

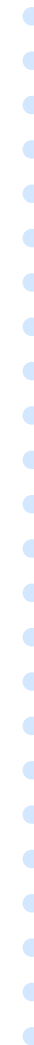
Расширяем покрытие приложениями и устройствами

Курпаченко Даниил,
ВКонтакте

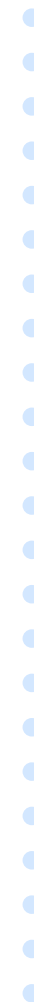
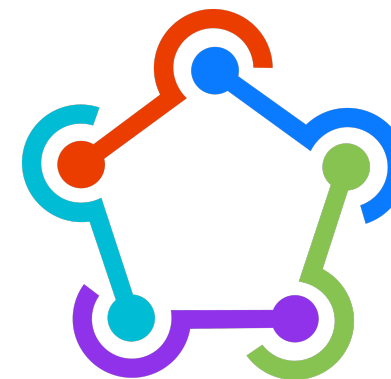
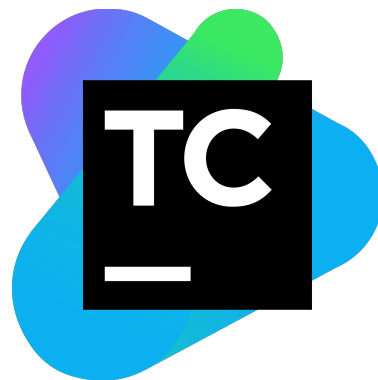
Кто я?

Курпаченко Даниил

Инженер по тестированию
в Core iOS QA



