

Менеджмент 360

Что такое операционная система компании?

Задачи и инструментарий COO

Александр Апазиди
04.09.2026



Представляюсь!

Александр Апазиди

СТО / 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

Давайте представим типовую компанию

Среднего размера ИТ-аутсорсер



600 сотрудников



40–50 проектов

одновременно



T&M и fix price

две разные логики поставки



Цель: рост ×2

без удвоения
управленческого слоя

Давайте представим типовую компанию

Среднего размера ИТ-аутсорсер



600 сотрудников



40–50 проектов

одновременно



T&M и fix price

две разные логики поставки



Цель: рост ×2

без удвоения
управленческого слоя

**Функции зрелые, руководители опытные, показатели есть.
Дальше начинается самое интересное.**

Все большие **молодцы**



Коммерческий
директор

План новых
контрактов
+25%



Директор по
производству

92% вех в срок



Разработка

Поставка
предсказуемое,
дефектов меньше



HR

Команды
комплекуются
в согласованный
срок



Закупки / Подрядчики

Снизили среднюю
ставку
-8%

Все большие молодцы



Коммерческий
директор

План новых
контрактов
+25%



Директор по
производству

92% вех в срок



Разработка

Поставка
предсказуемее,
дефектов меньше



HR

Команды
комплектуются
в согласованный
срок



Закупки /Подрядчики

Снизили среднюю
ставку
-8%

Каждый руководитель может честно **показать улучшение в своей зоне ответственности**

Но своих целей компания **не достигает**



Прибыльность

ниже плана



От сделки до денег

дольше



Ручная координация

**растёт быстрее
бизнеса**



СЕО

**разруливает все
конфликты**

Но своих целей компания **не достигает**



Прибыльность

ниже плана



От сделки до денег

дольше



Ручная координация

растёт быстрее
бизнеса



SEO

разруливает все
конфликты

Вопросов традиционно 2:

- 1) Кто виноват?
- 2) Что делать?

Каждый производит **свой** результат

Поэтому локально разумные решения не обязаны складываться в результат компании

Продажи



подписанный договор

Предпродажная подготовка



готовая оценка и предложение

HR



найденные и выведенные люди

Разработка



работающее решение

Производство



выполненная контрольная точка

Финансы



выставленный и оплаченный счёт



Компания

**принятый клиентом
результат, который
принёс деньги**

и не разрушил способность
работать завтра

Почему вообще появляется СОО

CEO нужен партнёр по управлению

- исполнитель стратегии
- партнёр или “вторая половина”
- другой слой внимания
- освобождение CEO от постоянной маршрутизации

Почему вообще появляется COO

CEO нужен партнёр по управлению

- исполнитель стратегии
- партнёр или “вторая половина”
- другой слой внимания
- освобождение CEO от постоянной маршрутизации

Компании нужно стать другой

- рост или масштабирование
- кризис и разворот
- новая бизнес-модель
- новый рынок или география

Почему вообще появляется COO

CEO нужен партнёр по управлению

- исполнитель стратегии
- партнёр или “вторая половина”
- другой слой внимания
- освобождение CEO от постоянной маршрутизации

Компании нужно стать другой

- рост или масштабирование
- кризис и разворот
- новая бизнес-модель
- новый рынок или география

Подготовка будущего лидера

- будущий CEO
- развитие сильного руководителя
- преемственность
- удержание критически важного лидера

Почему вообще появляется COO

CEO нужен партнёр по управлению

- исполнитель стратегии
- партнёр или “вторая половина”
- другой слой внимания
- освобождение CEO от постоянной маршрутизации

Компании нужно стать другой

- рост или масштабирование
- кризис и разворот
- новая бизнес-модель
- новый рынок или география

Подготовка будущего лидера

- будущий CEO
- развитие сильного руководителя
- преемственность
- удержание критически важного лидера

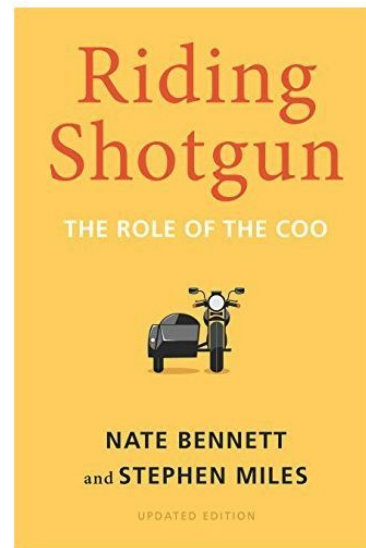
Сегодня — про COO, который отвечает за способность компании реализовать стратегию и пережить следующий масштаб.

