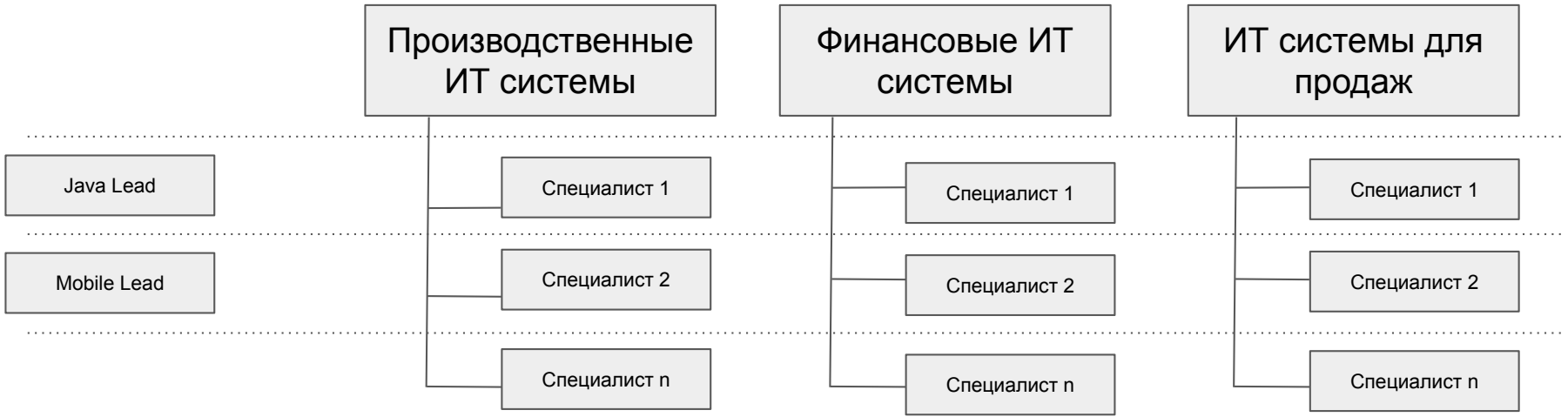
«Организации, проектирующие системы, ограничены дизайном, который копирует структуру коммуникации в этой организации»

Мелвин Конвей



Нельзя просто так выйти из матрицы

- Чтобы компенсировать минусы (отсутствие стандартов и т.д.), мы опять создаем матрицу.



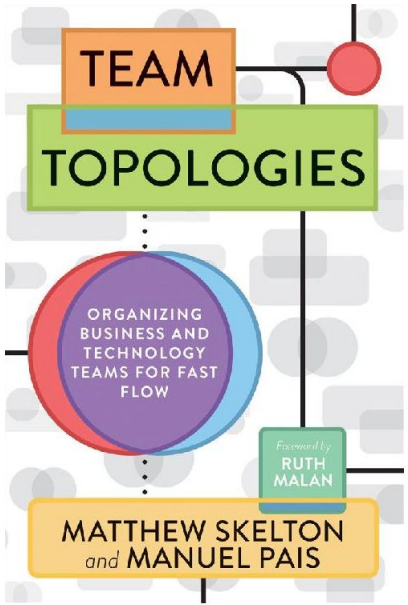
Если изменить структуру, но не изменить правила взаимодействия, проблемы остаются.

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Team Topologies

и как этот подход может помочь

Team Topologies:
Оргдизайн как инженерное мышление



Идея проста: думать о командах как о частях системы, а не строках в орг чарте

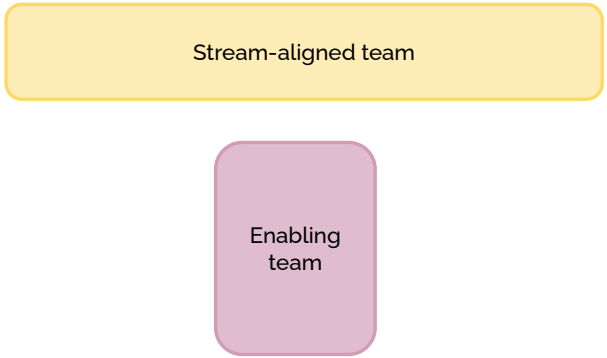
- Команды, как независимые сервисы
- Границы важны: подчеркивают ценность и ответственность
- Измеряем насколько быстро проходит изменение/запрос через команду, а не “сколько ресурсов выделено”
- Формализуем взаимодействие (Team API)