Что мы используем?



UI-тесты

- запуск на каждый НОВЫЙ КОММИТ
- параллелизация
- запуск тестов через ТС на ветках
- вовлечённость продуктовых команд
- «Хаки»

1000+

Тестов

~10 мин.

Smoke-прогон

5

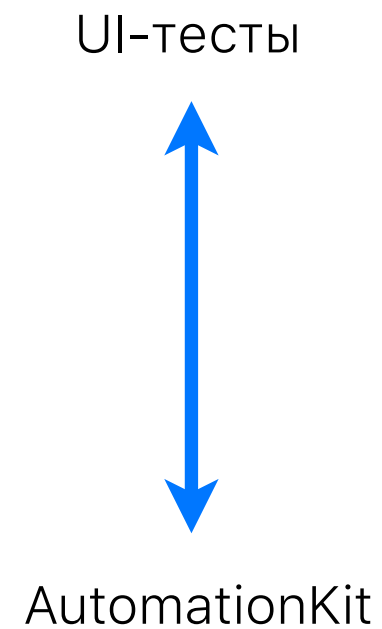
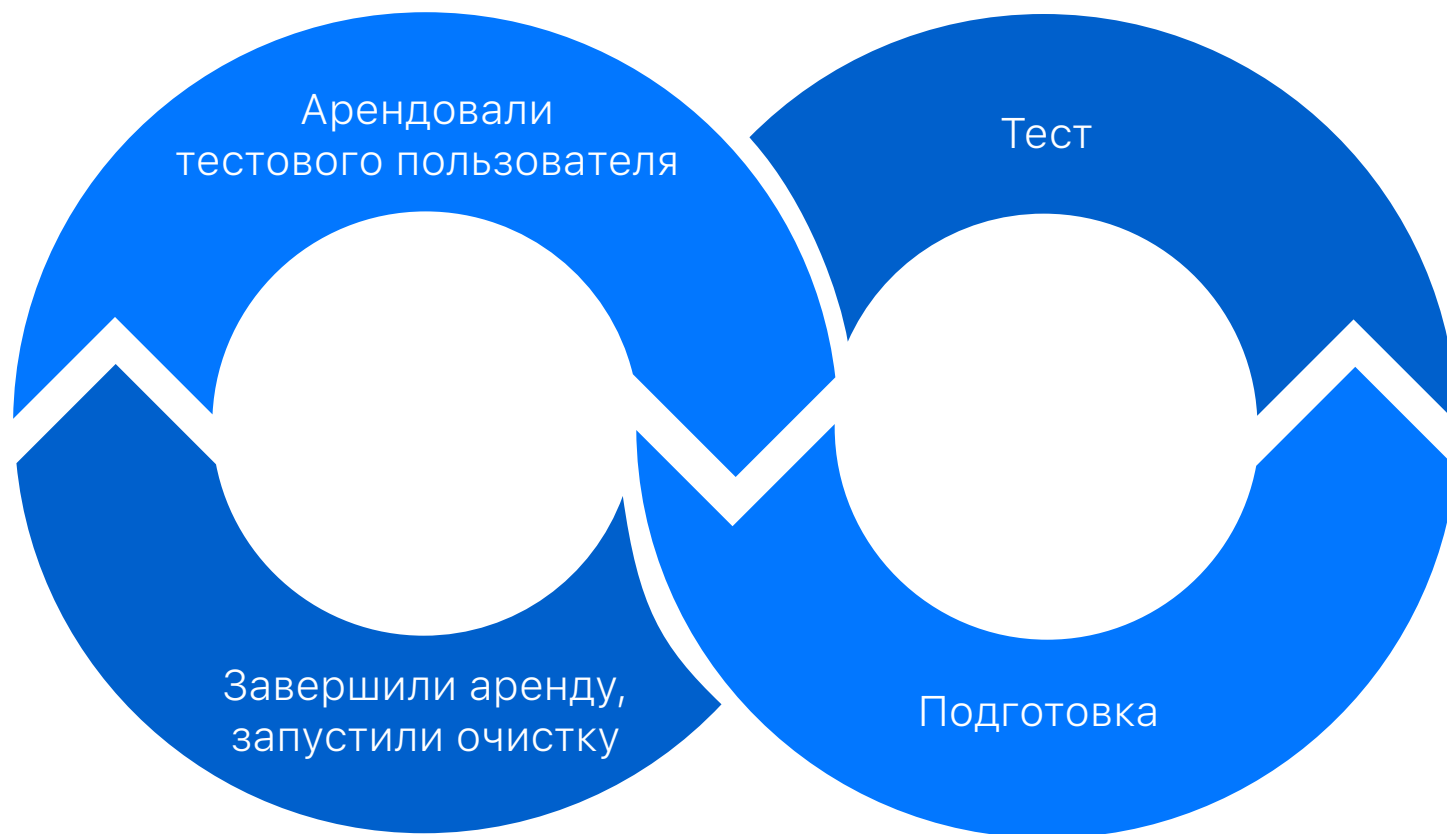
Батчей делят
1 прогон