Riding shotgun

Метафора роли COO

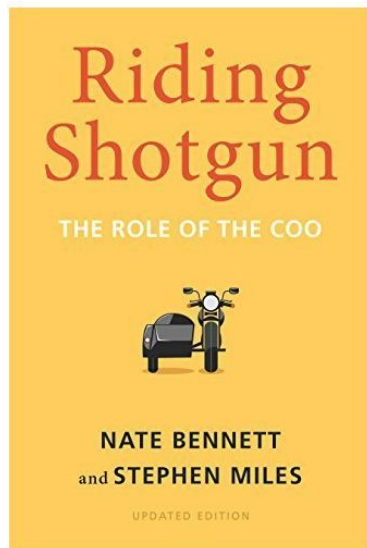
Riding shotgun — ехать рядом с водителем

Метафора роли COO



Riding shotgun — ехать рядом с водителем

Метафора роли COO



CEO выбирает путь и может вмешаться в любой вопрос



COO видит ту же дорогу, но постоянно держит внимание на других аспектах по пути



COO не пассажир — и не второй водитель, который всё время вырывает руль



Границы решений и источник мандата должны быть договорены заранее

Реальные СОО описывают работу так

“Говорить нет”

“Question everything! Why?”

“Гденьги?”

“Синтез противоречивых сведений”

“Давление в системе”

“Мы на одной стороне”

Нужно сильно менять систему, часто не имея прямой власти над всеми её частями.

В этом и есть главный конфликт роли.

Две естественные ошибки СОО

✕ “Сейчас я вам построю нормальную компанию”

Новые показатели, комитеты, регламенты, обязательная методология.

Получаем:

- формальное согласие
- параллельные таблицы
- скрытое сопротивление
- система работает только под личным давлением СОО

Две естественные ошибки СОО

“Сейчас я вам построю нормальную компанию”

Новые показатели, комитеты, регламенты, обязательная методология.

Получаем:

- формальное согласие
- параллельные таблицы
- скрытое сопротивление
- система работает только под личным давлением СОО

“Коллеги, давайте договоримся”

Больше встреч, больше фасилитации, поиск полного согласия, разговоры об общей цели.

Получаем:

- хорошие отношения
- длинные обсуждения
- отложенные решения
- старое устройство компании

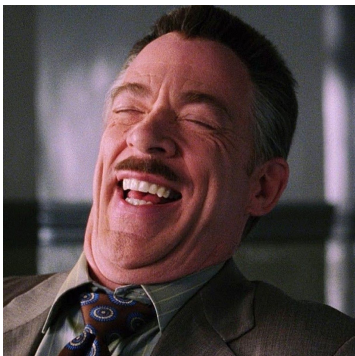
Две естественные ошибки СОО

✖ “Сейчас я вам построю нормальную компанию”

Новые показатели, комитеты, регламенты, обязательная методология.

Получаем:

- формальное согласие
- параллельные таблицы
- скрытое сопротивление
- система работает только под личным давлением СОО



👥 “Коллеги, давайте договоримся”

Больше встреч, больше фасилитации, поиск полного согласия, разговоры об общей цели.

Получаем:

- хорошие отношения
- длинные обсуждения
- отложенные решения
- старое устройство компании



Рабочая позиция СОО — между принуждением и бесконечным консенсусом

Гениальный продукт — ещё не операционная система



Компания создаёт выдающийся продукт.

Гениальный продукт — ещё не операционная система



Компания создаёт выдающийся продукт.

Теперь его нужно:



обеспечить компонентами



произвести миллионами
штук



доставить по миру



продать



обслужить



повторить со следующим
поколением

Гениальный продукт — ещё не **операционная система**



Компания создаёт выдающийся продукт.

