

рост команды на порядок

*или как не сойти с ума будучи frontend
тимлидом в привлечении Tinkoff.ru*



Александр Поломодов



Saint
Teamlead
Conf

Tinkoff.ru



привлечение tinkoff.ru

бизнес-цели

- много и дешево
привлекать новых клиентов
- удобный
и современный UI/UX
- эффективные
и масштабируемые процессы



технические цели

быстро и гибко управлять контентом

- безрелизно создавать страницы для продуктов
- проводить a/b тесты
- включать персонализации для определенных сегментов

быстро и эффективно создавать рекламные кампании в нужных каналах привлечения

уметь оценивать эффект наших действий

верхнеуровневая архитектура

каналы привлечения

- контекст
- RTB
- emails
- SEO
- tinkoff journal

фронтенд Tinkoff.ru



бекенд



фронтенд Tinkoff.ru

основной канал
коммуникации
с потенциальными
клиентами

СОСТОИТ ИЗ

интернет банка

неавторизированной
зоны

заказчики — все
бизнес unit'ы компании

ДЕБЕТОВЫЕ КАРТЫ

Visa FIFA World Cup™ Участвуй в «Чеканка Челлендж»

- Доход на остаток до 6% годовых
- Кэшбэк 10% на спортивные товары до конца лета

Оформить карту
Узнать подробнее

1/9



Кредитная карта
Тинькофф Платинум

Дебетовая карта
Tinkoff Black

ДЕБЕТОВЫЕ КАРТЫ

УНИКАЛЬНЫЙ ДИЗАЙН

Дебетовая карта со Сберкотом

Вклады

5 000 руб. в месяц До 3,3% 750 руб.

Страхование КАСКО

Получите полис онлайн

>

шаг 0

с чего все начиналось

архитектура фронта

организация кода



монорепозиторий
для всех команд



МОНОЛИТНЫЙ КОД



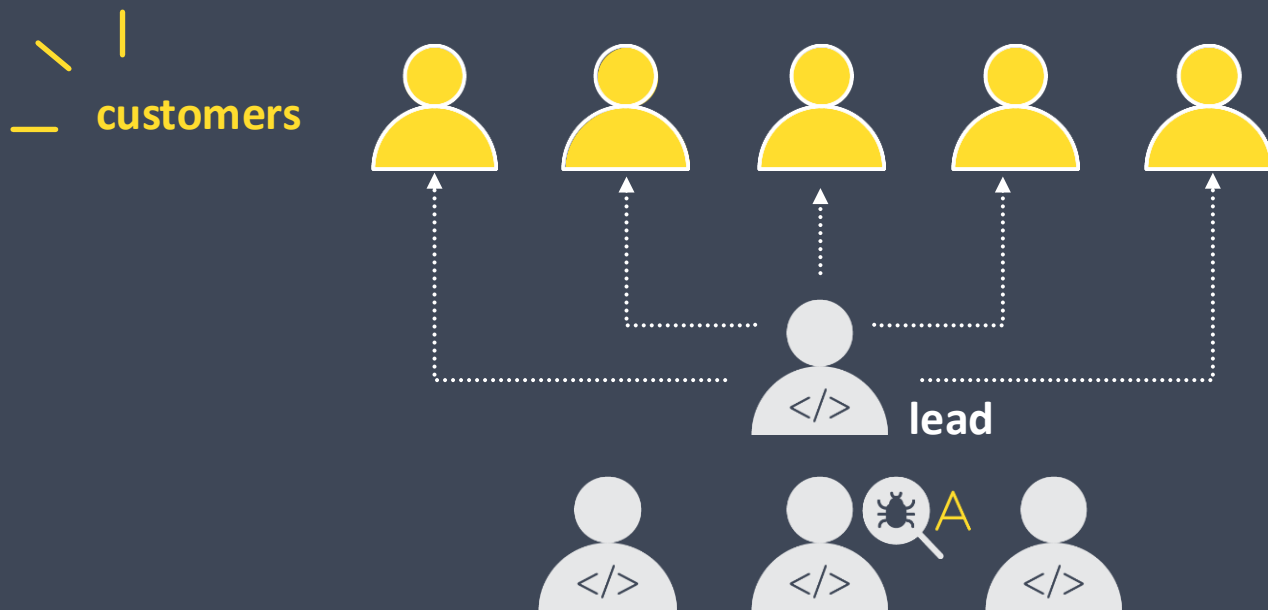
monolithic/layered

эксплуатация

- деплоим общий код на общую инфраструктуру
- обслуживаем им все запросы для всех страниц
- для релиза своих изменений надо закинуть код в общую ветку

структура команды

шаг 0



обязанности тимлида

шаг 0



общение

с заказчиками
напрямую



оценка

всех задач



ревью

всех задач



фикс

найденных
багов во время
общего регресса

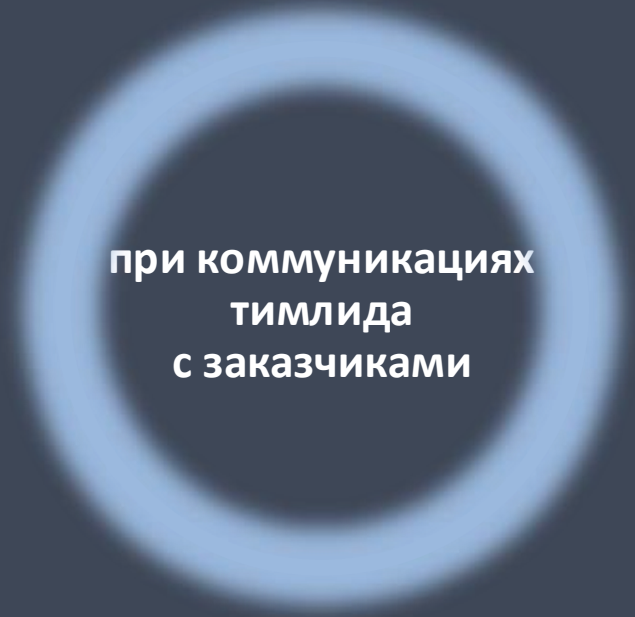
шаг 1

первые улучшения

проблема потери фокуса



при работе
с кодом
и деплоях



при коммуникациях
тимлида
с заказчиками

архитектура фронта

стало

шаг 1

организация кода



монорепозиторий
для всех команд



МОНОЛИТНЫЙ КОД

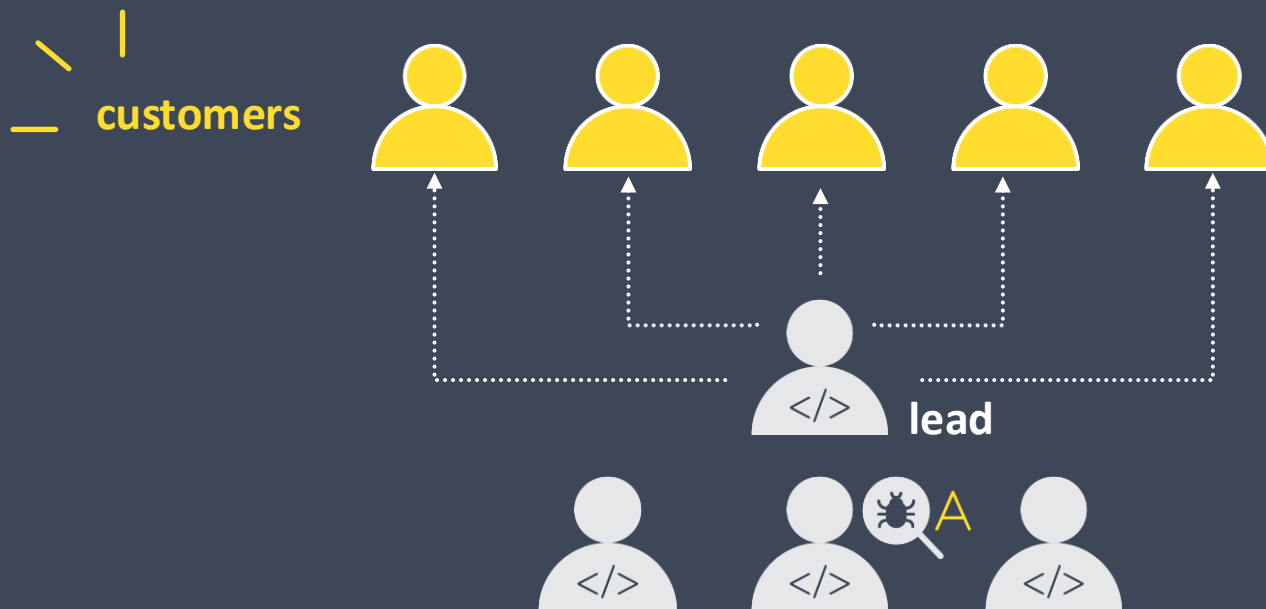


monolithic/layered

- деплоим общий код на **отдельную** инфраструктуру
- обслуживаем **нашим** приложением **наши** страницы
- релиз **своих изменений** из **своей релизной ветки**