4

Приложения
со своими UI-тестами

Немного об архитектуре



Структура пайплайна

Создаём
или переиспользуем
симулятор iOS



Собираем
приложение
и тесты



Передаём
результаты сборки



Создаём
или переиспользуем
симулятор iOS



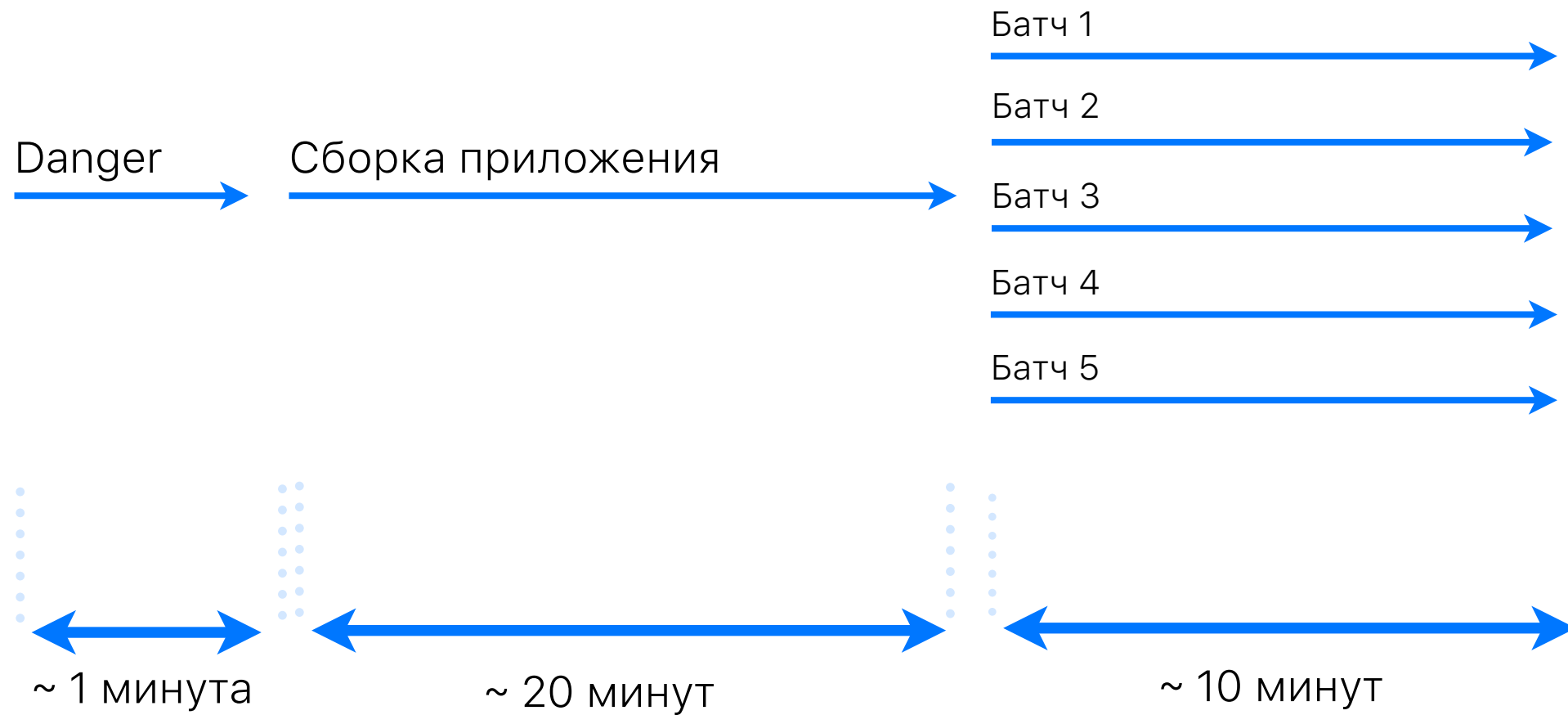
Запускаем
тесты



Mac билд-агентов



Структура пайплайна



В предыдущих сериях



- Просто реализовать
- Легко покрывать общие сценарии



- Сложно поддерживать
- Противоречит нормальному стилю
- Зависимость тестов от сборки нескольких приложений





Хотим тестировать
больше



Apple Watch?

Предыдущий доклад про Apple Watch



Особенности автоматизации на watchOS

- Наличие пары девайсов — Apple Watch и iPhone
- Основное приложение на iOS и приложение-компаньон на watchOS
- WCSSession не контролируется никем, кроме Apple
- Мало информации в открытом доступе

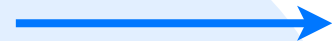


Структура пайплайна

Создаём
или переиспользуем
симулятор iOS

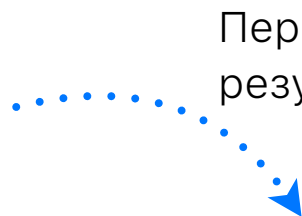


Собираем
приложение
и тесты

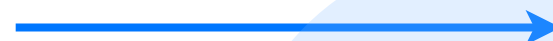


Компаньон на часы
собирается вместе
с основным
приложением

Передаём
результаты сборки



Создаём
или переиспользуем
симулятор iOS



Подготавливаем
окружение
под часы

Запускаем
тесты



Подготовка под тесты часов

1

Создать симулятор iOS

```
SimCtl.create_device
```

2

Создать симулятор WatchOS

```
SimCtl.create_device
```

3

Создать пару между iPhone и AW

```
xcrun simctl pair
```

4

Получаем тестового пользователя

5

Запустить основное приложение с дальнейшей авторизацией

```
xcrun simctl launch
```

6

Запустить приложение на часах

```
xcrun simctl launch
```

7

Запустить тесты

Ещё проблемы?

Параллелизация - клон хуже
синхронизируется с телефоном



Отключаем параллелизацию
в настройках схемы тестов

Что используем?

iPhone SE 2 gen



Apple Watch 5



Прогон тестов





А что
с тестами-то?



А что с тестами-то?