Теперь его нужно:



обеспечить компонентами



произвести миллионами
штук



доставить по миру



продать



обслужить



повторить со следующим
поколением

Продуктовая гениальность не отменяет задачу построить повторяемую “скучную”
машину исполнения.

Тим Кук: Riding shotgun → CEO



🔧 Продуктовый контур

- продуктовые решения
- дизайн и пользовательский опыт
- стандарты качества
- запуск продукта

🏢 Операционная система

- мировая цепочка поставок
- планирование и закупки
- производство и качество
- логистика и выполнение заказов
- продажи, сервис и поддержка

**Стив Джобс мог вникнуть в любую деталь
Тим Кук строил систему, которая должна работать каждый день**

Инструментарий COO



1. **Операционная стратегия задаёт требования к системе**



Бизнес-замысел

Что хотим получить
на рынке и для клиента



Способности компании

Что нужно уметь
повторять и
масштабировать



Операционная система

Потоки, мощности,
интерфейсы, данные,
полномочия, обратная связь

Операционная стратегия: кейс Metro C&C



2. Value Stream Mapping:

Что производит ваша компания?

Возьмем к примеру нашего ИТ-аутсорсера

Код?

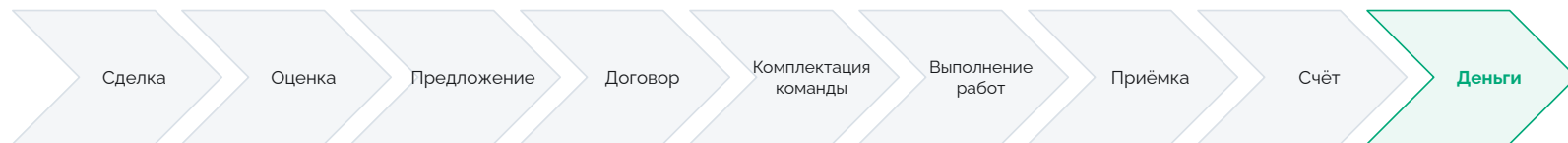
Проданные часы?

Проект?

Результат?

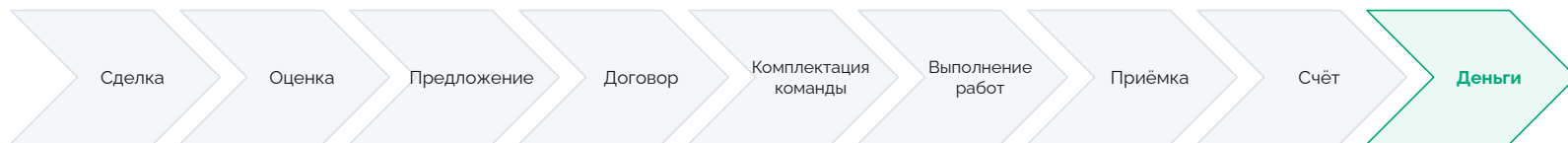
2. VSM: Что производит ваша компания?

