



# Автоматизация тестирования аналитики мобильного приложения

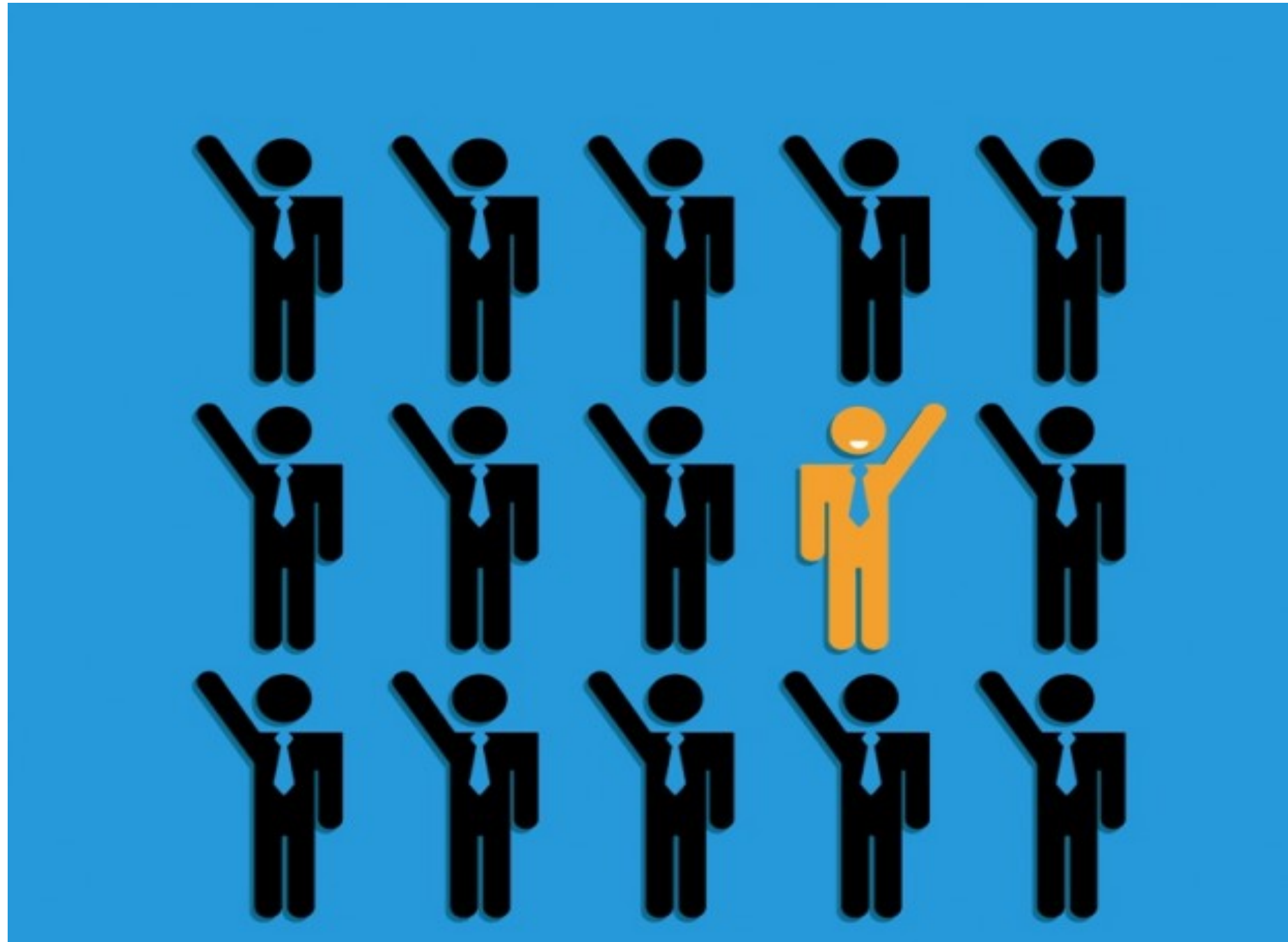
Павел Щедухин

Head of Mobile QA Ozon.ru

# Команда QA



**А кто-нибудь тестирует аналитику?**



В чем сложности

Как раньше тестировали

Как улучшить

Как можно автоматизировать

Вывод

# Долго доходит

- Собирается в батчи
- Отправляется
- Обработывается
- Становится доступна

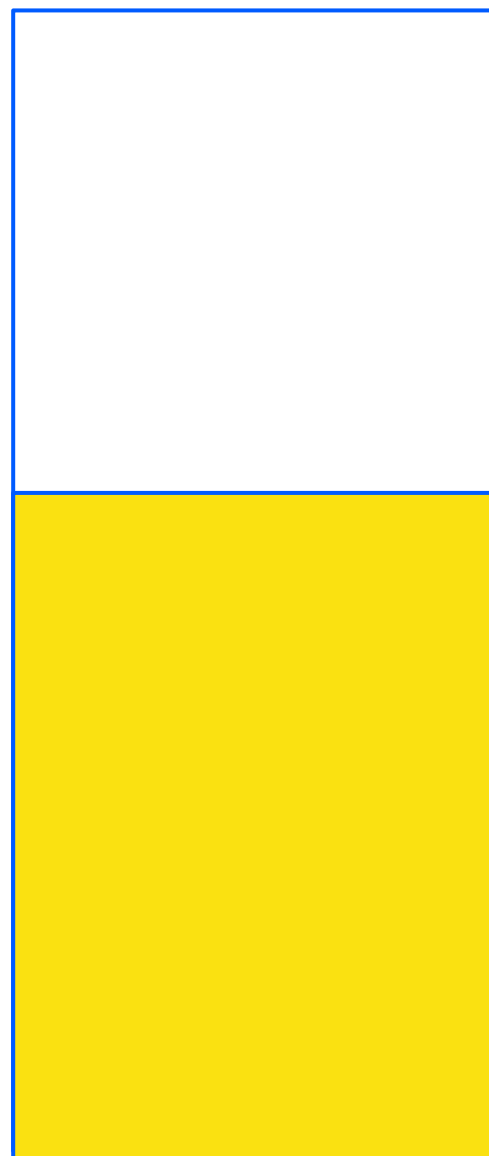


# Батчинг

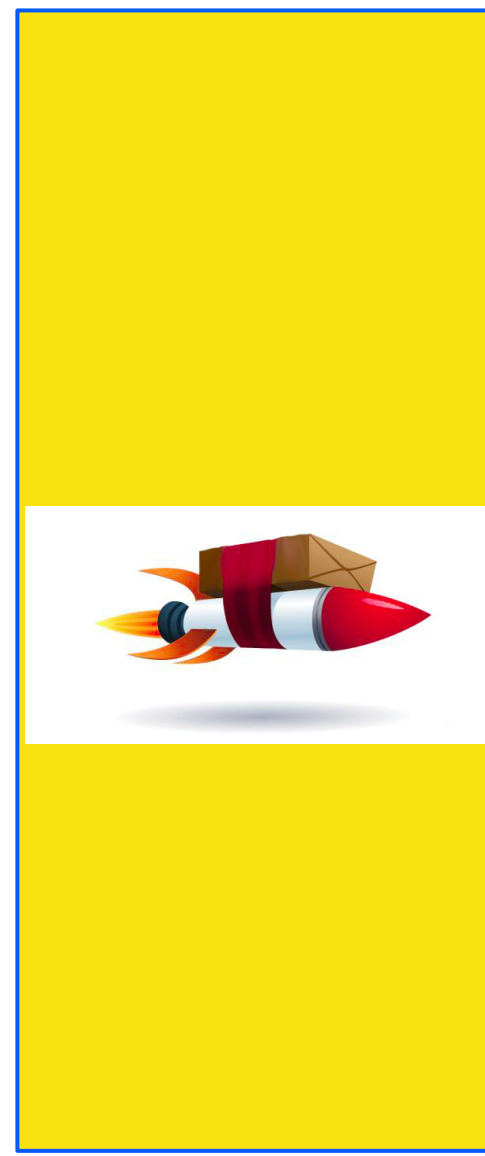