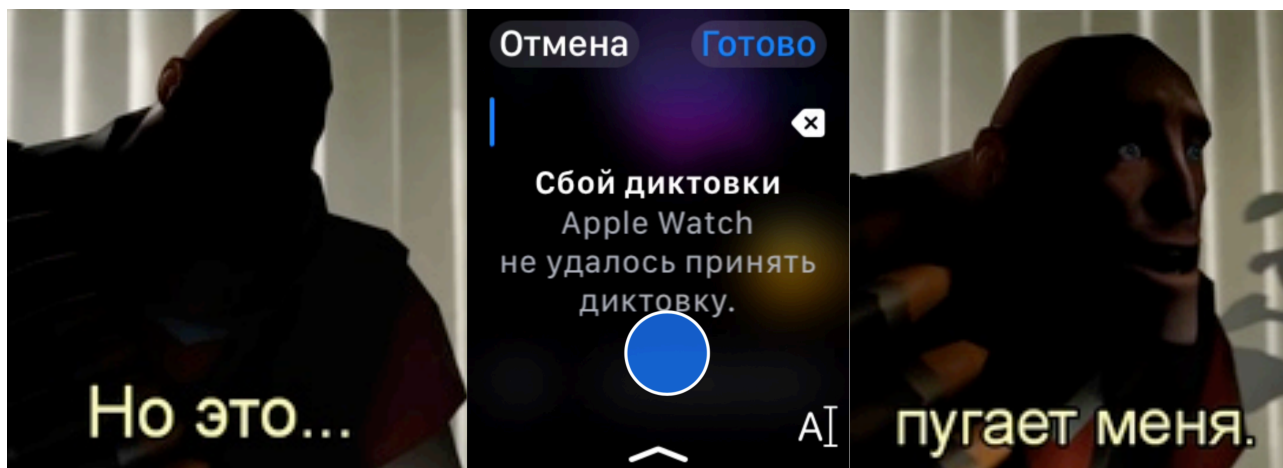


swipe

press

tap

coordinate



Сценарий ввода текста

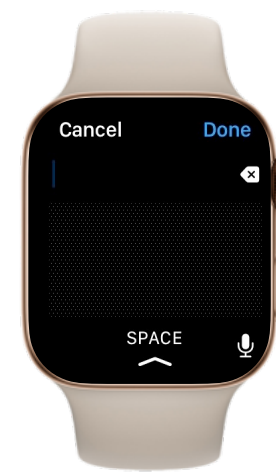
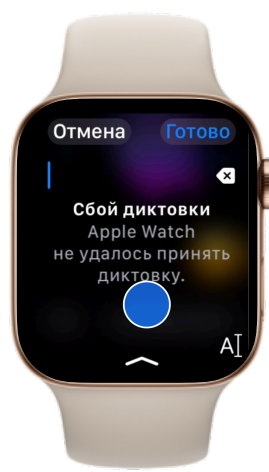
- Пермишены не обрабатываются из коробки
- Диктовка не работает
- Клавиатура не на всех девайсах