Пример для нашего ИТ аутсорсера



2. VSM: Что производит ваша компания?

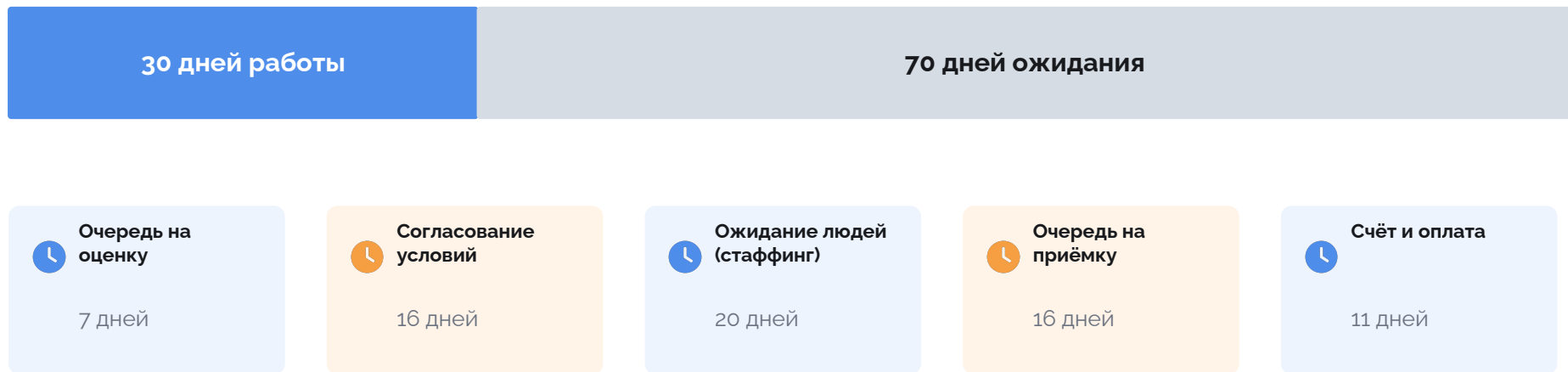
Пример для нашего ИТ аутсорсера



Единица ценности: **принятый клиентом и оплаченный результат**

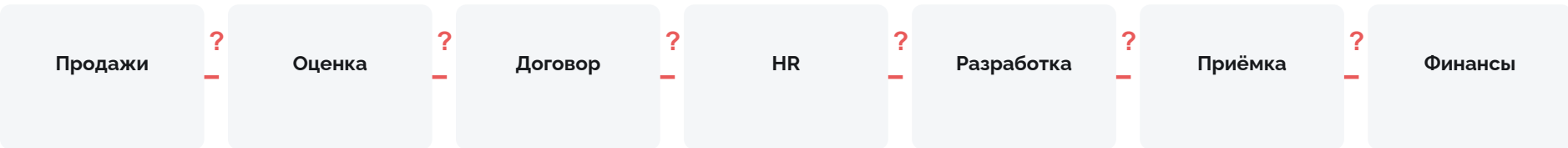
Зачем нужен VSM? Рассмотрим на примере

Работа занимает 30 дней. Клиент ждёт 100



Ожидание в > x2 раза больше времени реальной работы.

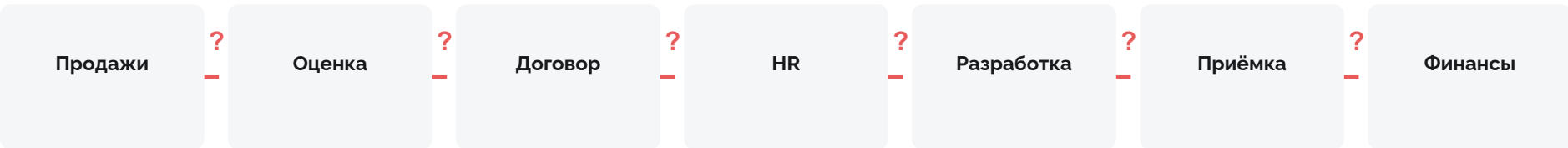
А кто отвечает за эти 70 дней ожидания?



У каждой функции есть владелец работы.

У очереди между функциями владельца часто нет.

А кто отвечает за эти 70 дней ожидания?



У каждой функции есть владелец работы.

У очереди между функциями владельца часто нет.