Нет данных



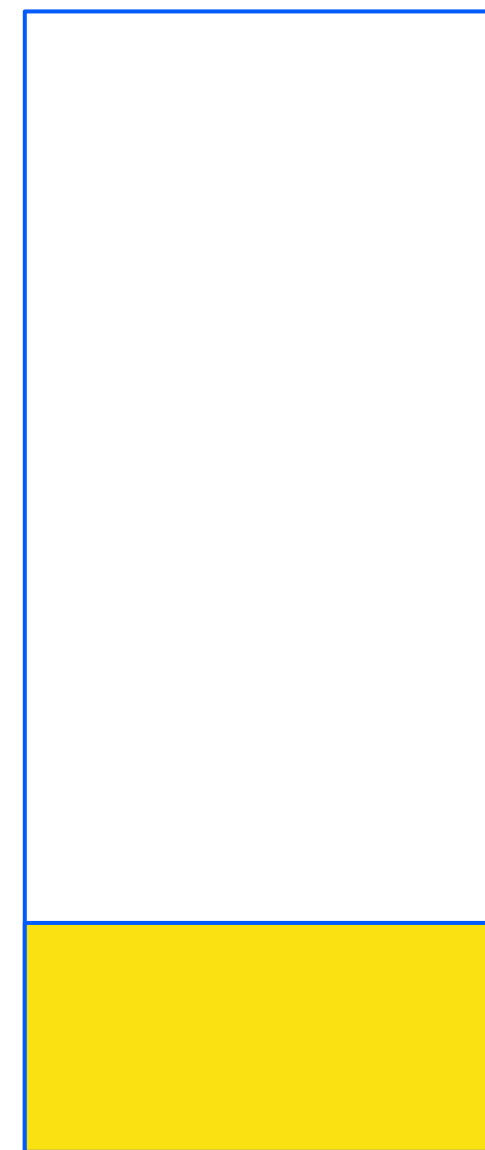
Собирается



Отправка



Собирается



# Сложно читаемые значения

cid af8eabed-2e8f-4a8b-b347-205c4ccf3020

\_s 461

pr1ca 0

tid UA-37420525-4

pa detail

\_crc 0

cd45

ck

cd112 1

cd1

\_u .7nBCL

**Трудно получить доступ**





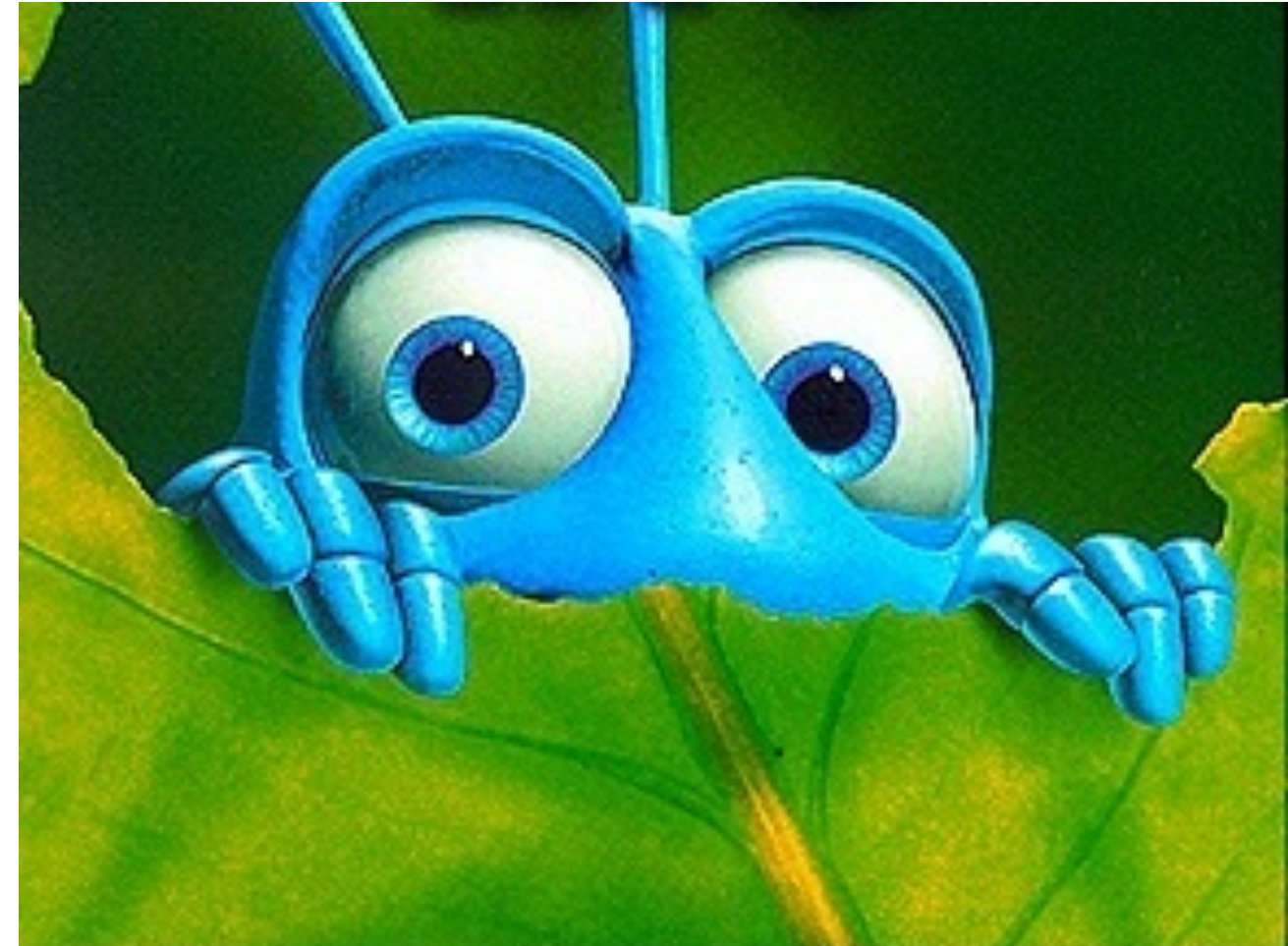
# Много разных систем

- Google Analytics
- AppsFlyer
- Yandex AppMetrica
- Facebook Analytics
- Что-то своё
- и многие другие...



