

**Перестаньте
отмазываться
от ответственности
за свой код**

или как мы живем 15 лет без роли QA

Mindbox

Облачный B2B SaaS

1200

клиентов,
геораспределенный
продукт, доступность 24/7

400

сотрудников,
200 в R&D

ТОП-10

клиентов
Yandex Cloud

15

лет пилим
один продукт

mindbox

Mindbox

Высокая нагрузка
и требования доступности:

- интеграция с кассами,
- процессинг баллов и скидочек
- платформа обрабатывает сотни миллионов коммуникаций с потребителем в час
- **1,5 млрд** потребителей в БД

Михаил Кузнецов

Technical fellow

5 лет в Mindbox



Два вида приемки

QA

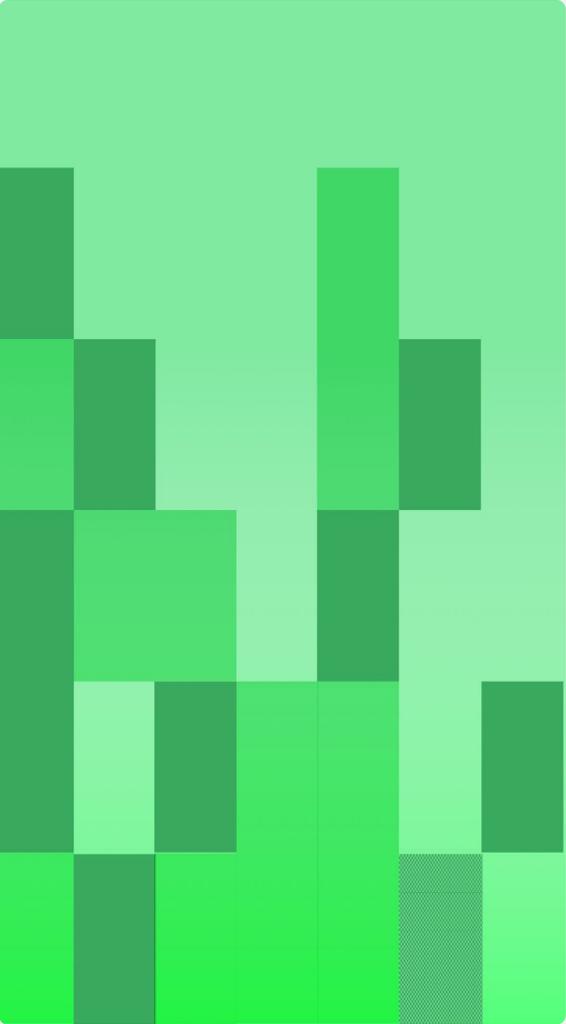
сделали
качественно

Продукт/БА

сделали то,
что нужно

У нас нет роли QA*

* на самом деле есть, только в одной команде — расскажу позже про этот кейс



Принципы

1. Максимально короткий feedback loop:
быстрее разработка → быстрее приемка →
быстрее фидбек от пользователей
2. CI = master-based development
3. CD = автоматически выкладываем много
раз в день на прод
4. Разработчик отвечает за свой код

Ручные тестировщики и CI/CD

Медленная приёмка

Неочевидный объем тестирования

С микросервисами - лучше, но всё равно - отслеживать зависимости, иногда может быть прям много

Ограничения

- Развиваемся на свои
- Жмём на эффективность
- Короче цепочки
- Удобнее менеджить
и балансировать ресурсы

Внутренние правила и принципы

01

**Быстрой деплой
и роллбек**

миграции — только с обратной
совместимостью

не выкладывать в пятницу
вечером :)

02

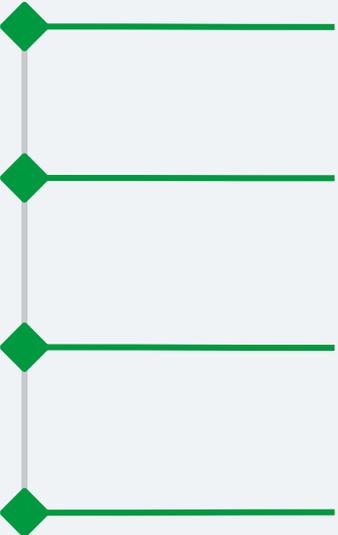
**У разработчиков есть
доступ к метрикам
и мониторингу прода**

03

**(Не)очевидный пункт –
возможность запускаться
и подебажить локально**

2009 - ...

Как эволюционировала система контроля качества



Unit-тесты. На старте – тысячи.
Сейчас – десятки тысяч

Интеграционные тесты

Автобаги и бесконечная починка
багов в подарок

Apdex

2013 - ...