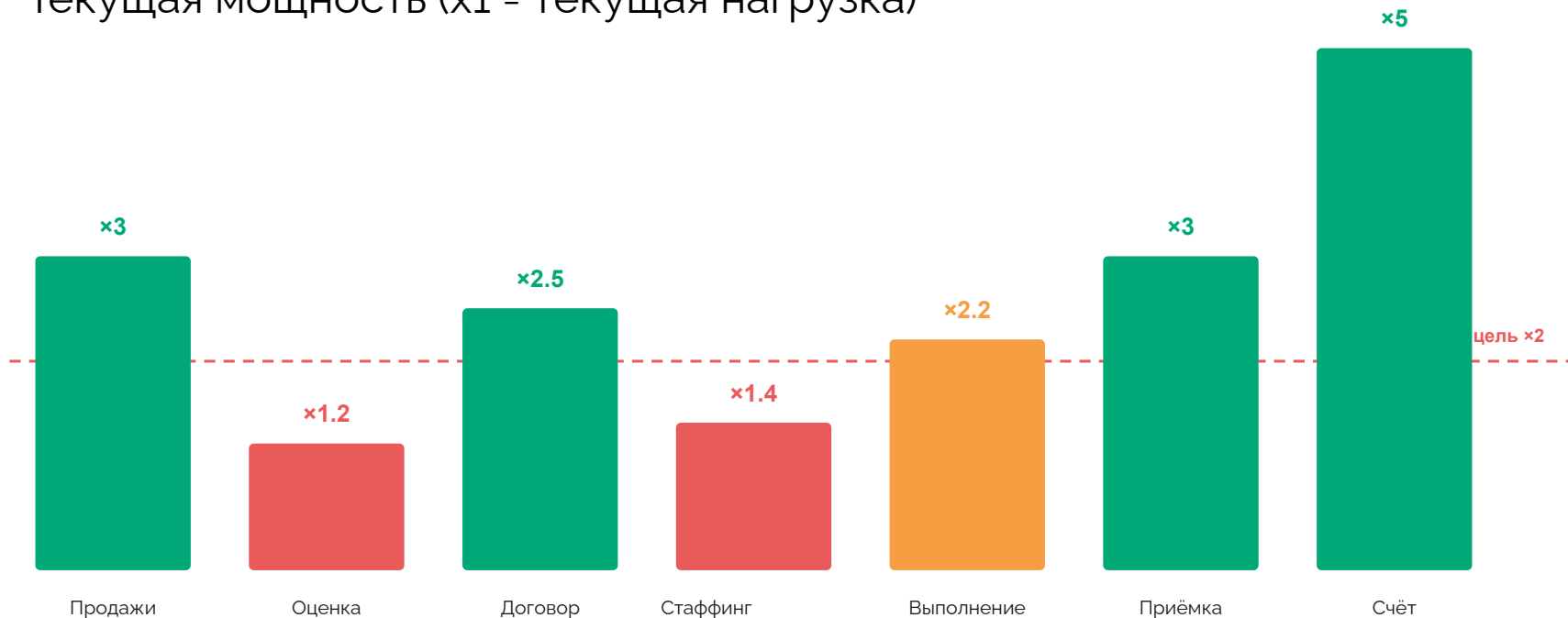
СОО — владелец не отдельных этапов, а скорости всего потока

Теперь хотим **вырасти в два раза**

Первая мысль-рефлекс: ок, нам нужно будет нанять в два раза больше разработчиков

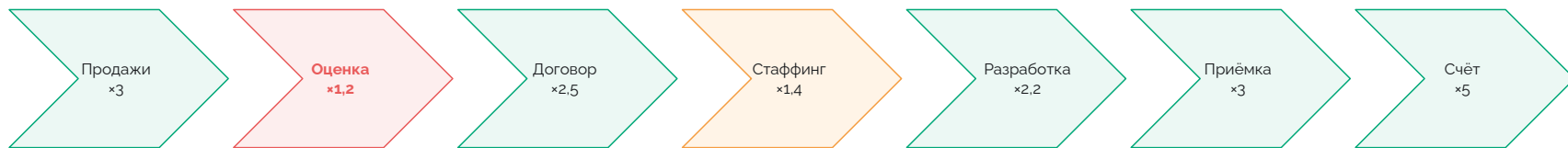
Теперь хотим вырасти в два раза

Текущая мощность (x_1 = текущая нагрузка)



Запас мощности уже есть, но не везде одинаковый — и расширять “всё подряд” дорого и бесполезно.

Руководители функций **будут просить x2,** но не везде он нужен



Первое
ограничение

После
расширения

И только
после этого
будет разработка

3. Бизнес-процессы

А как именно сократить ожидание?

30 дней работы

70 дней ожидания



Очередь на
оценку

7 дней



Согласование
условий

16 дней



Ожидание людей
(стаффинг)

20 дней



Очередь на
приёмку

16 дней



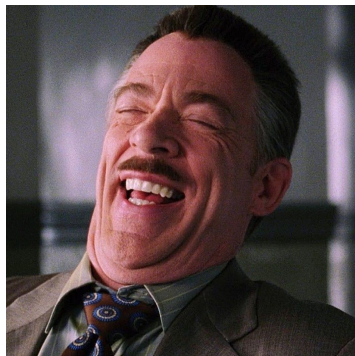
Счёт и оплата

11 дней

А как именно сократить ожидание?