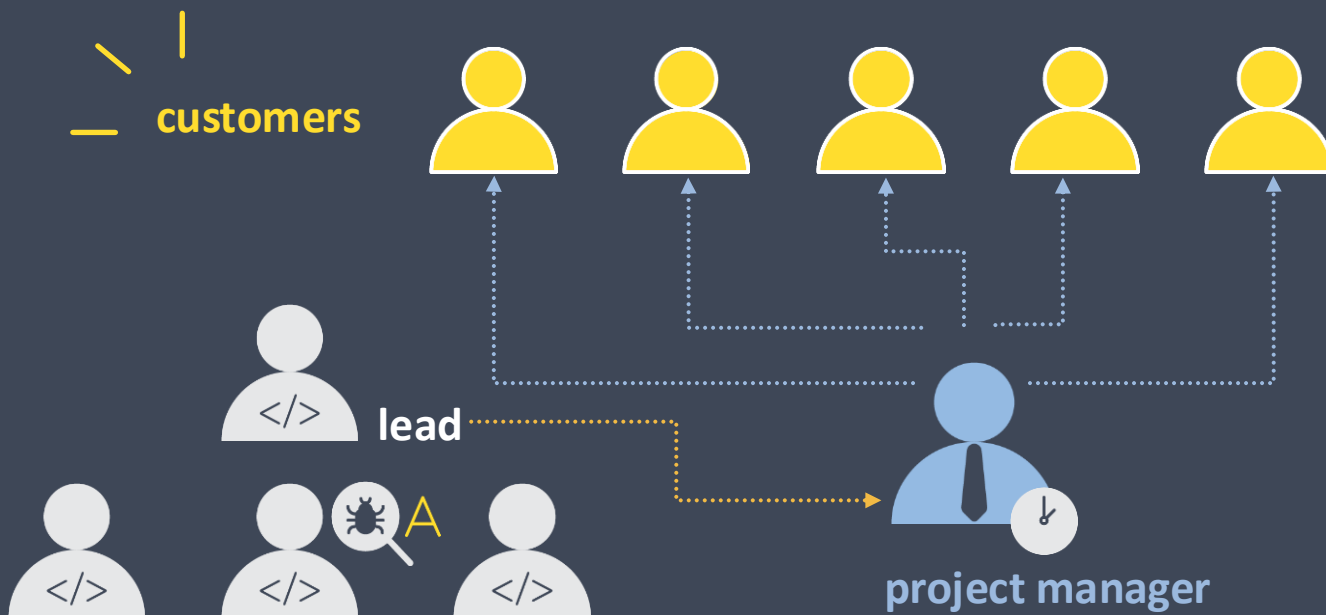
структура команды *было*

шаг 1



структура команды *стало*

шаг 1



обязанности project'a

шаг 1



общение

с заказчиками,
экранирование
команды



организация

процессов разработки
внутри команды:

- процесс работы
с беклогом
- процесс оценки задач
и планирования спринта



решение

проблем
со связанными
командами
(эксплуатация,
тестирование)

обязанности тимлида

шаг 1



общение
с project
manager'ом



оценка
задач
и планирование
спринта



ревью
всех задач



сбор
релизной ветки
и сопровождение
релиза

изменения

шаг 1



коммуникации
разработчиков
с заказчиками
через project'a

The diagram features a large solid yellow circle on the left. To its left, three short yellow lines radiate outwards at different angles. To the right of this circle is a large yellow circle outline. Inside this outline, the text 'отдельные релизы нашего приложения' is written in white. To the right of the yellow circle outline, three small white circles are arranged in a slight arc.