## Много неочевидных кейсов

- Дублирование событий
- Поведение зависит от... всего
- Обработка данных с бекенда



# Демотивирует

- Отнимает много времени
- Сложно в тестировании
- Легко что-то пропустить



## При добавлении/изменении

- Не проверяли при изменении фичи
- Не проверяли на регрессе



В чем сложности

Как раньше тестировали

Как улучшить

Как можно автоматизировать

Вывод

## По сигналу от аналитиков

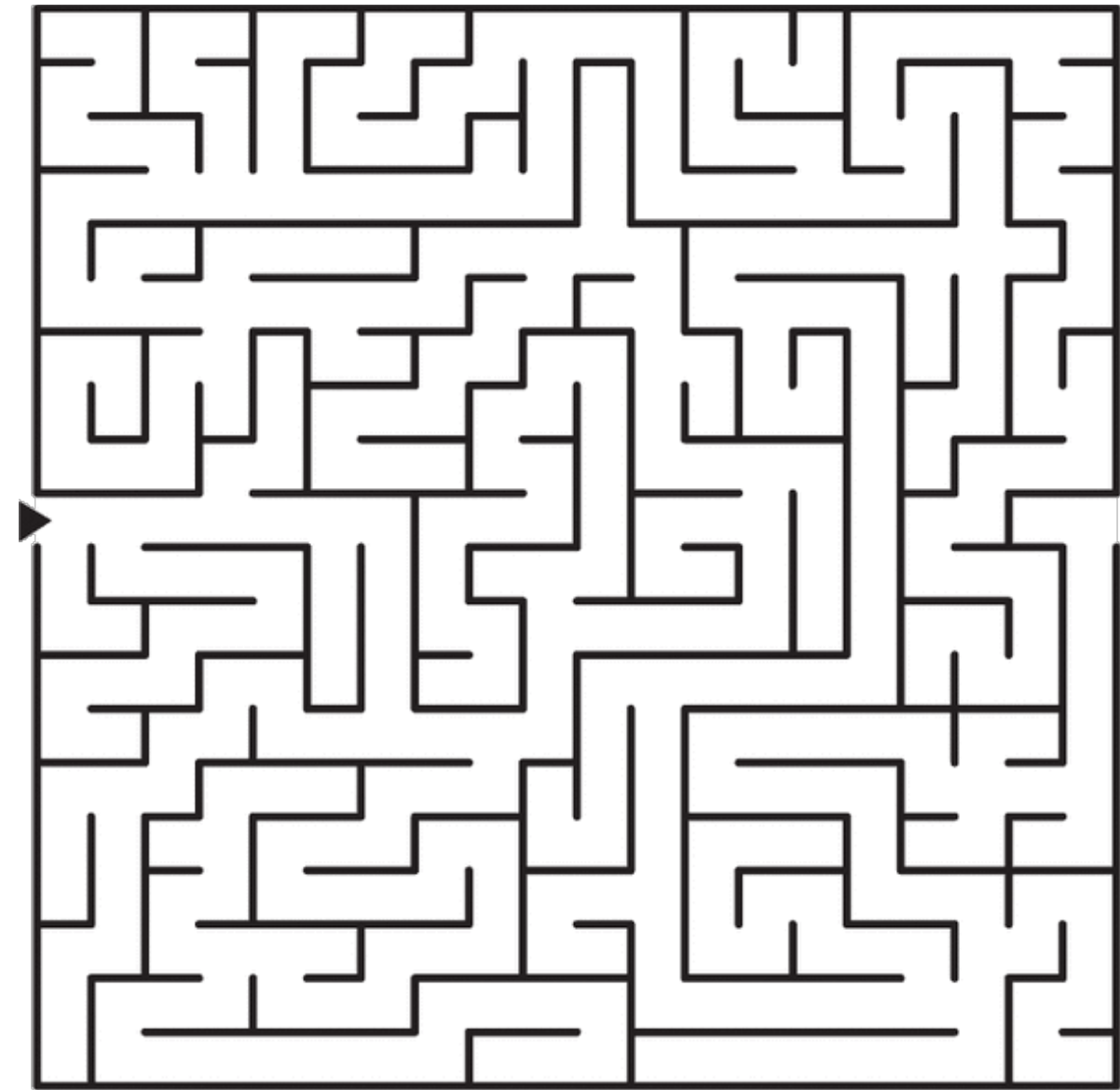
Непонятно где баг:

- Функционал
- Аналитика
- Мониторинг аналитиков



## К чему это приводило

- Много времени на поиск ошибки
- Может пройти несколько релизов с того времени как баг привнесен



## Еженедельные релизы на обе платформы

- Выходит что-то, что надо трекать
- Появляется то, что может сломать аналитику
- Нет возможности уделять много времени на...



## К чему это приводило

- Все данные кривые
- Долгое время не знаем о проблеме
- Потом надо исключать данные о проблемных версиях