Точнее пытаются,
но не обрабатываются



Series 7,8,9 Ultra,
Ultra 2



Итоги к этому часу

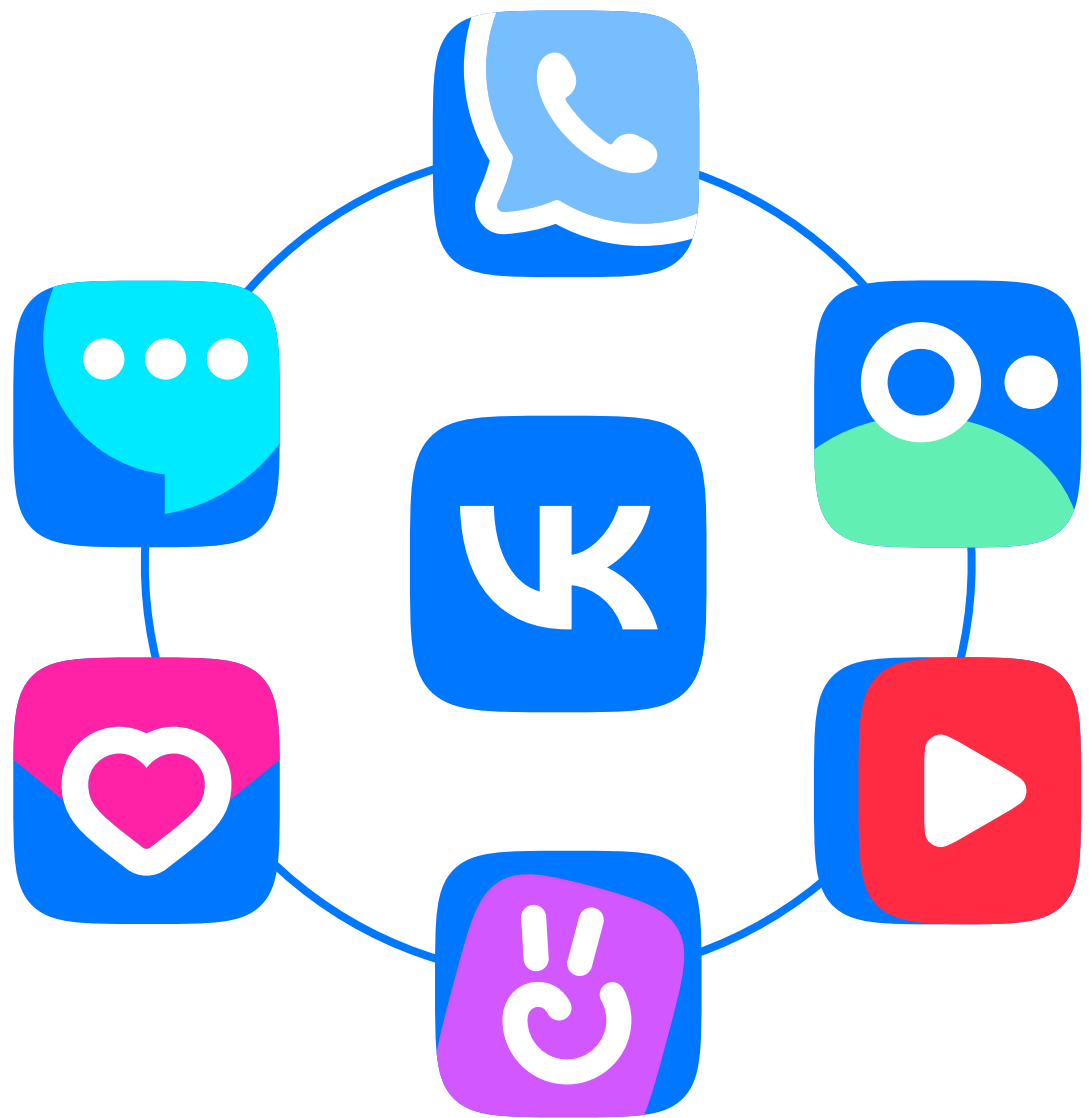
- Часы - новый девайс - требуют дополнительной настройки
- Синхронизация девайсов - штука нестабильная
- Приходится обрабатывать перемешены
- Ввод теста методом проб и ошибок



Другие приложения?



Надо!



Особенности приложения клипов

- Полноценное отдельное приложение
- Есть анонимный и неанонимный режим
- Отсутствие базы для автоматизации в приложении



Рoadmap

1

Подготовить MVP

2

Реализовать
возможности
основного таргета

3

Добавить
авторизацию

4

Покрыть основные
функциональности
приложения

5

Решить проблемы

6

Стабилизировать
прогон

7

Добавить
запуск на MR



Подготовить MVP

- Создаём таргет

- Подключаем фреймворки

- Пишем простейшие тесты



Выносим общую часть в пакеты
для переиспользования



Хорошая возможность
начать без легаси

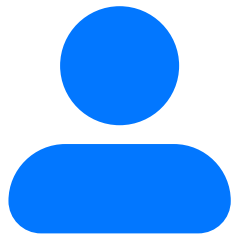


Прокачка тестов

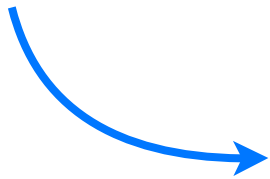
- Хаки навигации
- Отключение онбордингов
- «Хаки» для покрытия конкретных сценариев
- Систематизация работы с пермишенами



Авторизация



Арендванный
юзер



```
{  
  ...  
  id: ...  
  login: ...  
  password: ...  
  token: ...  
  ...  
}
```