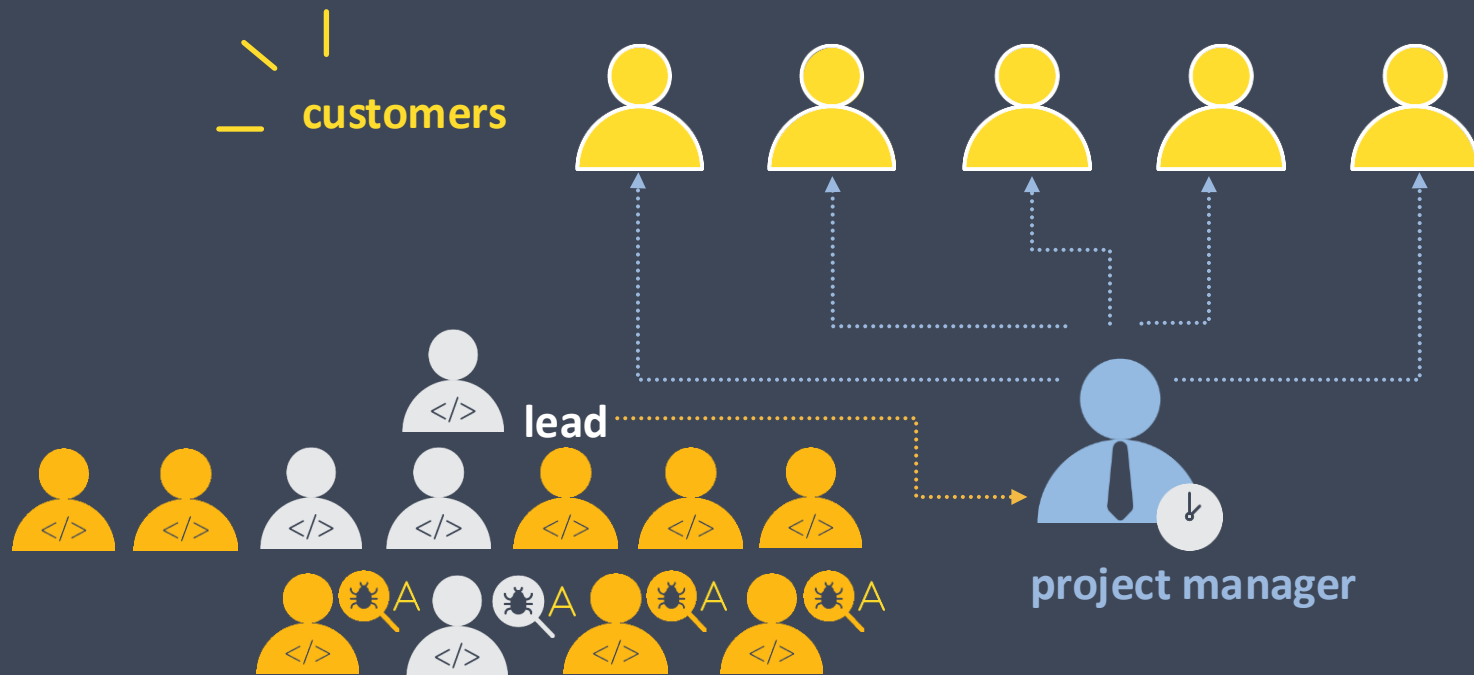
отдельные
релизы нашего
приложения

шаг 2

масштабируемся как есть

проблема масштаба

шар 2



архитектура фронта

стало

шаг 2

организация кода



монорепозиторий
для всех команд



МОНОЛИТНЫЙ КОД



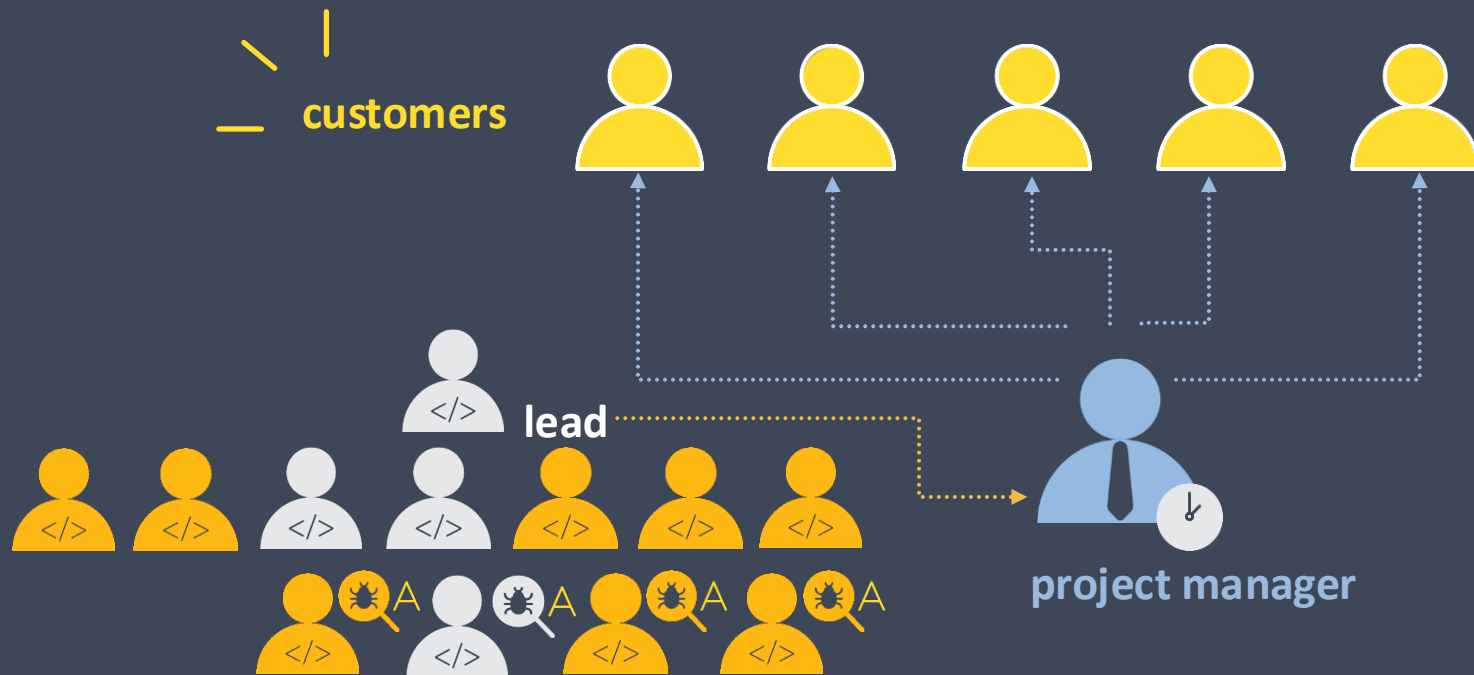
monolithic/layered

эксплуатация

- делим свое приложение на бизнес-части
- деплоим общий код на отдельную инфраструктуру для каждого бизнес-приложения
- обслуживаем своими приложениями свои url'ы
- релиз своих изменений из своей релизной ветки

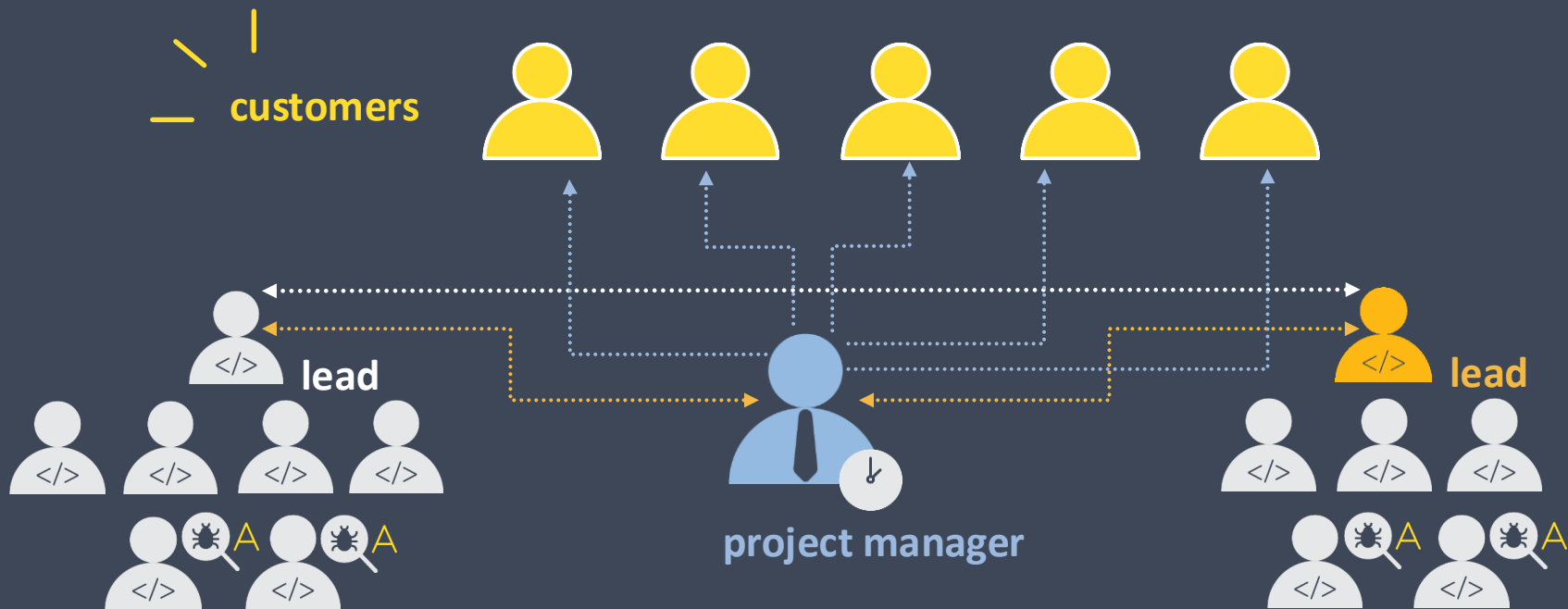
структура команды *было*

шаг 2



структура команды *стало*

шаг 2



обязанности тимлида

шаг 2



общение

с project manager'ом
и другими
тимлидами



организация

процессов разработки
внутри своей команды:

- процесс оценки задач
и планирования спринта
- процесс ревью
задач (peer review)
- процесс
документирования
(задавший вопрос
и получивший ответ
заносят его в FAQ)