Team Topologies: Четыре типа команд

Stream-aligned team

Кросс-функциональная команда
Команда, полностью ориентированная на один поток ценности: продукт, фичу, пользовательский сегмент или бизнес-направление. Имеет все необходимые компетенции для автономной работы от идеи до эксплуатации.

Team Topologies: Четыре типа команд



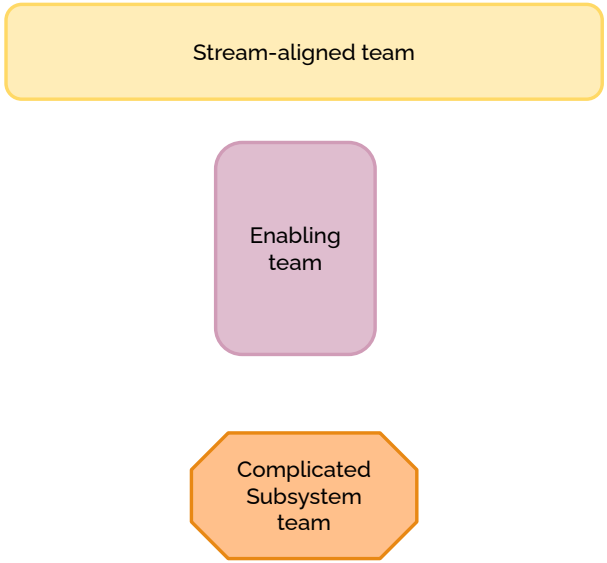
Кросс-функциональная команда

Команда, полностью ориентированная на один поток ценности: продукт, фичу, пользовательский сегмент или бизнес-направление. Имеет все необходимые компетенции для автономной работы от идеи до эксплуатации.

Команда поддержки / развития

Команда, помогающая другим командам осваивать новые практики, инструменты или технологии.
Не делает за них, а обучает, консультирует, устраняет барьеры.

Team Topologies: Четыре типа команд



Кросс-функциональная команда

Команда, полностью ориентированная на один поток ценности: продукт, фичу, пользовательский сегмент или бизнес-направление.
Имеет все необходимые компетенции для автономной работы от идеи до эксплуатации.

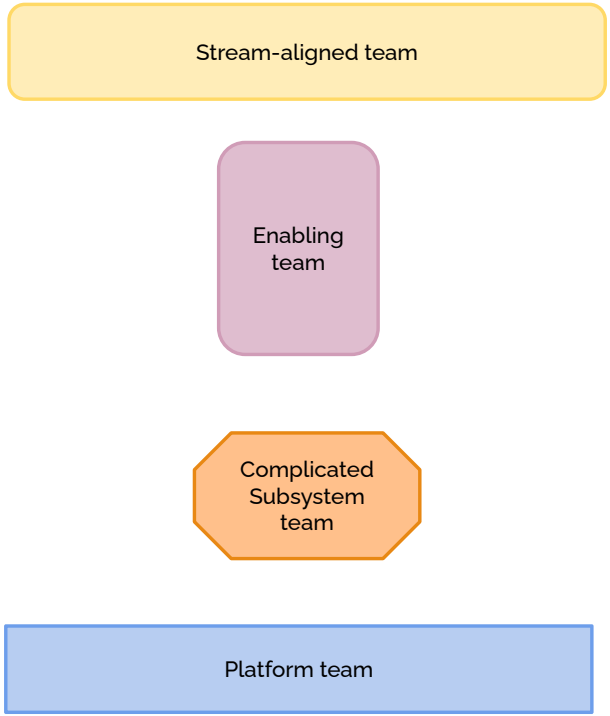
Команда поддержки / развития

Команда, помогающая другим командам осваивать новые практики, инструменты или технологии.
Не делает за них, а обучает, консультирует, устраняет барьеры.