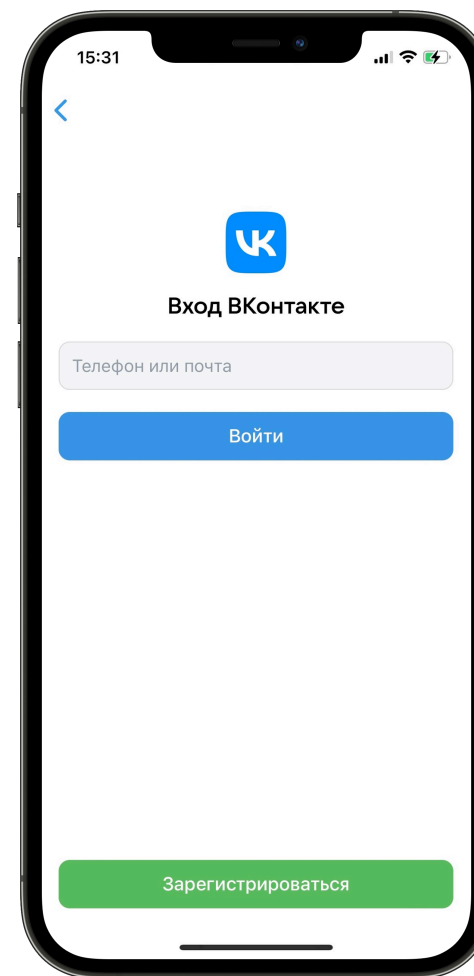


«Честный» путь

```
{  
  ...  
  id: ...  
  login: ...  
  password: ...  
  token: ...  
  ...  
}
```

→ Честно - через UI

→ Не очень честно -
в коде



Быстрый путь

```
func setupUITestsAuth() {  
    /// Достаём токен  
    ...  
    /// Токен есть - создаём сессию  
    ...  
    /// Токена нет - завершаем активные сессии  
    ...  
}
```

- Риск получить отклонения от «настоящих» результатов
- Сложность реализации
- Зависимость от токена



По токену



Покрыть основные функциональности приложения

... и нарваться на проблемы

- Анимации



Отключаем
для тестов

- Хинты



Отключаем
для тестов

- Баги



Правим :)

- Фича-
ТОГГЛЫ



Тесты и тогглы

- Бэкенд умеет отличать ботов
- Включение фичи из теста

Параметризированные
прогоны

Подготовка тестов

Боты UI-тестов

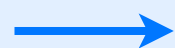
АБ/Эксперименты



Стабилизировать прогон

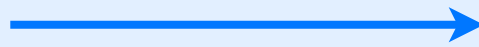
- Упрощение длительных нестабильных тестов
- Атомарность проверок
- Отказ от покрытия нестабильной функциональности

Следим за временем прогонов тестов



Упрощаем

Следим за стабильностью
конкретных тестов



Правим/
Убираем



Добавить запуск на MR

- Тесты критичны и блокируют влитие
- Дежурные разбирают проблемы и мьютят/отключают тесты
- Ответственный правит тест



Что всплывает?

- Краши
- Плавающие баги
- Нестабильные тесты



И что по итогу?

- Новый таргет - хорошее поле для экспериментов
- Не нужно перетягивать всё
- Не всё так просто
- Запустить всё и сразу не получится



Реальный девайс?



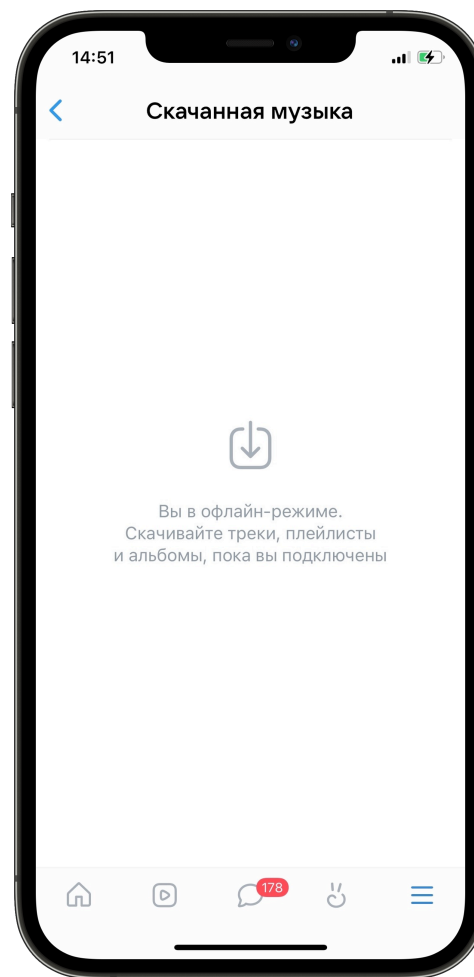
Отличия симулятора от девайса

- Используются ресурсы мака
- Отсутствие некоторых возможностей (управление сетью, камера, ...)
- Сложность реализации отдельных сценариев
- Сетевые запросы идут с девайса