обучение и наставничество

для разработчиков
внутри команды



сбор

релизной
ветки
и сопровождение
релиза

как мы искали тимлидов

как мы искали тимлидов



внутри компании

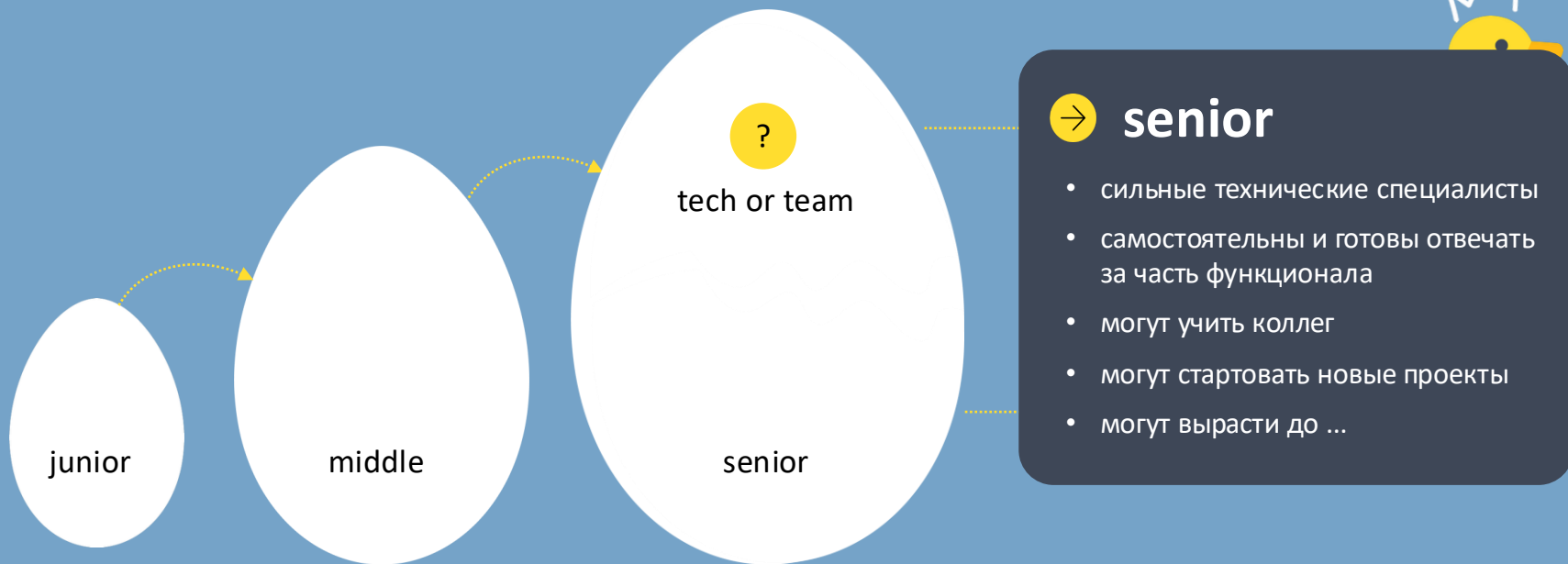
- ввели внутри нашего департамента процедуру performance review
- провели review с нашими разработчиками
- выделили потенциальных тим и техлидов

вне компании

- сформировали требование к кандидатам
- открыли отдельный набор на эту вакансию

performance review

профессиональный рост разработчика



изменения

шаг 2



шаг 3

оптимизируем усилия

проблема повторного использования общих наработок



архитектура фронта

стало

шаг 3

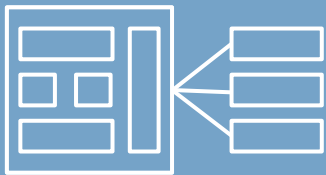
организация кода



монорепозиторий
для всех команд



выделяем в своих приложениях
общие части и выносим
в отдельные репозитории
и пакеты



monolithic/layered

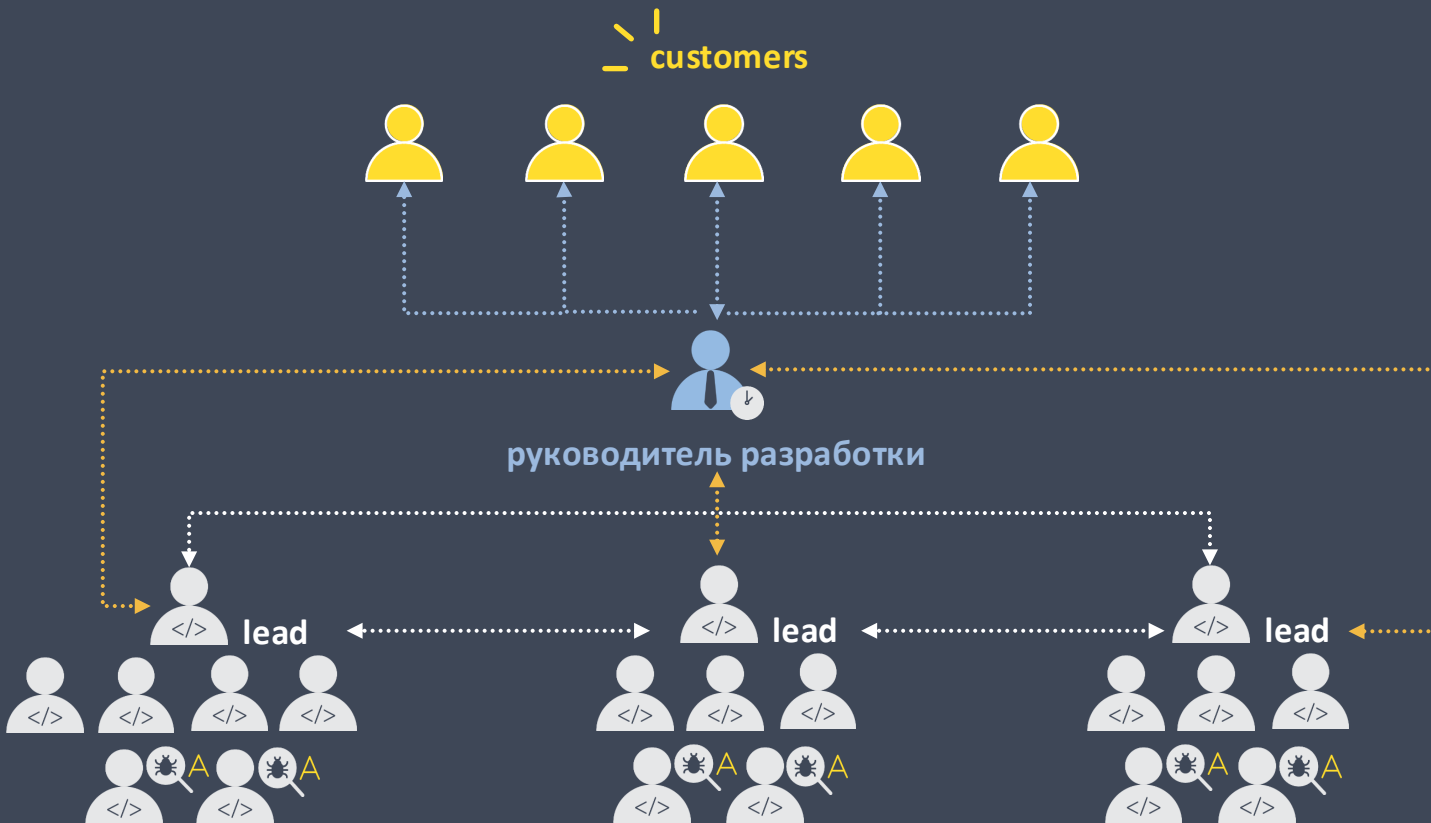
эксплуатация

- делим свое приложение на бизнес-части
- деплоим общий код на отдельную инфраструктуру для каждого бизнес-приложения
- обслуживаем им все запросы для всех страниц
- релиз своих изменений из своей релизной ветки