Команда сложных систем

Команда, работающая над технологически сложной или критически важной частью системы, требующей высокой специализации.
Обычно отвечает за стабильность и эволюцию этой подсистемы.

Team Topologies: Четыре типа команд



Кросс-функциональная команда

Команда, полностью ориентированная на один поток ценности: продукт, фичу, пользовательский сегмент или бизнес-направление.
Имеет все необходимые компетенции для автономной работы от идеи до эксплуатации.

Команда поддержки / развития

Команда, помогающая другим командам осваивать новые практики, инструменты или технологии.
Не делает за них, а обучает, консультирует, устраняет барьеры.

Команда сложных систем

Команда, работающая над технологически сложной или критически важной частью системы, требующей высокой специализации.
Обычно отвечает за стабильность и эволюцию этой подсистемы.

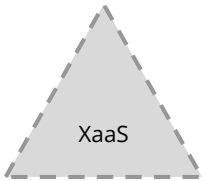
Платформенная команда

Команда, создающая и поддерживающая внутренние сервисы и инструменты, которые используют другие команды.

Team Topologies: Три типа взаимодействия

Плотное взаимодействие (high bandwidth)

Две команды временно тесно сотрудничают, чтобы решить сложную задачу, снизить неопределённость или освоить новую область.
Высокая плотность общения, идеально - в очном режиме. Важно — это временный и осознанный союз, а не перманентная зависимость.



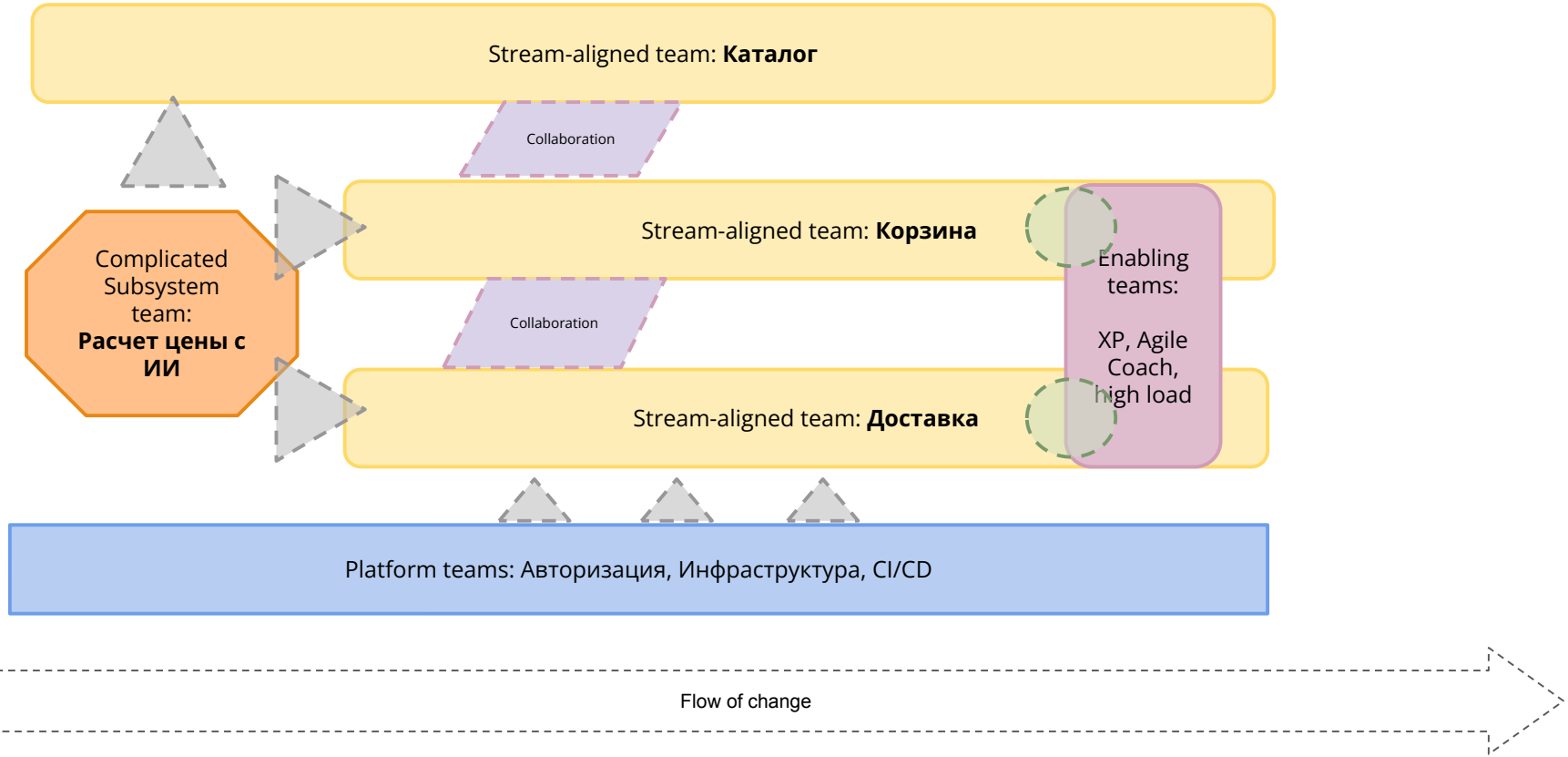
X as a Service (low bandwidth)