Самый естественный ответ:

“Пусть люди отвечают быстрее”



А как именно сократить ожидание?

Самый естественный ответ:

“Пусть люди отвечают быстрее”

Но ожидание чаще возникает из-за устройства интерфейса:



Нет критериев
входа



Нет владельца
решения



Согласования идут
последовательно



Стандартные
случаи смешаны с
исключениями

А как именно сократить ожидание?

Самый естественный ответ:

“Пусть люди отвечают быстрее”

Но ожидание чаще возникает из-за устройства интерфейса:



Нет критериев
входа



Нет владельца
решения



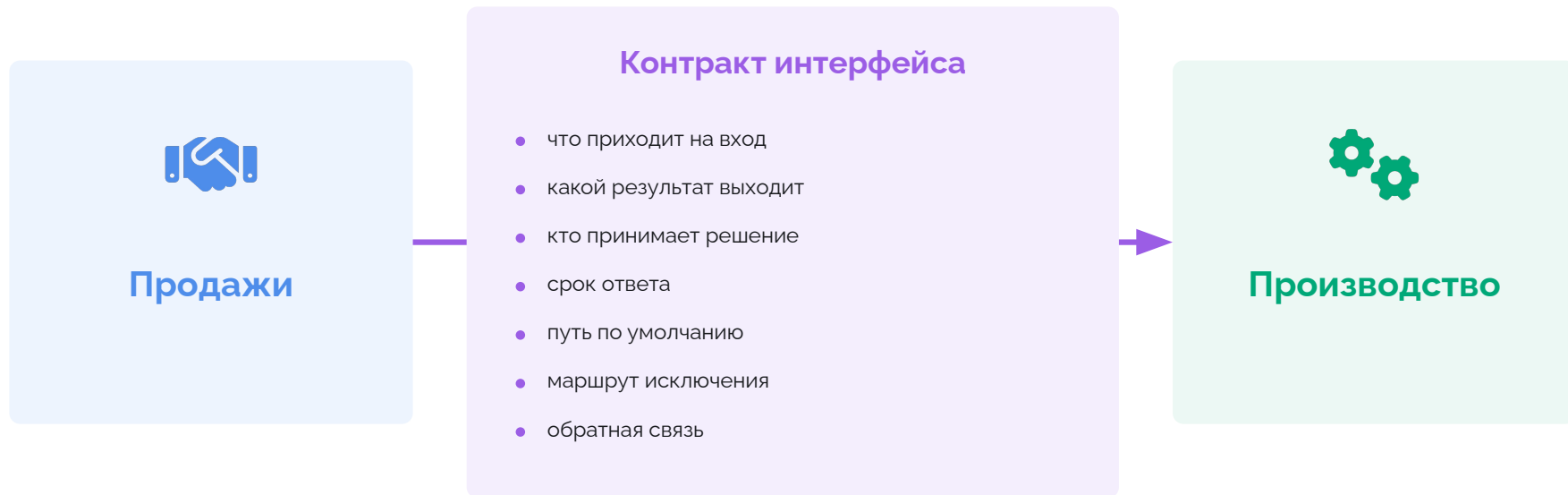
Согласования идут
последовательно



Стандартные
случаи смешаны с
исключениями

**Ускорять человека приказами проще
Перепроектировать взаимодействие — сложнее, но полезнее**

Бизнес-процесс — это как API между подразделениями



Слабые интерфейсы создают очереди даже между сильными функциями.

Пример интерфейса: продажи → производство

× Было

- сроки уже обещаны клиенту
- стек и объём уже названы
- возможность собрать команду не проверена
- критерии приёмки расплывчаты
- нестандартность обнаруживается после договора
- все спорные решения уходят наверх

✓ Стало

- до обещания проверена реализуемость
- стандартные предложения идут по каталогу
- нестандартные — по отдельному маршруту
- подтверждена возможность собрать команду
- критерии приёмки входят в договор
- у исключения есть владелец и срок решения
- часть работ запускается параллельно

Процессное управление — не только про BPMN.
Это рефакторинг процесса: убрать лишние передачи, сделать интерфейсы явными и увеличить мощность потока

Factorio: почему работающий процесс всё равно придётся переделывать



Кстати, Factorio — неплохой тренажёр СОО (если есть свободная пара тысяч часов)