структура команды

было

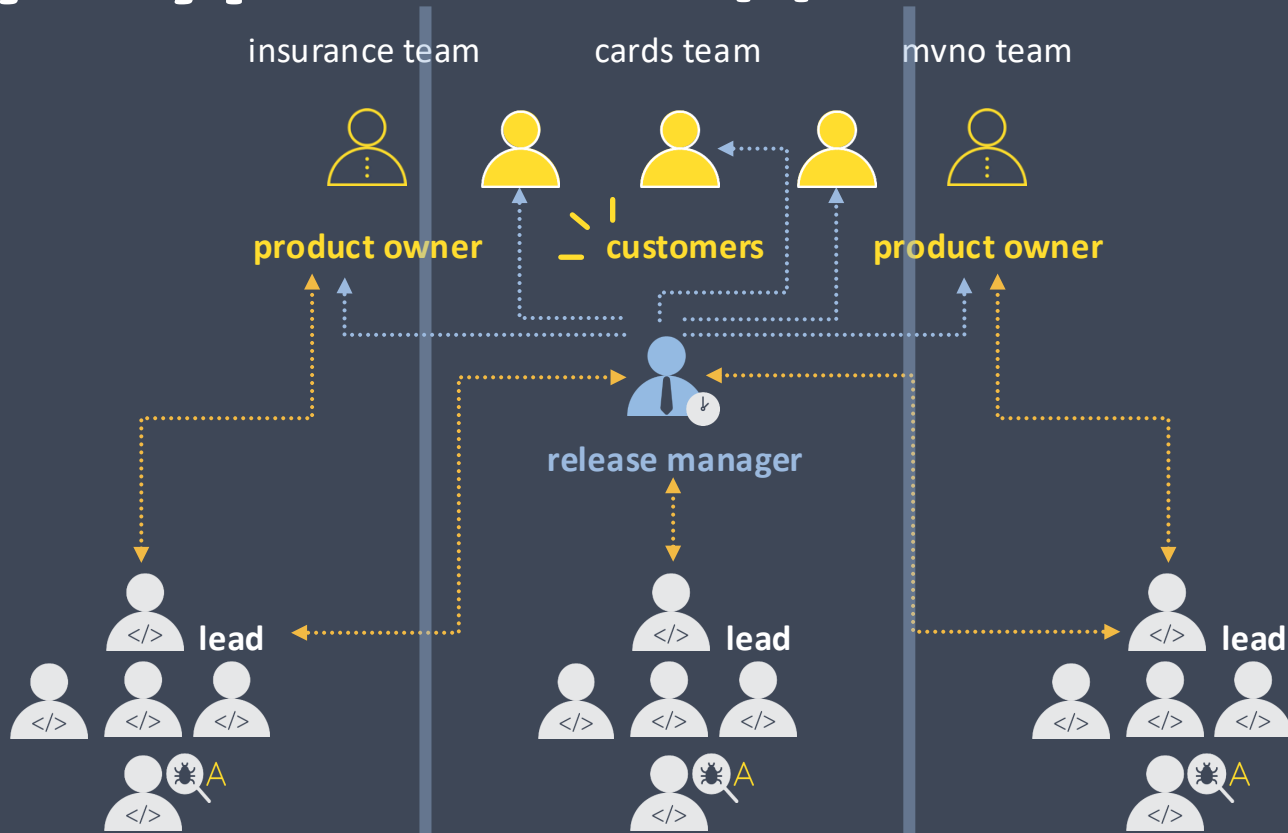
шаг 3



структура команды

стало

шаг 3



релиз-менеджеры и их роль



ТЮНИТЬ

процессы
в команде
разработки



ВЫСТУПАТЬ

входной точкой
для требований
в отсутствии
product'a



ОТВЕЧАТЬ

за содержание
релизов
и их сроки



РАЗГРУЖАТЬ

тимлидов
от организационной
работы
и координировать
работу команды
(daily, planning, retro)

обязанности тимлида

шаг 3



общение
с product'ом
/release
manager'ом
и другими
тимлидами



ведение
технического
роадмапа
своего продукта



общение
с core-командой
по доработкам
в общий
инструментарий



обучение
и наставничество
для разработчиков
внутри команды



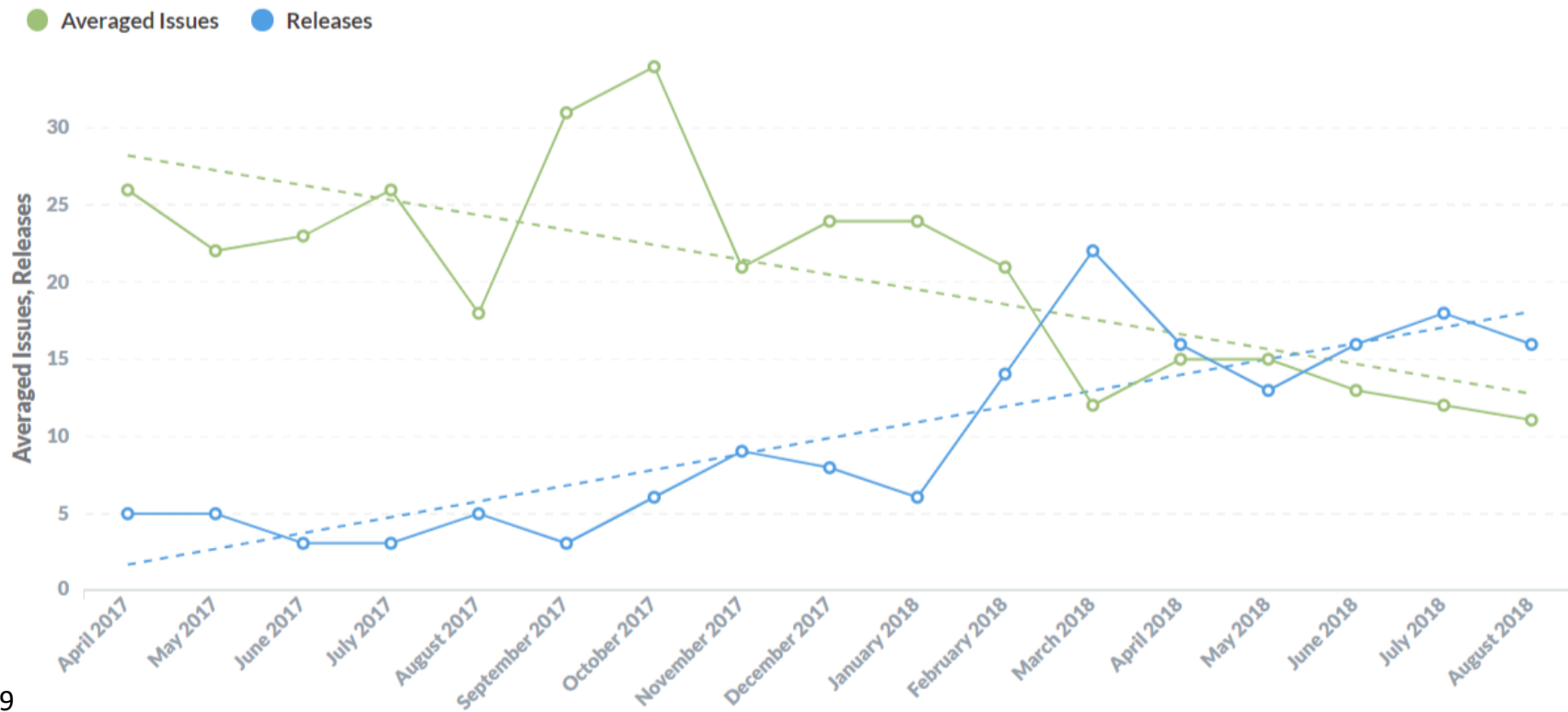
сбор
релизной ветки
и сопровождение
релиза