Усложняемся и растем

- ◆ End-to-end тесты
- ◆ Puppeteer-тесты
- ◆ Тесты на контракты
- ◆ Анализаторы, включая самописные
- ◆ Хартбиты

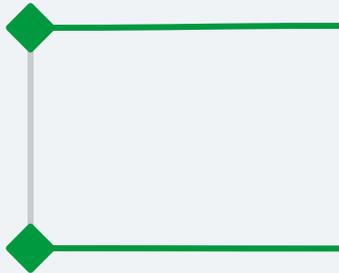
2016 - ...

Усложняемся и растем 2

- ◆ Канарейка
- ◆ Метрики
- ◆ Алерты
- ◆ Фиче-тоглы
- ◆ Sync → Async
- ◆ Трейдофф надежности в пользу скорости, но не везде (см.SLA)

2017 - ...

Появляется SLA



SLA – это часть продукта с точки зрения ПО
выяснять и обновлять по потребностям клиента

И с точки зрения кода

встроено и поставляется вместе с основной
функциональностью

автоматический сбор статистики, алерты

2023 - ...

Усложняемся и растем 3

◆ — Инфраструктурные проблемы → делим
на контуры

бласт радиус, скейлинг нагрузки

Метрики



Не фиксируем
coverage



Бюджет
недоступности



% времени
блокировки
пайплайна

Минусы нашего подхода

Десятки тысяч тестов. Это тоже код, который надо поддерживать, а это замедляет разработку

Ощутимое время на выполнение чеков. Тоже замедляет разработку.

Где зафейлились

- ◆ Выносили/переписывали устаревший компонент логики из монолита в микросервис
фич не прибавилось, но потрогали очень много кода
- ◆ Продукт **очень** гибкий:
шаблонизатор, фильтры, параметры
- ◆ Развалился % корнер-кейсов на % клиентов

Ретроспективно считаем, что QA бы все равно никак не помог

Возвращаясь к * – все-таки QA есть

трайбы

В Mindbox самоуправляемые трайбы (большие команды) – захотели и наняли Mobile QA. Это был трайб Web&Mobile и наняли его под mobile SDK.

трейдофф

больше надежности, меньше скорости

специфика

роллбек невозможен
деплой – тоже непредсказуем
канарейки нет
метрики не помогут
цена ошибки – очень высока, а контроля над ней – минимум

Различия в майндсете

цель разработчика

чтобы
работало

цель тестировщика

сломать, найти баг

при этом он работает с БА,
знает JTBD, а разработчик?

Отчасти это правда, и можно это усилить организационными мерами

А можно предлагать людям брать ответственность за результат

«А почему, собственно, я должен...?»

Не должен, конечно, мамин пирожок

«Писать тексты = Мыть полы» и такое мнение есть



Но: Что с TDD?

А с пинг-понгом задачи на приёмке?

А с быстрыми правками в процессе?

Как в Mindbox

Мы много хотим:

- проактивность
- инициативность
- ответственность



О работе инженеров
в Mindbox

Как в Mindbox

Мы много даем:

- бирюзовая организация,
- право вето у каждого сотрудника,
- открытые финансы,
- безлимитные отпуска,
- большая автономия трайбов,
- разработчики знают и работают на выручку своего продукта. Челленджат ПО, помогают ему, а могут и уволить.



О работе инженеров
в Mindbox

«И где таких взять?»

Стажировка

36% от состава в R&D выращены внутри из лучших кандидатов (студентов тех. вузов)

Типичная воронка:

- 150 откликов с тестовым
- 3-4 стажёра
- 2-3 джуна

«И где таких взять?»

Найм с рынка

Обычно для сеньоров. Нанимаем небыстро, въедливо и тщательно

Смотрим на: **софтскиллы;**

МОТИВАЦИЮ ОТ ПОЛЬЗЫ, а не от технологий

Как прокачиваем в QA экспертизу

Культура, коллеги, best practices

Пару слов про экономику

Хороший автотестер стоит примерно как разработчик

При этом:

- сам пофиксить баг «не может»
- фичу сделать «не может»

Плохой – напишет плохие тесты. А тесты – это код, со всеми вытекающими:

- дорогие,
- неподдерживаемые
- тормозящие тесты

Ручные тестировщики - да, дешевле, но обсудили выше

Применимость подхода



Зачем?

Если цена ошибки велика → релизный цикл медленнее

Если не SaaS → релизный цикл медленнее

В какой-то момент может оказаться, что **ценность CI/CD отсутствует**

Как внедрять у себя – общие соображения

Зависит от вашей зоны ответственности и майндсета коллег
Внутри команды практически всегда можно (надеюсь)



Начните с себя,
если никто
не против



Если будет польза –
люди будут
копировать



Масштабируйте
на команду, отдел,
всю разработку

Так работает, только если окружающие – про пользу

Как продать на уровне бизнеса/компании

Agile/fail fast

Ограниченный
эксперимент

Метрики
здоровья

Бюджет
недоступности

Скорость
разработки

Если
получается :)

mindbox

Теперь слово вам



Михаил Кузнецов
technical fellow Mindbox



@mishka64