В чем сложности

Как раньше тестировали

Как улучшить

Как можно автоматизировать

Вывод

# Проверять на регрессе

- Уходит очень много времени
- Команда не любит регресс



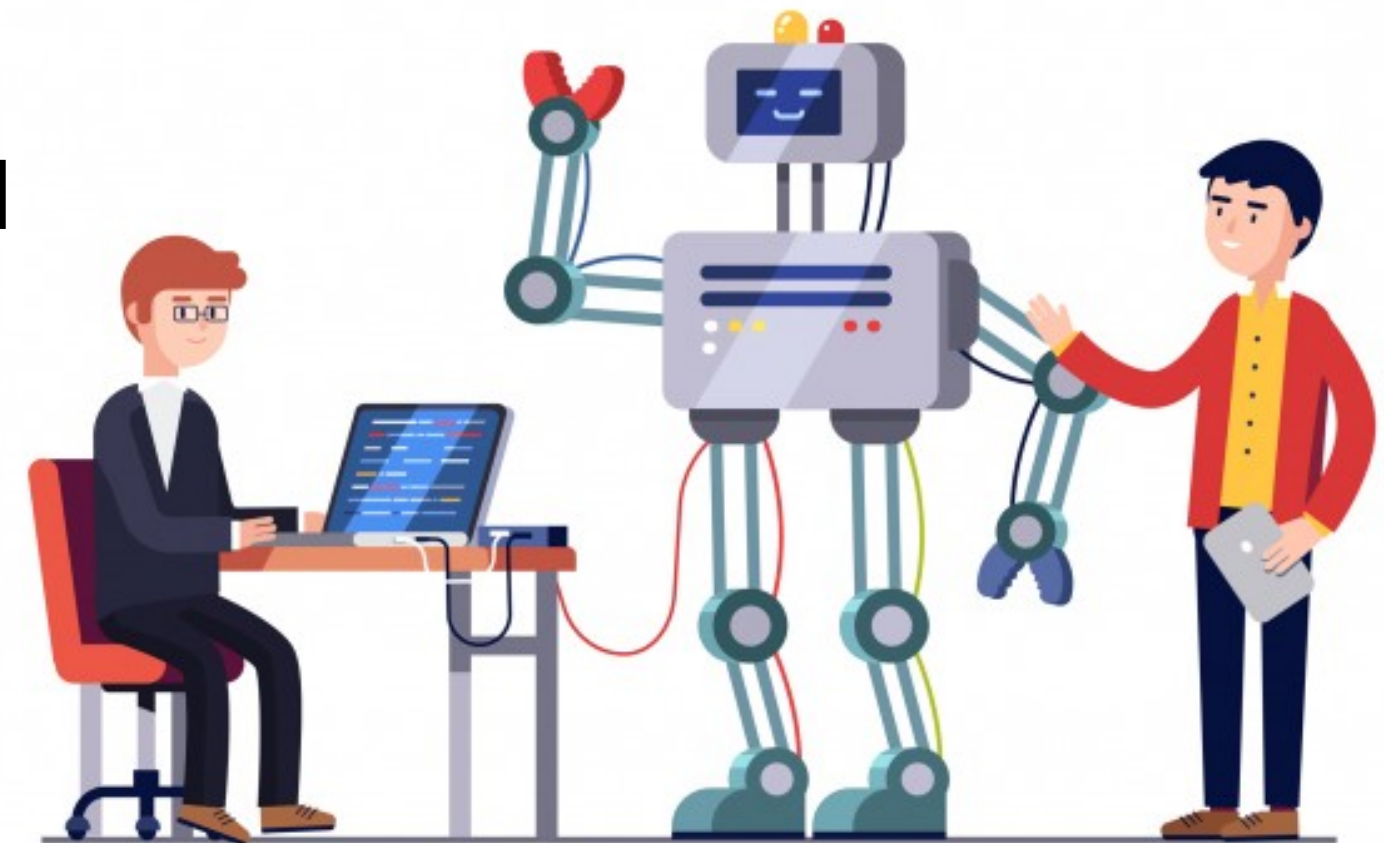
# Подключить к тестированию аналитиков

- Слабая вовлеченность
- Быстро отваливаются



# Автоматизировать тестирование аналитики

- Освобождает от ручного тестирования
- Нужно меньше проверок, чем обычным UI тестам
- Реже ломаются, чем UI



В чем сложности

Как раньше тестировали

Как улучшить

Как можно автоматизировать

Вывод

# Как автоматизировать

1. По логам приложения

2. Брать данные с бекенда

3. Перехватывать отправленные данные



# Framework

Android



+



iOS



Логи

# Плюсы

+ Легкая реализация по считыванию логов

+ Удобно встраивается в текущие

тесты

+ Легкая реализация по считыванию логов

```
2019-01-06 00:03:32.344 939-991/? I/WifiService: reportActivityInfo uid=1000
2019-01-06 00:03:32.344 939-991/? I/WifiService: getSupportedFeatures uid=1000
2019-01-06 00:03:32.348 28452-23884/? W/TileUtils: Found com.android.settings.backup.BackupSettingsActivity for intent Intent { a
2019-01-06 00:03:32.362 28452-23884/? D/Settings: No enabled state changed, skipping updateCategory call
2019-01-06 00:03:32.363 28452-28452/? D/DashboardSummary: Listening for condition changes
2019-01-06 00:03:32.364 28452-28452/? D/DashboardSummary: onConditionsChanged
2019-01-06 00:03:32.364 28452-28452/? D/DashboardAdapter: adapter setConditions called
2019-01-06 00:03:32.364 28452-28452/? D/DashboardSummary: conditions refreshed
2019-01-06 00:03:32.364 28452-28452/? D/APM_Condition: APM condition refreshed
2019-01-06 00:03:32.364 28452-28452/? D/APM_Condition: setActive was called with false
2019-01-06 00:03:32.364 939-2071/? I/WifiService: getWifiApEnabledState uid=1000
2019-01-06 00:03:32.370 28452-23885/? D/SuggestionParser: Day 86 for com.android.settings.suggested.category.FIRST_IMPRESSION
2019-01-06 00:03:32.397 28452-23885/? D/NewDeviceIntroSugg: is suggestion expired: true
2019-01-06 00:03:32.432 939-991/? E/BatteryExternalStatsWorker: no controller energy info supplied for bluetooth
2019-01-06 00:03:32.460 668-668/? E/QCOM PowerHAL: extract_stats: failed to open: /d/wlan_wcnss/power_stats Error = No such file
2019-01-06 00:03:32.479 28452-23641/? D/BatteryInfo: time for getStats: 135ms
2019-01-06 00:03:32.480 28452-23886/? D/BatteryInfo: time for regular BatteryInfo: 1ms
2019-01-06 00:03:32.481 28452-23886/? D/BatteryInfo: time for getBatteryInfo: 1ms
2019-01-06 00:03:32.482 28452-28452/? D/BatteryInfo: time for callback: 0ms
2019-01-06 00:03:32.669 2121-2121/? I/CastMirroringProvider: onDiscoveryModeChanged, mode=0
2019-01-06 00:03:32.670 939-939/? V/MediaRouter: Removing route: RouteInfo{ name=Living Room Speaker, description=Google Home Mir
2019-01-06 00:03:32.671 939-2211/? W/ActivityManager: Permission Denial: Accessing service ComponentInfo{com.google.android.music
2019-01-06 00:03:32.671 1485-1485/? V/MediaRouter: Removing route: RouteInfo{ name=Living Room Speaker, description=Google Home M
2019-01-06 00:03:32.675 2121-2121/? W/DiscoveryManager: BroadcastReceiver is already registered
2019-01-06 00:03:32.676 28452-28452/? V/MediaRouter: Removing route: RouteInfo{ name=Living Room Speaker, description=Google Home
2019-01-06 00:03:32.676 2121-23882/? I/CastMirroringProvider: onDiscoveryModeChanged removing all remote displays
2019-01-06 00:03:32.677 2121-23882/? I/DiscoveryManager: Filter criteria(%urn:x-cast:com.google.cast.media) scannerFlags(2)
2019-01-06 00:03:32.679 939-2071/? I/WifiService: getWifiEnabledState uid=10011
```

# Минусы

- Сборка отличная от пользовательской
- Не проверяем отправку
- Разработчикам надо добавлять логи



# Как это трекать

## Android

```
driver.get_log('logcat')
```

## iOS

```
{"showIOSLog": true}
```



**android**  
**Logcat**

# Логи

- Мы тестируем на продакшен сборках
- Не раз ловили баги, когда логи есть, а отправки нет



# Перехват отправленных данных

# Плюсы

- + Проверяем именно то, что отправлено с девайса
- + Средняя скорость получения данных



# Минусы

- Медленно
- Потребляет много ресурсов
- Для параллельного запуска нужно несколько экземпляров сниффера





# Запустить эмулятор с прокси

## Android

```
emulator -avd test-proxy -http-proxy http://ip:port
```

## iOS

Настроить прокси в MacOS

Прокси-сервер Вручную >

Имя хоста proxy.example.com

Порт 8080

Исключения example.com,mycomp.test.c...