измеримость процессов

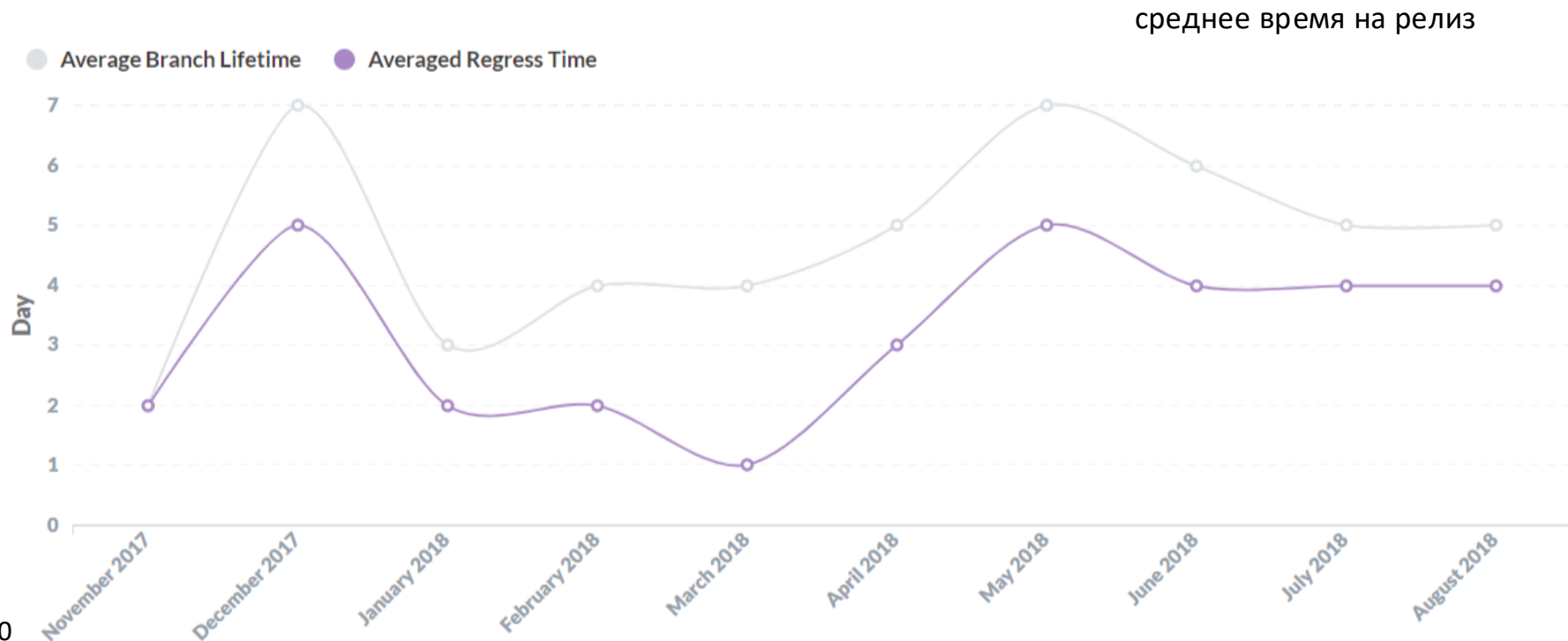
где нельзя терять



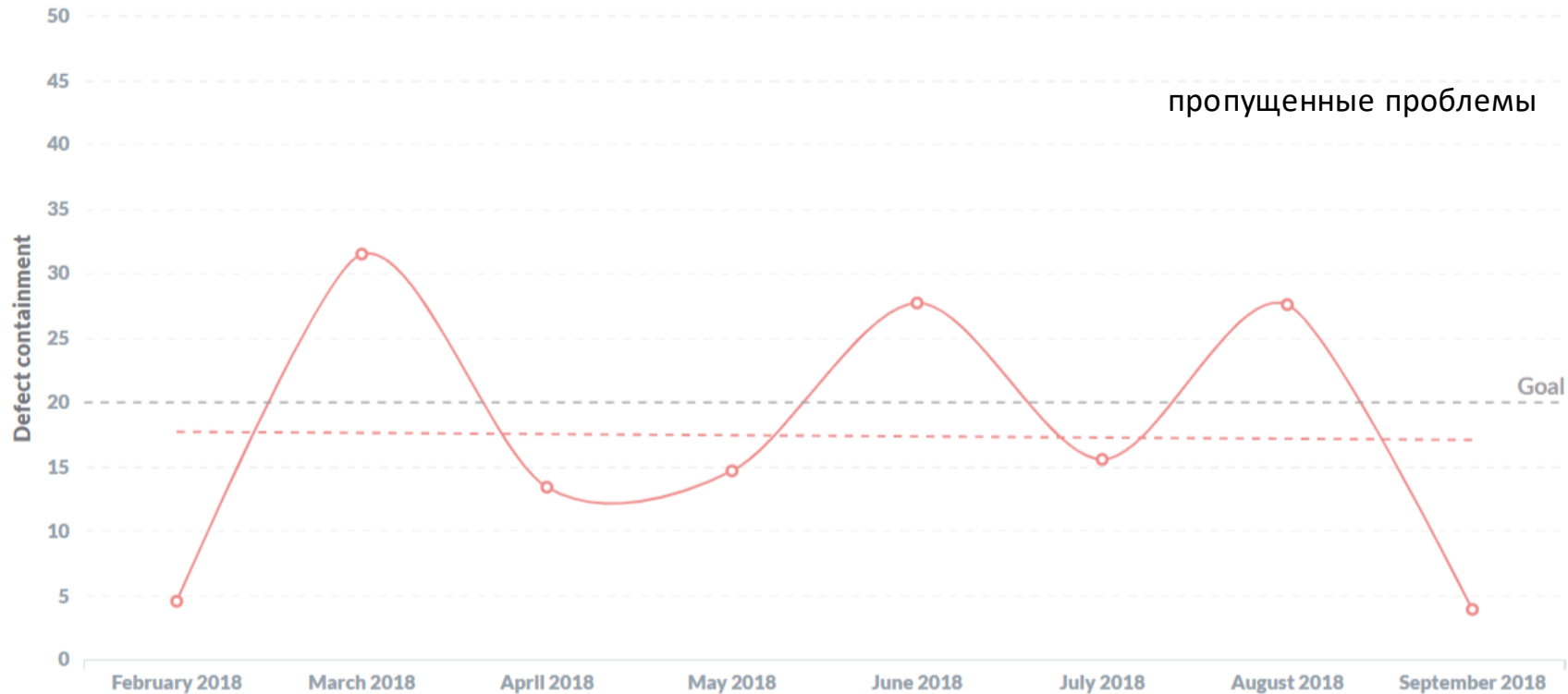
объем функционала



скорость выпуска



качество



изменения

шаг 3



шаг 4

делегуем ответственность

проблема распределения ответственности

**монорепозиторий
и общий код
приводит к размытию
ответственности
тимлида**

**внешние
эффекты
от чужих
изменений**

**сложно пробовать
новое (нигде ничего
не должно ломаться)**

архитектура фронта

стало

шаг 4

организация кода



отдельные репозитории
для приложений



общий инструментарий
подключаем через зависимости