4. Учимся на ошибках

Метафора: мой Inbox — это Dead Letter Queue

4. Учимся на ошибках

Мой Inbox — Dead Letter Queue

Очередь сообщений, которые стандартная система не смогла обработать



“Помоги согласовать”



“Они не отвечают”



“Конфликт приоритетов”



“Непонятно, кто решает”



“Срок уже завтра”

4. Учимся на ошибках

Мой Inbox — Dead Letter Queue

Очередь сообщений, которые стандартная система не смогла обработать



“Помоги согласовать”



“Они не отвечают”



“Конфликт приоритетов”



“Непонятно, кто решает”



“Срок уже завтра”

Что это может означать

нет владельца

нет полномочий

нет данных

нет маршрута исключения

не работает стык

4. Учимся на ошибках: Fix it twice

Первый фикс спасает ситуацию. Второй — систему.

Сигнал из входящих

Первый фикс

Второй фикс

Проект застрял из-за ресурсов

Позвонить СТО и договориться

Сделать мощность видимой, определить правила приоритизации

Клиент не принимает этап

Разрулить спор и сохранить клиента

Включить критерии приёмки в договор и назначить владельца

Срок уже завтра

Найти ручной обход

Создать маршрут исключений и ранний сигнал

4. Учимся на ошибках: Fix it twice

Первый фикс спасает ситуацию. Второй — систему.

Сигнал из входящих

Первый фикс

Второй фикс

Проект застрял из-за ресурсов

Позвонить СТО и договориться

Сделать мощность видимой, определить правила приоритизации

Клиент не принимает этап

Разрулить спор и сохранить клиента

Включить критерии приёмки в договор и назначить владельца

Срок уже завтра

Найти ручной обход

Создать маршрут исключений и ранний сигнал

Культурный код: ошибаться можно. Повторять одну и ту же ошибку — уже дефект системы.

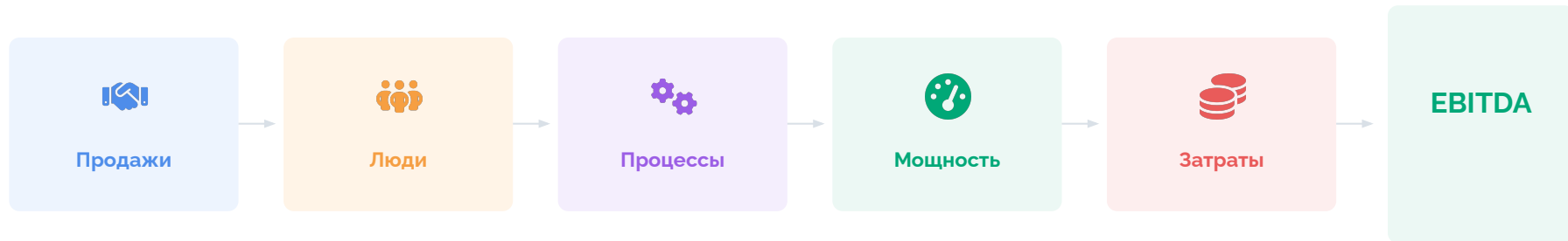
**И всё это легко превратить
в корпоративную физкультуру и ИБД/сибурду**

И всё это легко превратить
в корпоративную физкультуру и ИБД/сибурду

Главным вопросом остается:
Гденьги?

5. Экономика и финансы

Почему **EBITDA** часто становится метрикой COO



Она агрегирует решения нескольких функций, а COO отвечает за механизм, который должен собрать их в общий результат.

EBITDA — датчик операционной системы, а не цель самой системы.

CFO защищает финансовую реальность. COO меняет операционную реальность, из которой появляются цифры.

Одинаковая EBITDA может означать разные вещи

Заработали

- сняли ограничение
- сократили переделки
- увеличили полезный результат
- снизили реальную стоимость работы

Система действительно стала сильнее.

Одолжили у будущего

- отложили обучение
- не наняли заранее
- перенесли обслуживание
- использовали запас мощности

Сегодня лучше — завтра придётся вернуть.

Обменяли на риск

- убрали резерв
- усилили зависимость от ключевых людей
- снизили контроль
- приняли опасный контракт

Дешевле сейчас — уязвимее система.