Настройки IP DHCP >

Изменить сеть

Удалить эту сеть

## Установить SSL сертификат

- Скачать сертификат
- Получить хеш
- Установить сертификат



## Android

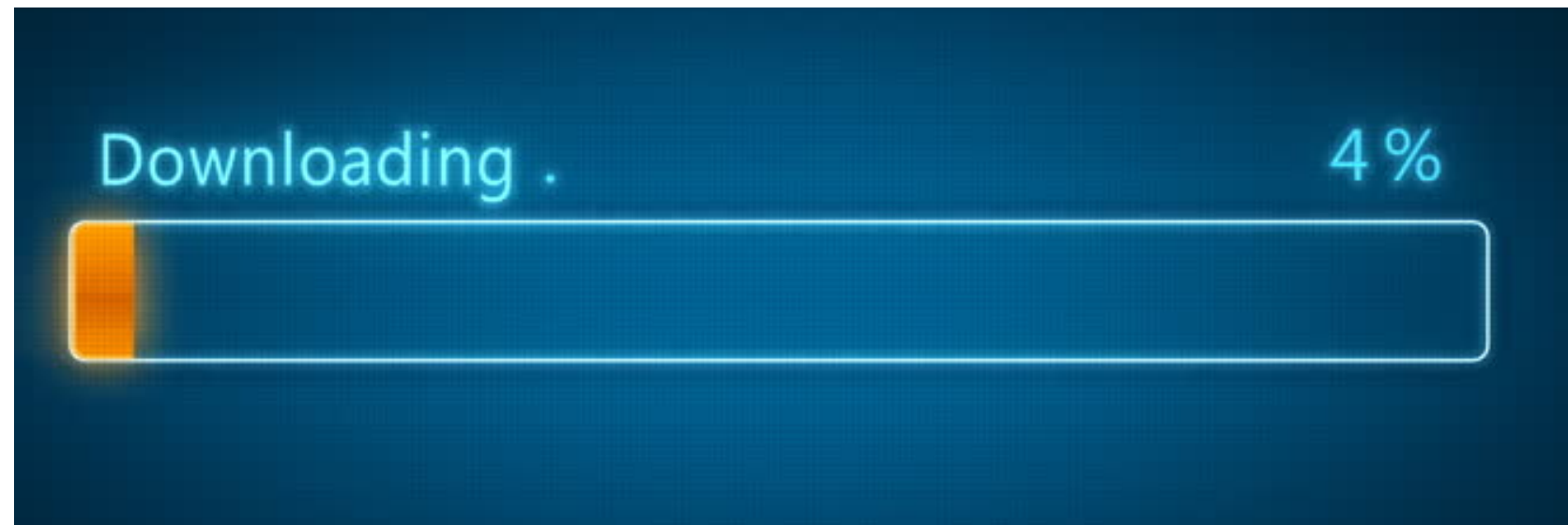
```
adb push $c /system/etc/security/cacerts/$hash.0
```

## iOS

```
driver.executeScript("mobile:installCertificate",args)
```

## Скачать сессию

— Обращаемся к Charles по:  
урлу + `session/export-json`



# Сниффер

1. Проходит тест
2. Скачивается сессия
3. Парсятся события
4. Получаем результат



# Web Interface

- Старт/Стоп записи сессии
- Очистка текущей сессии
- Скачивание сессии



# Web Interface

## Recording

Status: Recording

- [Start](#)
- [Stop](#)
- [Back](#)

## Tools

- [Breakpoints](#)
- [No Caching](#)
- [Block Cookies](#)
- [Map Remote](#)
- [Map Local \(Enabled\)](#)
- [Rewrite](#)
- [Black List](#)
- [White List](#)
- [DNS Spoofing](#)
- [Auto Save](#)
- [Client Process](#)
- [Back](#)

- [Throttling](#)
- [Recording](#)
- [Tools](#)
- [Session](#)
- [Quit](#)

## Session

- [Clear Session](#)
- [Export Session as XML](#)
- [Export Session as JSON](#)
- [Export Session as Trace](#)
- [Export Session as CSV](#)
- [Export Session as HAR](#)
- [Download Session](#)
- [Back](#)

# Тест проходит долго

- Скачивание сессии затягивается
- Сниффер добавляет тормозов

# Кончается память

- В контейнере
- В снифффере





# Сложно параллелить

- Либо все стучатся в один сниффер
- Либо надо поднимать по снифферу на каждый поток



# Баги Чарльза

- Не чистит сессию
- Зависает на фильтрации



# И еще много всего

- Установка сертификата
- Получение ip-адреса
- Доступы между сетями



# А главное



Данные из базы

# Плюсы

+ Максимально приближено к потребителям аналитики



# Минусы

- Сложно с доступами
- Долго обрабатываются данные
- Тестируем бекенд, а не приложение



# Что за база такая?

У нас – своя система  
Ozon.Tracker

Можно использовать любую:

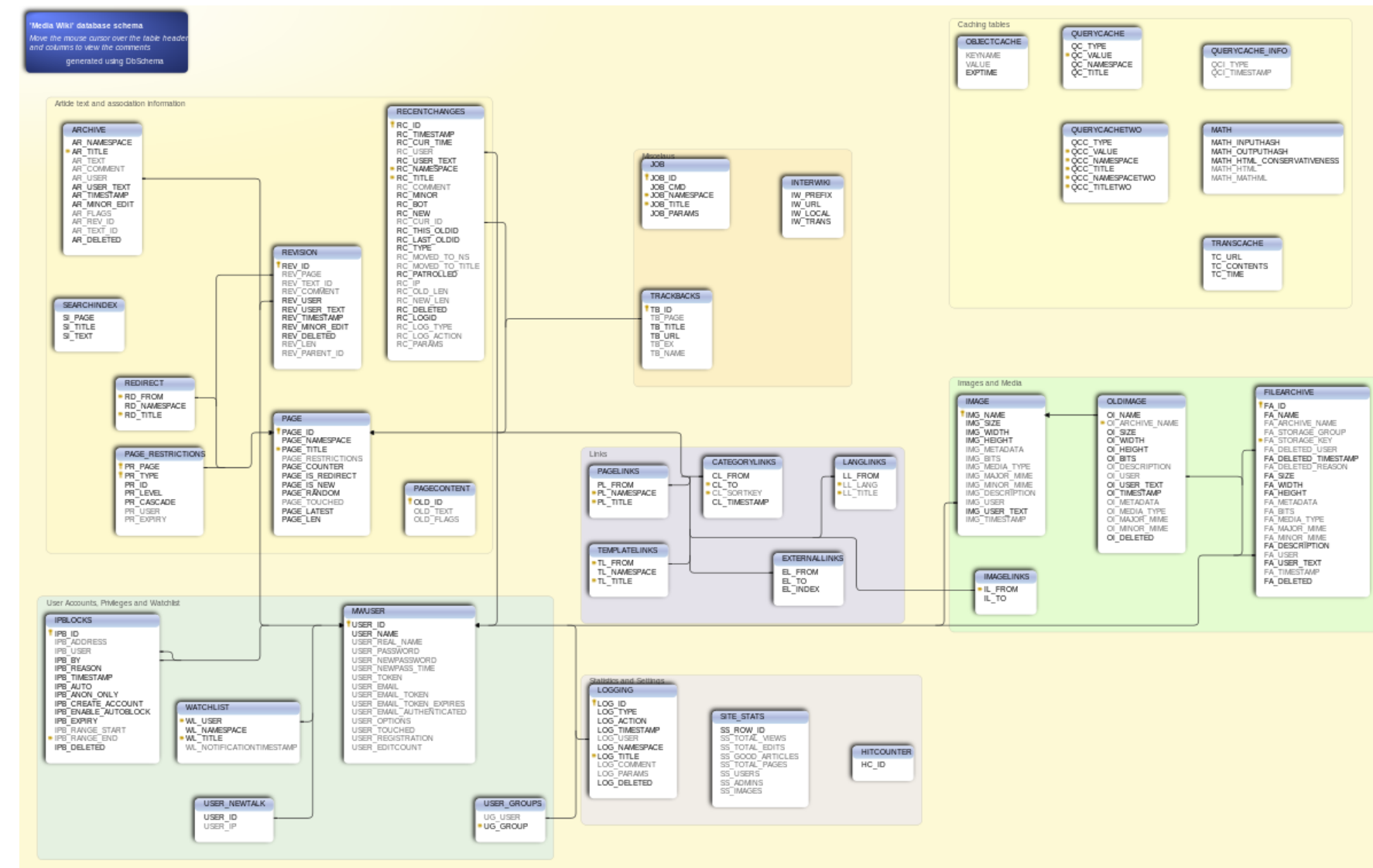
- Google Analytics API
- API Яндекс.Метрики
- И т.д.





# БД

1. Доступы получили
2. Переписали тесты



**UI отдельно, аналитика отдельно**

Разделили тесты, которые создают события и которые проверяют события

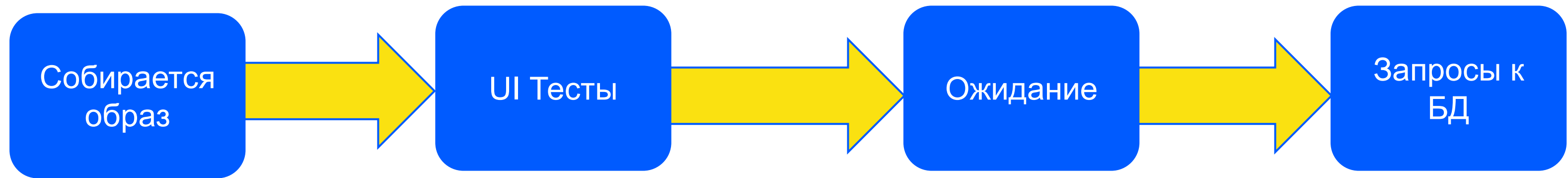


**Универсально под обе платформы**

**Просто отправляем запрос к базе**



# Как проходят



# Фейлы могут быть от запросов к БД

- Время на девайсе/на сервере/в реальности
- Неправильный запрос

# Как понять чьи события

- Все пользователи должны быть авторизованы
- У каждого теста своя учетная запись



## Как выглядит тест. Авторизация

```
page.menu.profile.wait_visible()  
page.menu.profile.click()  
page.auth_actions.input_phone_number(phone)  
page.auth_actions.auth_by_phone_sms(phone, '1111', user['ID'])
```

## Как выглядит тест. Генерация событий

```
page.deepLink_actions.open_deepLink(deepLink)
```



## Как выглядит тест. Форспуш

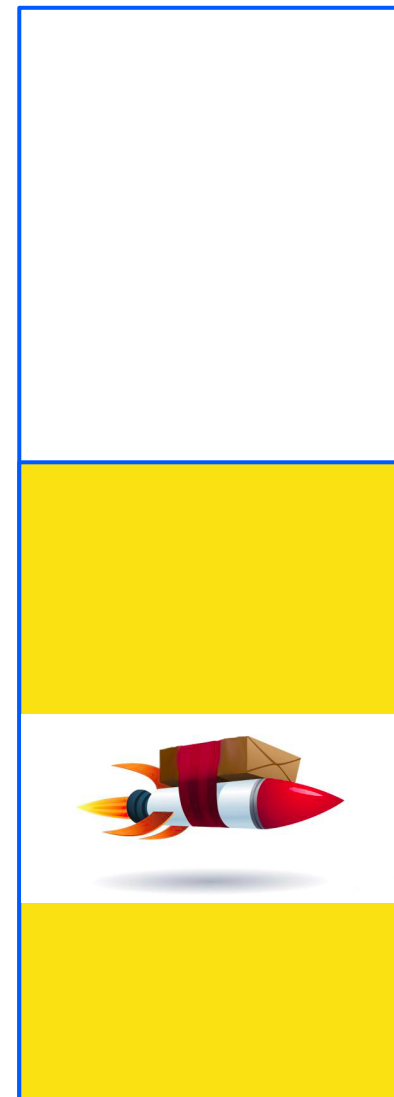
```
deepLink = "ozon://products/%s" % item_id  
page.deepLink_actions.open_deepLink(deepLink)
```

# Форс пуш

Нет данных



Отправка



## Как выглядит тест. Аналитика

```
prod_user_pdp = ProdUser.clickhouse_pdp()  
prod_user_id = prod_user_pdp['ID']  
page_id = 'pdp'  
assert clickhouse.page_view(page_id, prod_user_id, platform)
```

# Как выглядит тест. Запрос

```
query = "select attributes_platform, action_type, object_type, page_current, user_client_id from tracker.events " \
        "prewhere date >= today()-1 and attributes_platform = '{}'" \
        "and server_timestamp >= now() - {} and object_type = 'page' and page_current = '{}'" \
        "and user_client_id = {} " \
        "limit 1".format(platform, self.get_time(), page_id, user_id)
```

## Что проверяем

- Что события отправляются
- Работу форс-пушинга событий
- Корректность отправки событий