Одна команда предоставляет другой стандартизированный сервис, инструмент или платформу — с чёткими интерфейсами, документацией и правилами использования.
Взаимодействие минимальное, по принципу “подключил — и работает”.

Фасилитация

Одна команда помогает другой стать лучше — через менторство, обучение, устранение препятствий.
Цель — развить автономность, а не делать за них.

Team Topologies: Пример (eComm)



стратоплан
ШКОЛА МЕНЕДЖМЕНТА

А когда будет про устойчивые команды?

Что такое устойчивая команда?

Устойчивая команда - это команда, которая предсказуемо доставляет результат

+

сохраняет это свойство при росте нагрузки / изменениях / потере людей.

Из чего состоит время разработки?

Delivery time =

Из чего состоит время разработки?

Delivery time = Build time + Wait time + Decision time

Из чего состоит время разработки?

Delivery time = Build time + Wait time + Decision time

10% 70% 20%

Из чего состоит время разработки?

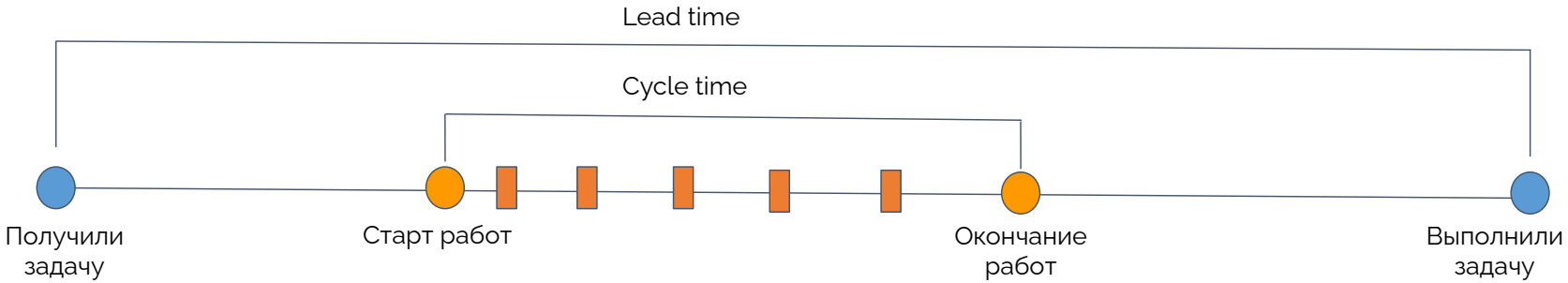
Delivery time = Build time + Wait time + Decision time

10% 70% 20%

При росте масштаба, в хрупких/неустойчивых структурах в первую очередь растет wait + decision time