Оффлайн-музыка - зачем?

- В нашем случае скачивание не работает на симуляторе
- Оффлайн сценарии для симулятора практически не реализовать
- Интересная задача



Вернёмся к пайплайну

Создаём
или переиспользуем
симулятор iOS



Собираем
приложение
и тесты



Передаём
результаты сборки



Создаём
или переиспользуем
симулятор iOS



Запускаем
тесты



Вернёмся к пайплайну

Получаем
подключённый
девайс



Собираем
приложение
и тесты



Передаём
результаты сборки



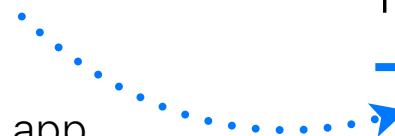
Получаем
подключённый
девайс



Запускаем
тесты



Конвертируем app
в ipa и ставим на девайс



Ищем подключённый девайс



И в этом нам **НЕ** поможет
fastlane DeviceManager



Почему?



Почему?



Получаем список
USB-девайсов



Получаем список девайсов
через `instruments -s devices`



Находим совпадение USB-
девайса с выводом `instruments`



Возвращаем `Device()`



Deprecated since Xcode 13



Нужен ли нам весь объект?



Почему?



Получаем список USB-девайсов



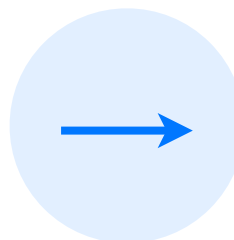
Получаем список девайсов через `instruments -s devices`