# Перфоманс трекинг

- Network
- Отрисовка картинок
- Загрузка картинок



# Перфоманс трекинг. Автоматизация

- Отправка
- Значения не нулевые



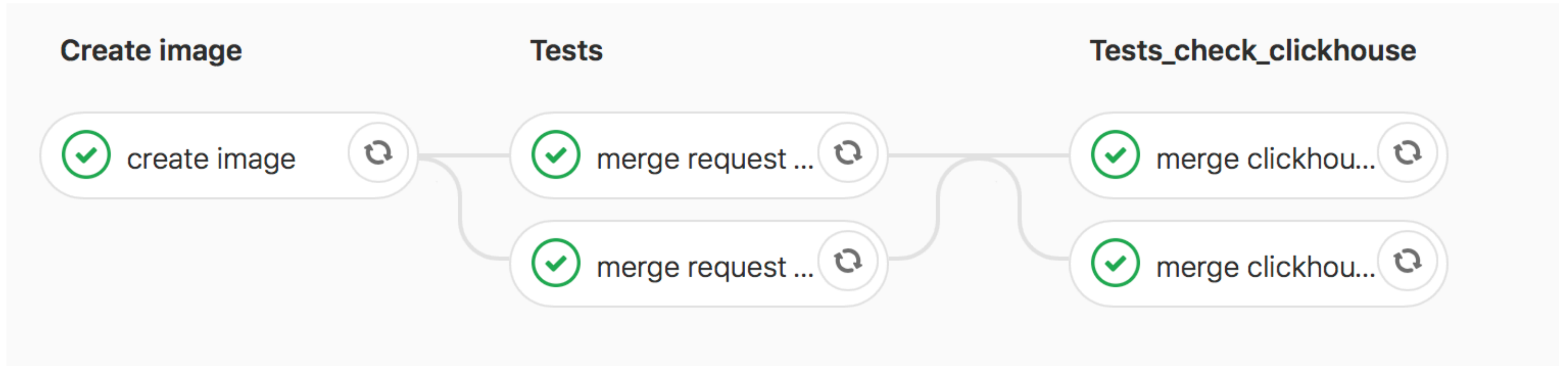
## Как используем

- При сборке «релизного» билда
- Ежедневный прогон по утрам
- При каждом merge request