инструментарий пишем в стиле
open source проектов

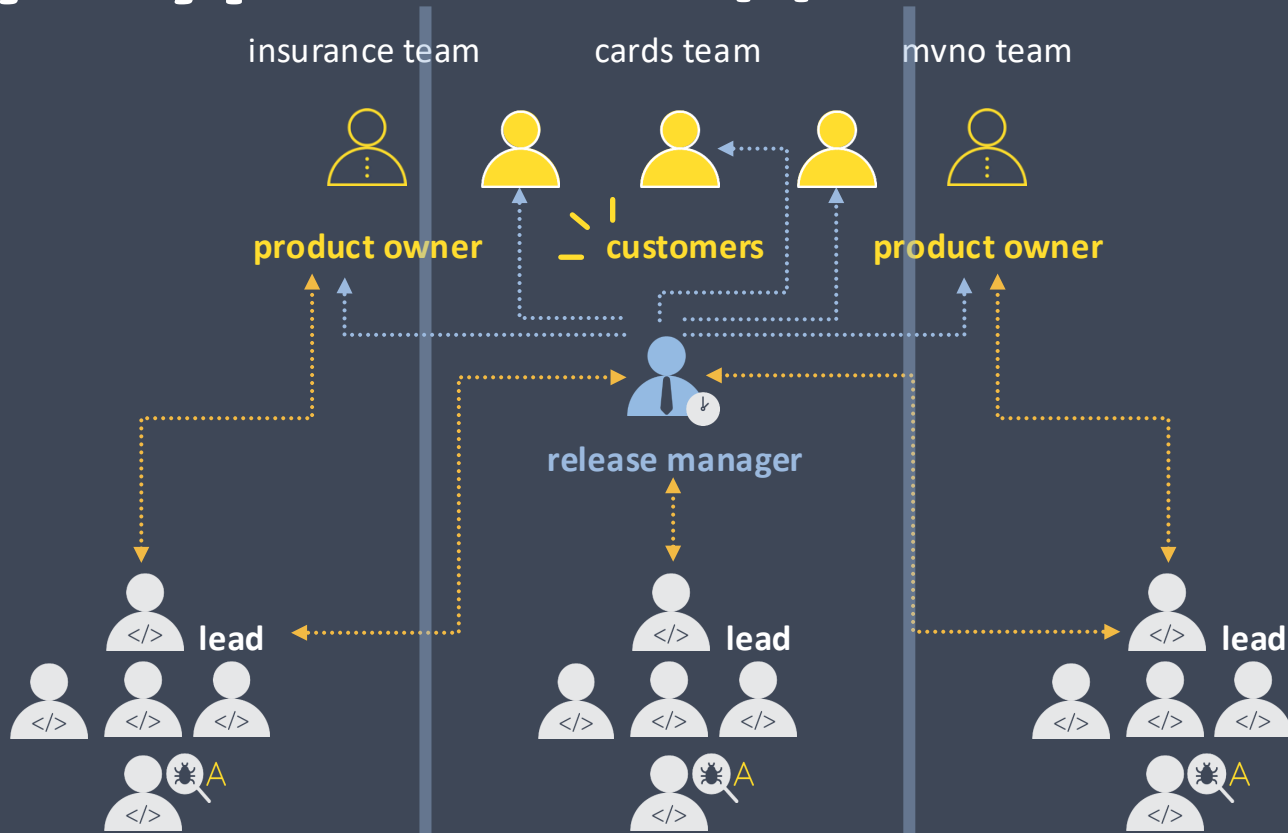
эксплуатация

- деплоим код на отдельную инфраструктуру для каждого бизнес-приложения
- обслуживаем своими приложениями свои url'ы
- релиз из своего repo из master'а

структура команды

было

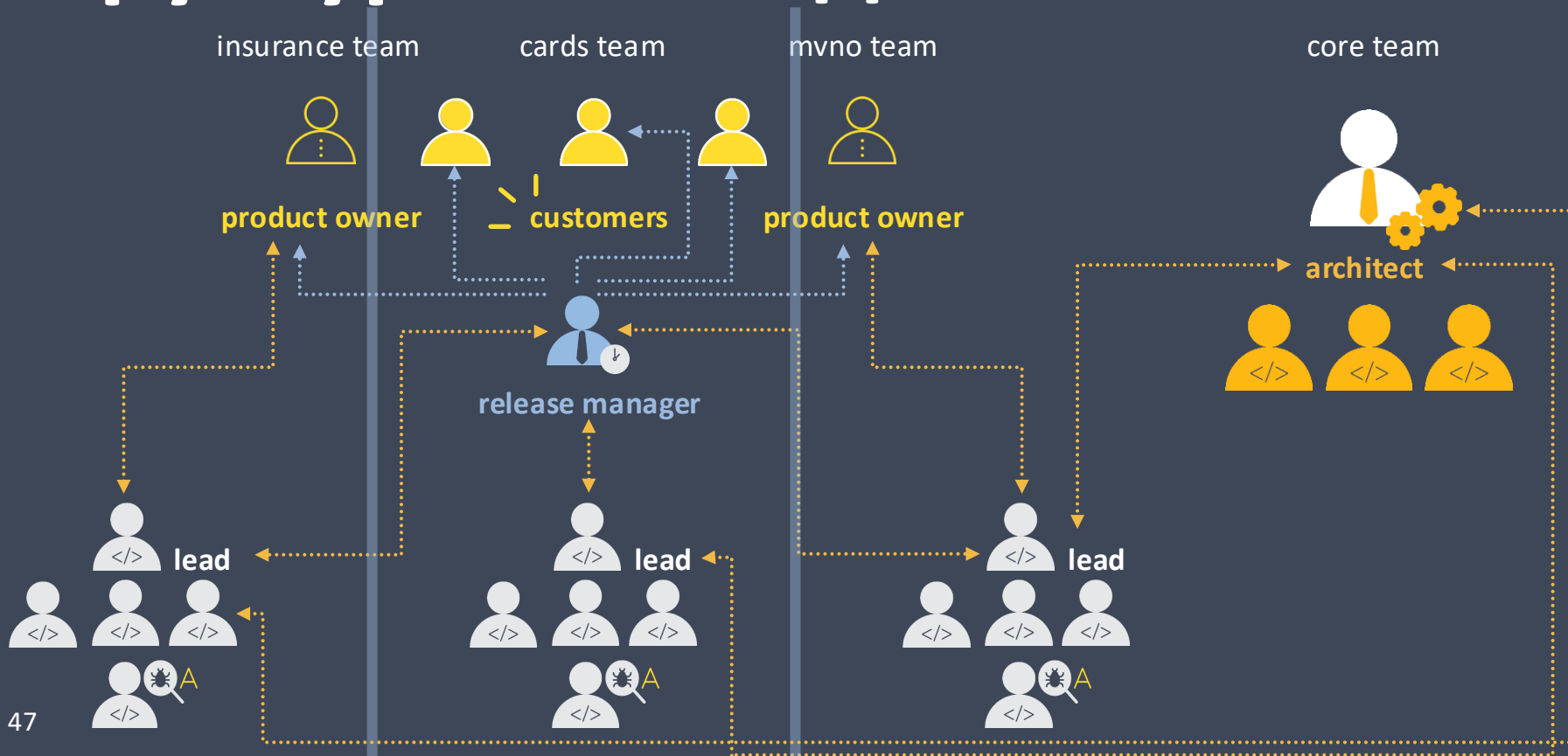
шаг 4



структура команды

стало

шаг 4



структура бизнес и технических направлений

бизнес-продукты

- cards
- loans
- sme
- insurance
- mvno

...

технические продукты

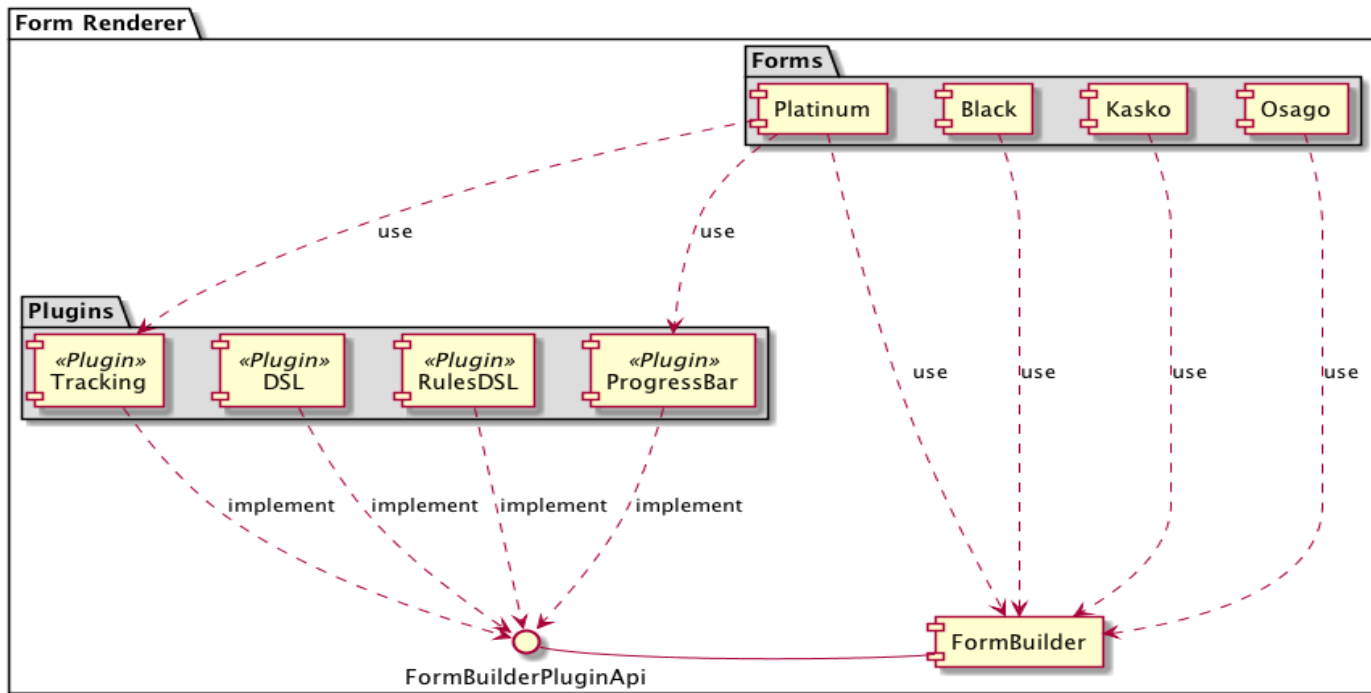
- движок страниц
- блоки для страниц
- движок форм
- формы продуктов
- трекинг

...