Находим совпадение USB-девайса с выводом `instruments`



Возвращаем `Device()`



Получаем список USB-девайсов



Получаем список девайсов через `xctrace list devices`



Находим совпадение USB-девайса с выводом `xctrace`



Находим нужный нам девайс по имени



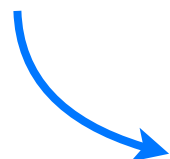
Возвращаем `udid`



Превращаем собранный app в ipa

Создаём

директорию Payload



Переносим наш .app
в эту директорию



Сжимаем
директорию в архив

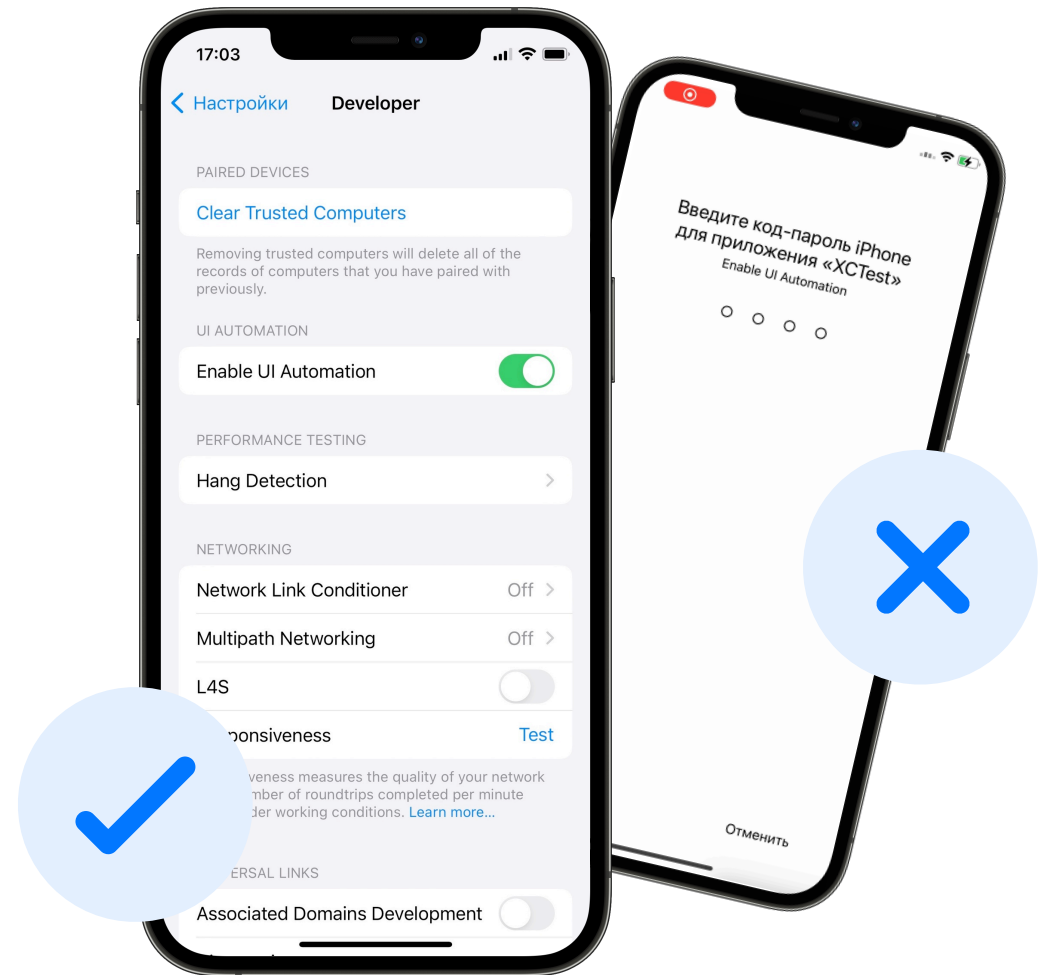


Меняем .zip на .ipa



Что с девайсами?

- Добавлены в аккаунт разработчика
- Отсутствует код блокировки
- Разрешена автоматизация в настройках разработчика



Изменения в сценариях

- Необходимость симулировать оффлайн
- Чистки за тестами
- Зависимость скачивания от сети



Реализуем очистку треков в коде



Готовим «быстрые» треки



Симуляция оффлайна



Симуляция оффлайна

- Siri
⋮
Долго
- Шторка
⋮
Долго
- Action button
⋮
iPhone 15 Pro
- Системные настройки

Открываем системные настройки



Переключаем, если нужно



XCUApplication.launch() не завершит предыдущую не тестовую сессию



Что получилось

- ✓ 1 мак, 1 айфон, готовые скрипты для запуска тестов по триггеру
- ✓ Покрытие основной функциональности
- ✓ Готовность к масштабированию

- Тесты на физических девайсах вполне могут быть оправданы
- Свои особенности и лимиты
- Баги, баги, баги



За рамками тестов



Mac и iPhone с привилегиями
во внутреннем контуре



- Информационная безопасность
- Физическая безопасность

Выводы

Девайс vs симулятор



- Затратней
- Поломки девайса
- Девайс медленней
- Сложности эксплуатации



- Возможность покрыть некоторые сценарии



Что есть уже сейчас

4

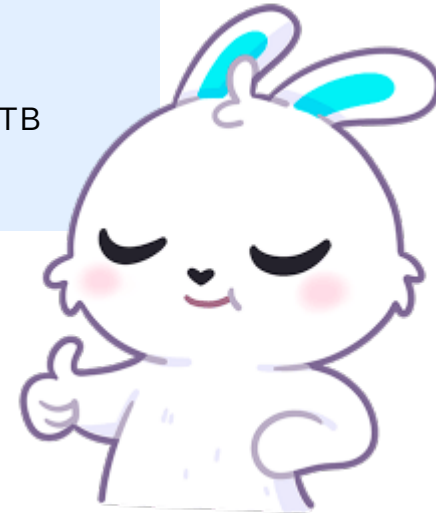
Приложения
покрыто тестами

5

Таргетов
UI-тестов

3

Типа устройств



Выводы

- Новые девайсы, конечно, несут новые сложности
- Ожидаемые решения не всегда работают, как хотелось бы
- Почти всегда можно выбрать или создать наиболее удобное решение



Артефакти



Запуск нового таргета



Физ.девайсы



Будем ВКонтакте!



vk.com/blin