Wait time: представим его визуально

- Время выполнения задачи - lead time (от получили до выполнили)
- Фактические работы - оранжевые
- Все остальное - ожидания
  - Либо внутри команды (чаще)
  - Либо внешние (реже)

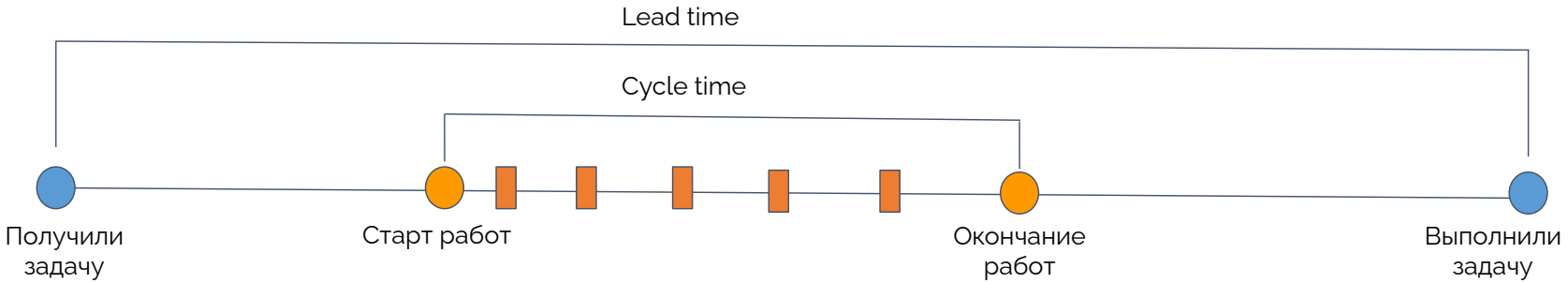


Насколько устойчива моя команда?

Прокси метрика, которая может быть индикатором устойчивости:

Стабильность потока =

p90 lead time (срок 10% самых долгих задач) / p50 lead time (медиана lead time)



Насколько устойчива моя команда?

- Пример: В среднем за последние 90 дней команда закрывала однотипные задачи:
  - В среднем (медиана) - типовая задача занимает 3 дня
  - Но 10% самых медленных задач (хвост) - в среднем занимает 15 дней

$$p90 / p50 = 5$$

Насколько устойчива моя команда?

- Пример: В среднем за последние 90 дней команда закрывала однотипные задачи:
  - В среднем (медиана) - типовая задача занимает 3 дня
  - Но 10% самых медленных задач (хвост) - в среднем занимает 15 дней

$$p90 / p50 = 5$$

- Примерные ориентиры:
  - Зелёный уровень <2 (идеал = 1)
  - Желтый уровень <3
  - Красный уровень >3

Wait time = org. latency

- Мы привыкли мерить latency, но не измеряем скорость коммуникаций между командами

Wait time = org. latency

- Мы привыкли мерить latency, но не измеряем скорость коммуникаций между командами

Идеи для метрик wait time

Внешние зависимости:

- Общее время в blocked by / waiting
- Количество переходов между отделами при поставке

Wait time = org. latency

- Мы привыкли мерить latency, но не измеряем скорость коммуникаций между командами

Идеи для метрик wait time

Wait внутри команды:

- Work In Progress (WIP) на разработчика
- WIP aging (мало/средне/много)
- “Протухшие задачи” - без движения долгое время

- Досконально считаем RPS, но редко измеряем когнитивную нагрузку на команду

Как понять уровень когнитивной нагрузки в команде?

WIP = прокси к когнитивной нагрузке

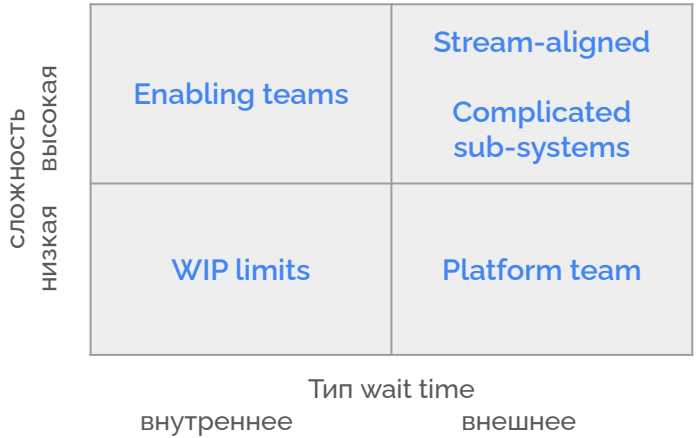
- Досконально считаем RPS, но редко измеряем когнитивную нагрузку на команду

Как понять уровень когнитивной нагрузки в команде?