задачи команд

функционал	команды	core	insurance	sme	home
движок страниц		+	+	+	+
блоки страниц		+	+	+	+
движок форм		+	+	+	+
формы		+	+	+	+
трекинг		+	+	+	+

пример декомпозиции на инструменты



обязанности тимлида

шаг 4



общение

с product'ом/
release
manager'ом
и другими
тимлидами



ведение

технического
роадмапа
своего
приложения



общение

с core-командой
по доработкам
в общий
инструментарий



обучение

и наставничество
для разработчиков
внутри команды



сбор

релизной ветки
и сопровождение
релиза



полная

ответственность
за свое
отдельное
приложение

*технические продукты
как внутренний open source*

МОТИВАЦИЯ

хорошая
архитектура
с правильной
декомпозицией

повторное
использование
наработок между
командами

наличие
внутренней
конкуренции
(продукты
получаются лучше)

повышение уровня
ответственности
лидов команд
за свои приложения

как это должно работать



**общее
понимание
целей
бизнеса**



**согласованное
видение
архитектуры
системы**



**правильная
декомпозиция
компонентов
системы
на части**



**параллельная
работа
над техническими
продуктами
в рамках
развития бизнес-
продуктов**

какие могут быть проблемы

нет согласованного
видения архитектуры
системы

работа
над техническими
продуктами
по остаточному
принципу

нет трансляции
бизнес-целей
в технические

«кривая»
декомпозиция
компонентов
системы

работа над
техническими
продуктами
в отрыве
от бизнес команд

почему это работает у нас



есть общее
понимание
целей
бизнеса —
шаг 0 – шаг 1



есть
согласованное
видение
архитектуры
системы —
шаг 1 – шаг 2



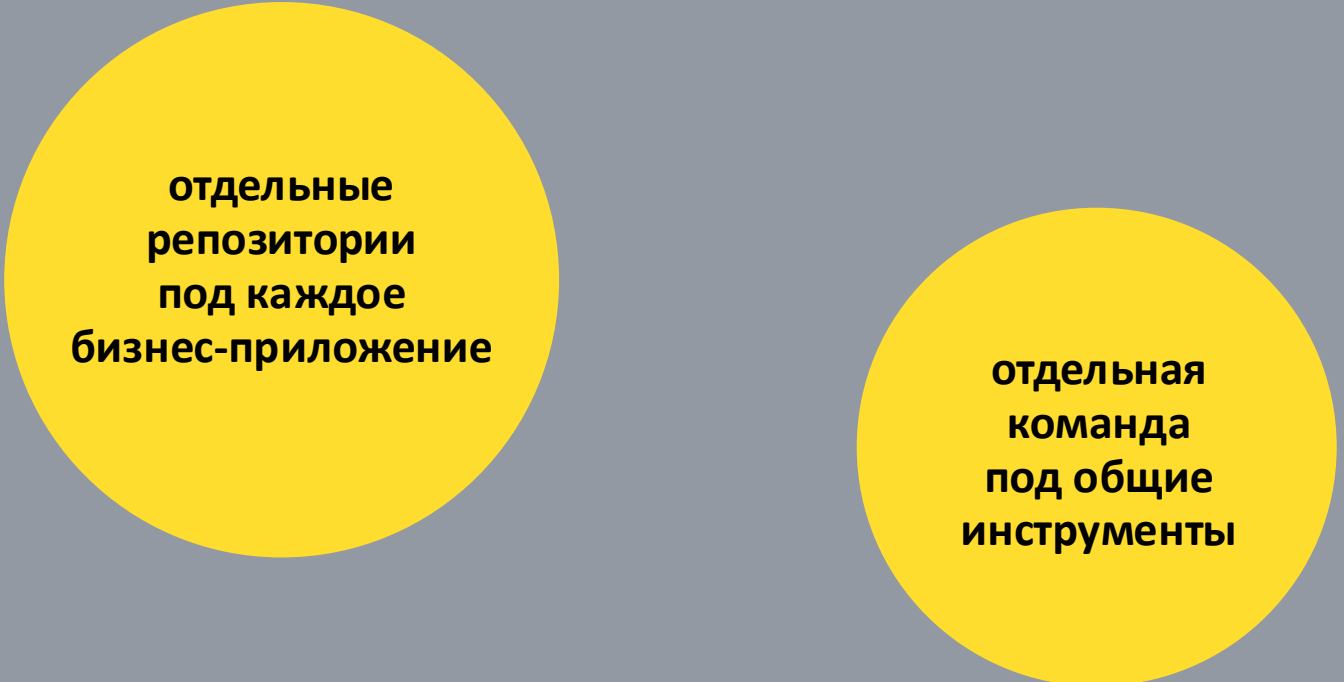
технические
продукты – это
тоже продукты —
шаг 3 – шаг 4

- есть product owner'ы
технических продуктов
- есть roadmap
развития продуктов



есть общий
архитектор
и product owner
для всех
инструментов —
шаг 4

изменения



**отдельные
репозитории
под каждое
бизнес-приложение**

**отдельная
команда
под общие
инструменты**

~~шаг 5~~

которого еще нет

факты



почти ушли от монолита,
но уже пришли к отдельным
приложениям с вынесенными
зависимостями



ушли от процессов «как есть»
к измеримым и постоянно
улучшаемым процессам

как менялась роль лида

детский сад

- монолит
- общие релизы



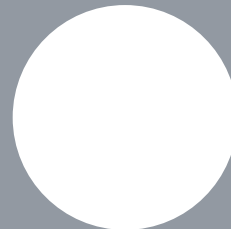
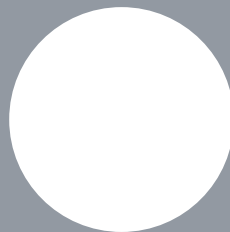
ШКОЛА

- монолит
- отдельные релизы



университет

- монолит для каркаса приложения
- общие либы в отдельных репах
- подключаем их через зависимости



взрослая жизнь

- отдельные приложения
в отдельных репах
- инструменты выбираешь
сам ... аргументированно
- полностью отвечаешь
за свое приложение



как не сойти с ума тимлиду

четко понимать
цели бизнеса

отчетливо осознавать
этап жизненного цикла
развития команды
и обязанности тимлида
на этом этапе

честно для себя
решить надо ли
тебе это ...
или стоит
продолжать
писать код

вопросы

mailto: alexander.polomodov@gmail.com

ask: www.facebook.com/alexander.polomodov

шаг 1

первые
исправления

- отдельные релизы
- общение через project'a

шаг 2

масштабируемся

- 12 человек
- деление на отдельные команды по продуктам

шаг 3

оптимизируем
работу

- выделение общих инструментов
- Появление релиз-менеджеров

шаг 4

делегируем
ответственность

- 40 человек, 10 тимлидов
- отдельная core-команда
- измеримые процессы

~~шаг 5~~