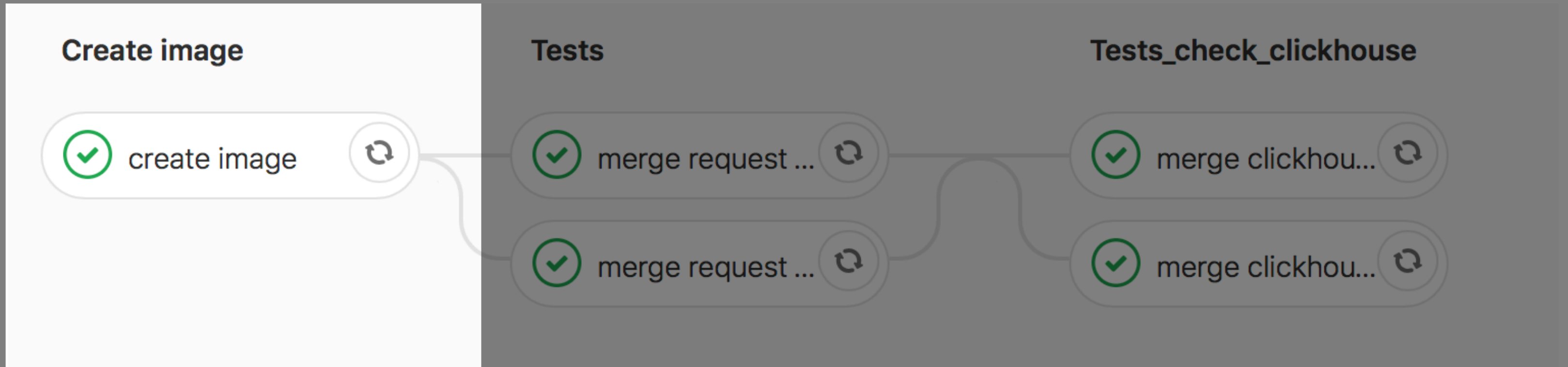




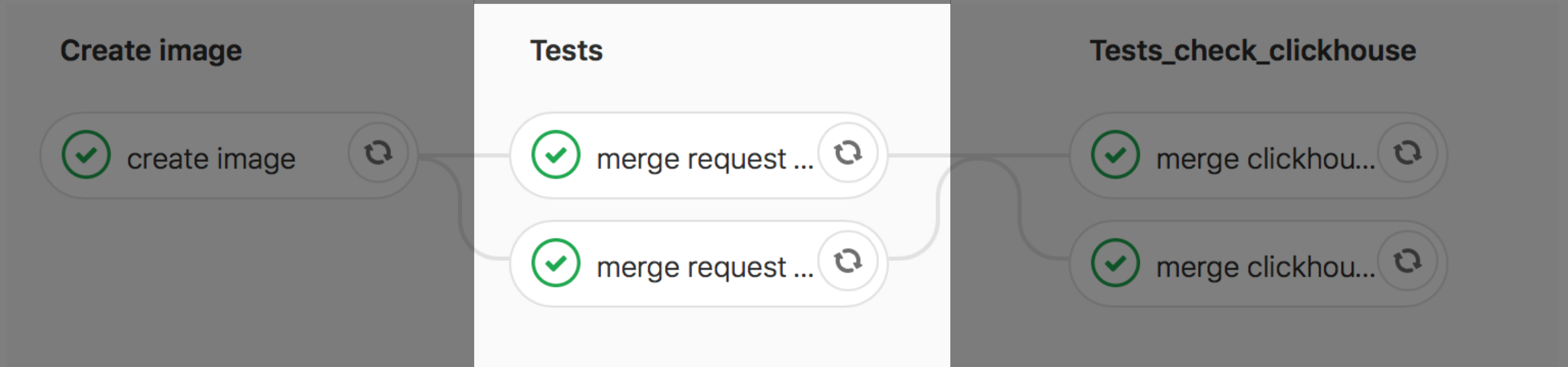
# Pipeline Merge Request



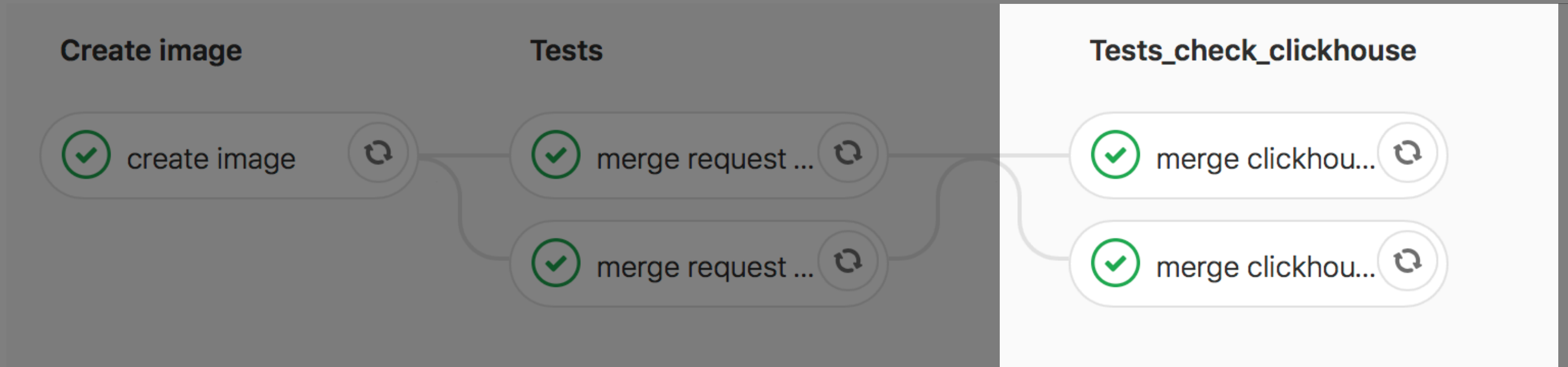
# Pipeline Merge Request



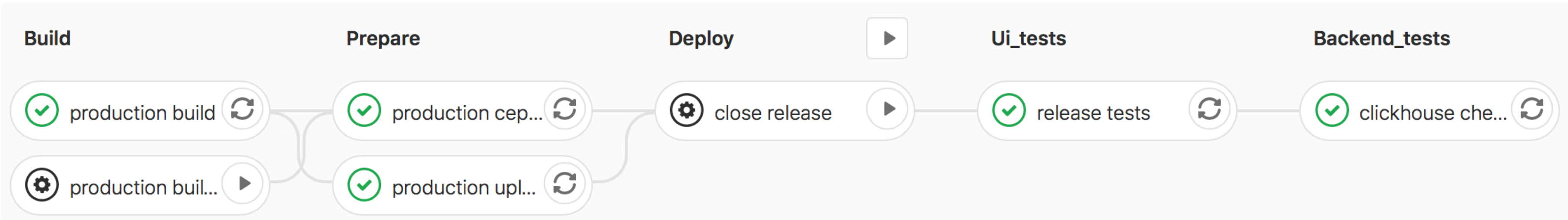
# Pipeline Merge Request



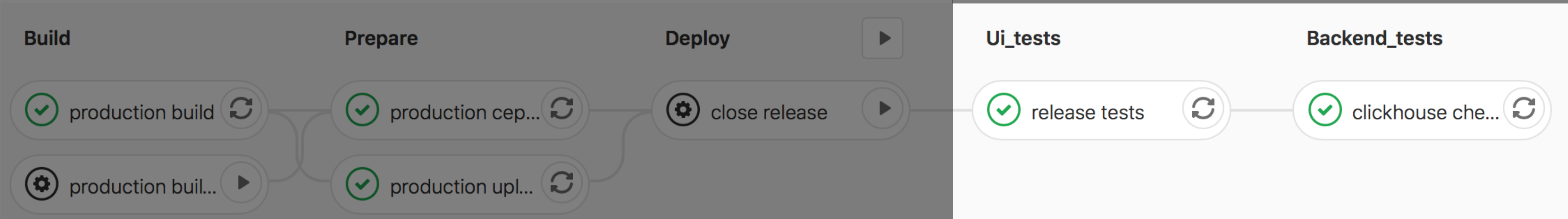
# Pipeline Merge Request



# Pipeline Release



# Pipeline Release



## Что еще надо сделать

- «Большие» тесты на «полный» батч
- Увеличить количество систем, которые мы проверяем
- Сделать комбинированные тесты Сниффер+БД

## Что мы получили

- Регулярные проверки
- Сразу узнаем о проблемах и заводим баги





В чем сложности

Как раньше тестировали

Как улучшить



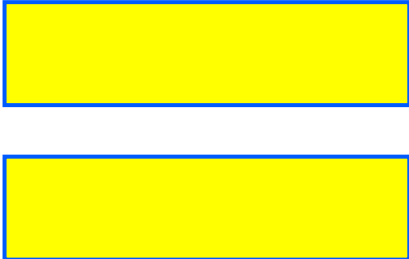
Как можно автоматизировать

Вывод

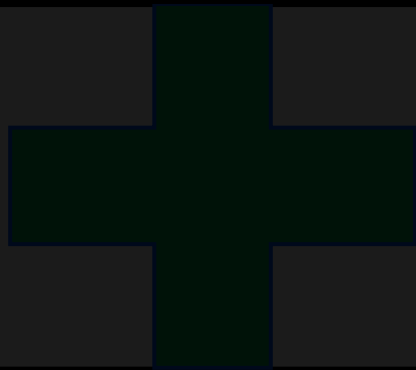
# Сравнение подходов

	Логи	База Данных	Сниффер
Скорость			
Сложность настройки			
«Честность» проверки			


# Сравнение подходов

	Логи	База Данных	Сниффер
Скорость			

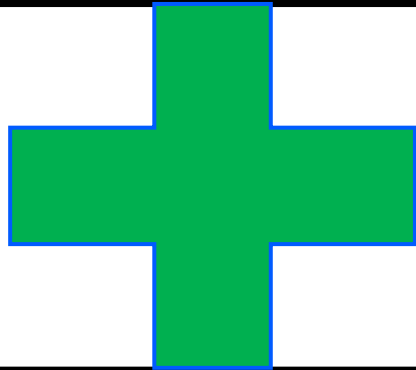


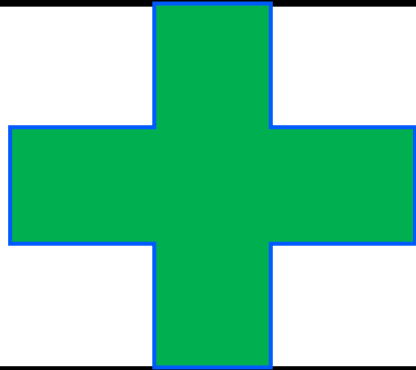
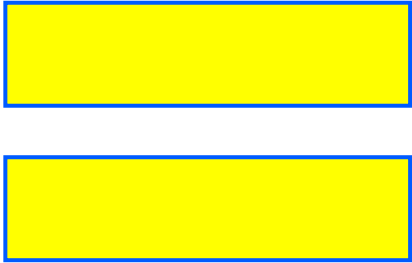



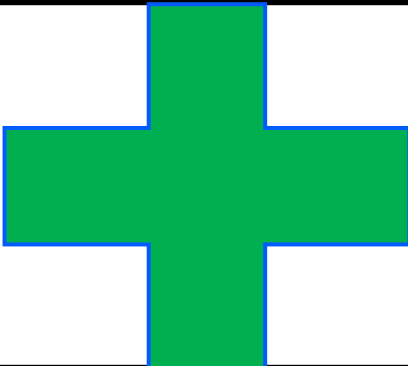
# Сравнение подходов

	Логи	База Данных	Сниффер
Скорость			
Сложность настройки			

# Сравнение подходов

	Логи	База Данных	Сниффер
Скорость			
Сложность настройки			
«Честность» проверки			

# Сравнение подходов

	Логи	База Данных	Сниффер
Скорость			
Сложность настройки			
«Честность» проверки			

# Вывод



**OZON**

**Спасибо за внимание!**

Павел Щедухин

 [pshchedukhin@ozon.ru](mailto:pshchedukhin@ozon.ru)