Второй и третий варианты иногда оправданы. Ошибка — не различать их и выдавать временный эффект за улучшение системы.

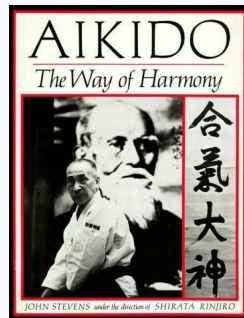
**Так а как это все добро внедрять, если зачастую
формальной власти нет?**

6. От силы контроля — к силе поддержки



Контроль

“Делай как я сказал”



Влияние

“Я меняю направление движения”



Поддержка

“Я меняю условия движения”

Какой способ сильнее?

Контроль: **быстро** двигает задачу



Когда работает

- кризис
- безопасность
- жёсткая граница
- необратимый риск
- нет времени на поиск

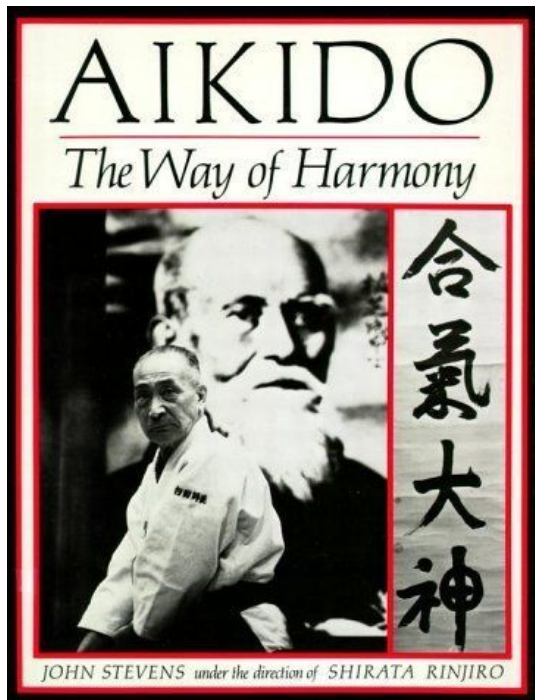
Цена

- все исключения идут наверх
- люди учатся исполнять
- масштаб ограничен вниманием СОО

**Пример: клиентский кризис сегодня.
СОО вправе назначить ответственного и срок — без
фасилитации.**

**Контроль решает инцидент, но редко строит
операционную систему**

Влияние: айкидо вместо лобового удара



- 🎯 Начать с цели и риска другого руководителя
- 👁️ Создать общую базу фактов
- 🔑 Не отбирать у коллеги его профессиональную зону
- ▶️ Предложить ограниченный эксперимент вместо "большой методологии"
- 👤 Оставить авторство решения владельцу функции

Жёсткость — к результату
Недирективность — к способу его достижения

Поддержка: кёрлинг



Камень уже движется. Команда не толкает его руками — она меняет условия движения.



сделать данные
видимыми



защитить
эксперимент



убрать препятствие



создать запас
мощности



дать полномочия



не забирать решение
обратно при первой
ошибке

**Поддержка — не мягкость.
Это создание условий для
самостоятельного результата**

**Так что же такое в итоге операционная система
компании?**

Операционная система компании



Цели и приоритет

операционная стратегия



Ценность и поток

что производим



Наблюдаемость

метрики и экономика



Планирование ресурсов

мощность и ограничения



Компания достигает целей

Повторяемо, масштабируемо, без героизма



Права и управление

контроль, влияние, поддержка



Бизнес-процессы

интерфейсы между функциями



Учимся на ошибках / антихрупкость

входящие / исправь дважды

Операционная система компании

Задача СОО – построить ее, но не стать ее центром



Цели и приоритет

операционная стратегия



Наблюдаемость

метрики и экономика



Планирование ресурсов

мощность и ограничения



Компания достигает целей

Повторяемо, масштабируемо, без героизма



Права и управление

контроль, влияние, поддержка



Ценность и поток

что производим



Бизнес-процессы

интерфейсы между функциями



Учимся на ошибках / антихрупкость

входящие / исправь дважды

Q&A

Александр Апазиди

СТО//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