- Work In Progress (WIP) на разработчика
- WIP aging (мало/средне/много)
- “Протухшие задачи” - без движения долгое время

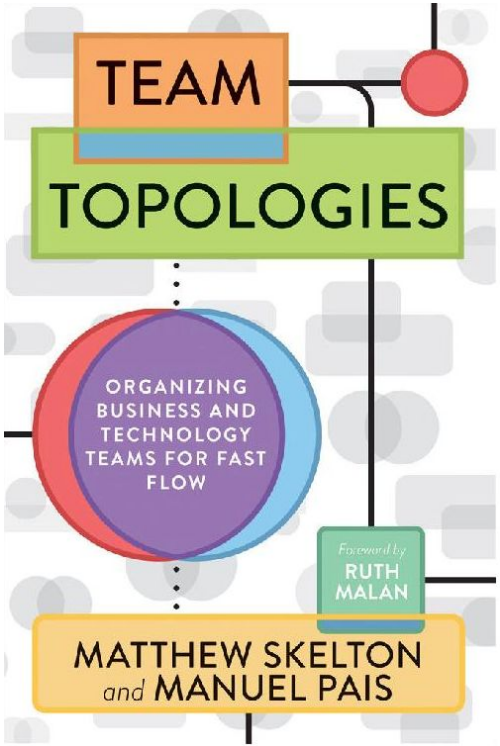
Как оптимизировать wait time?

- Stream aligned team - меньше зависимостей от других команд (снижаем внешний wait)
- Platform/Complicated subsystem переводят часть зависимостей в self-service и увеличивают предсказуемость, разгружая stream-aligned команды.
- Enabling team - помогают бороться с сложными и “протухшими” задачами
- WIP limits - ограничиваем параллельность, растет предсказуемость (p50/p90)



Критика Team Topologies

- Не решена проблема раздробленности организации: наоборот, TT поощряет локальную оптимизацию, а не всей системы.
- Слишком инженерно. Люди — не компьютеры, и архитектурные подходы в IT (API, etc.) не работают, если не работать над культурой.
- Это не завершенный фреймворк - только идеи, без указания “что делать”



стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Что можно сделать уже завтра?

Что можно сделать уже завтра?

- Посчитать p90/p50 на типовых задачах для вашей команды

- Оценить внутреннее (WIP, aging, “протухшие” задачи) и внешнее время wait time

- Попробовать взглянуть на свою организацию и команду, как на поток, а не как на “квадратики в оргчарте”

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Подведем итоги

Итоги

- Проблемы производительности в организациях - это проблемы взаимодействия. Мы предполагаем, что оно будет, а ему нужно учить
- Зачастую мы сами закладываем проблемы в работу - через орг дизайн
- Устойчивая команда - та которая, предсказуемо выдает результат в различных условиях
- Ключом к управлению предсказуемости является wait time

Вопрос в конце - на саморефлексию

Представьте, что завтра в вашей компании отменили половину sync встреч. Как это повлияет на результат?

1. Все остановится: значит скорость держалась на ручной координации, много зависимостей, внутреннего и внешнего wait
2. Упадет качество: значит на встречах определяются границы решений “на стыках” возможно стоит их четче очертить
3. Упадет контроль: это сигнал о том, что работает ручное управление
4. Ничего плохого не случится: отлично, значит отменяем половину встреч

Этот вопрос не про плохих людей или процессы - а про то, как встречи подменяют организационную систему и становятся “костылем” взаимодействий

стратоплан
ШКОЛА МЕНЕДЖМЕНТА

Спасибо за внимание!

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