

Опыт портирования нативных приложений на React Native

AnjLab **TechTalks** 2018 г. Владимир

Королев Юрий

Дисклеймер

все сказанное может быть уже устарело и не правда

- Релизы React Native выходят каждый месяц
- все актуально для RN v43 (Сейчас 54)

История приложений

"Клиентское"

- ~3 года iOS
- ~2 года Android

"Партнерское"

- ~6 мес iOS
- ~6 мес Android

Зачем мы решили переписать на RN?

- Убрать технический долг
- Синхронизировать релизы на платформах
- Сровнять скорость разработки мобильных клиентов с веб клиентом
- Снизить стоимость разработки

Исходные технологии

iOS

- Swift
- YapDatabase
- RxSwift
- Redux

Android

- Java
- Kotlin
- RxJava
- Realm

Используемые технологии

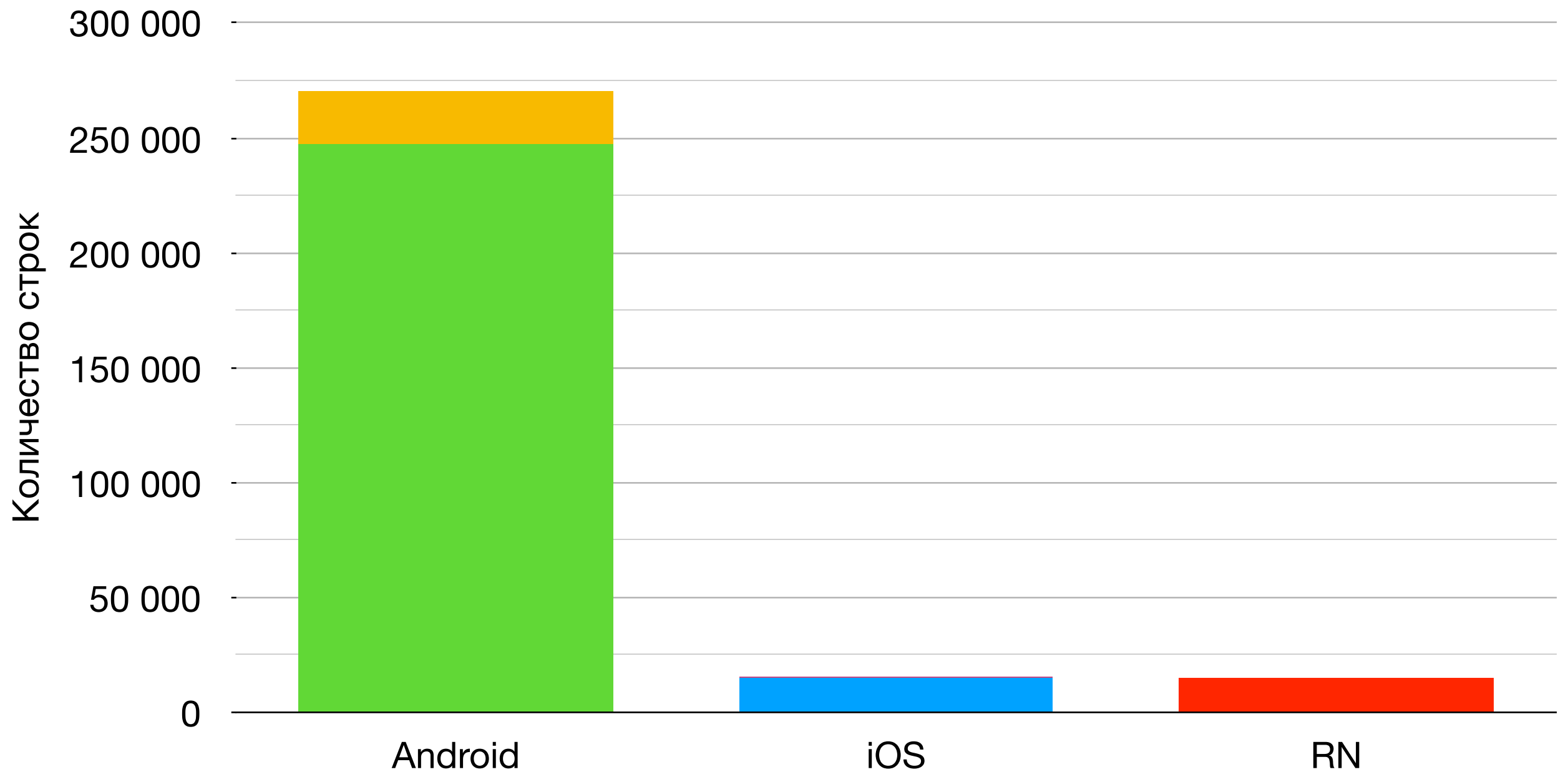
- Javascript: ES6 + babel
- Навигация react-navigation
- База данных: Realm
- Архитектура: redux + redux-saga
- Типизация: flowtype
- Тесты: jest
- Менеджер пакетов: yarn
- Нотификации: firebase messaging
- Сбор крашлогов: sentry.io + crashlytics
- fastlane + twitter app.beta

Время

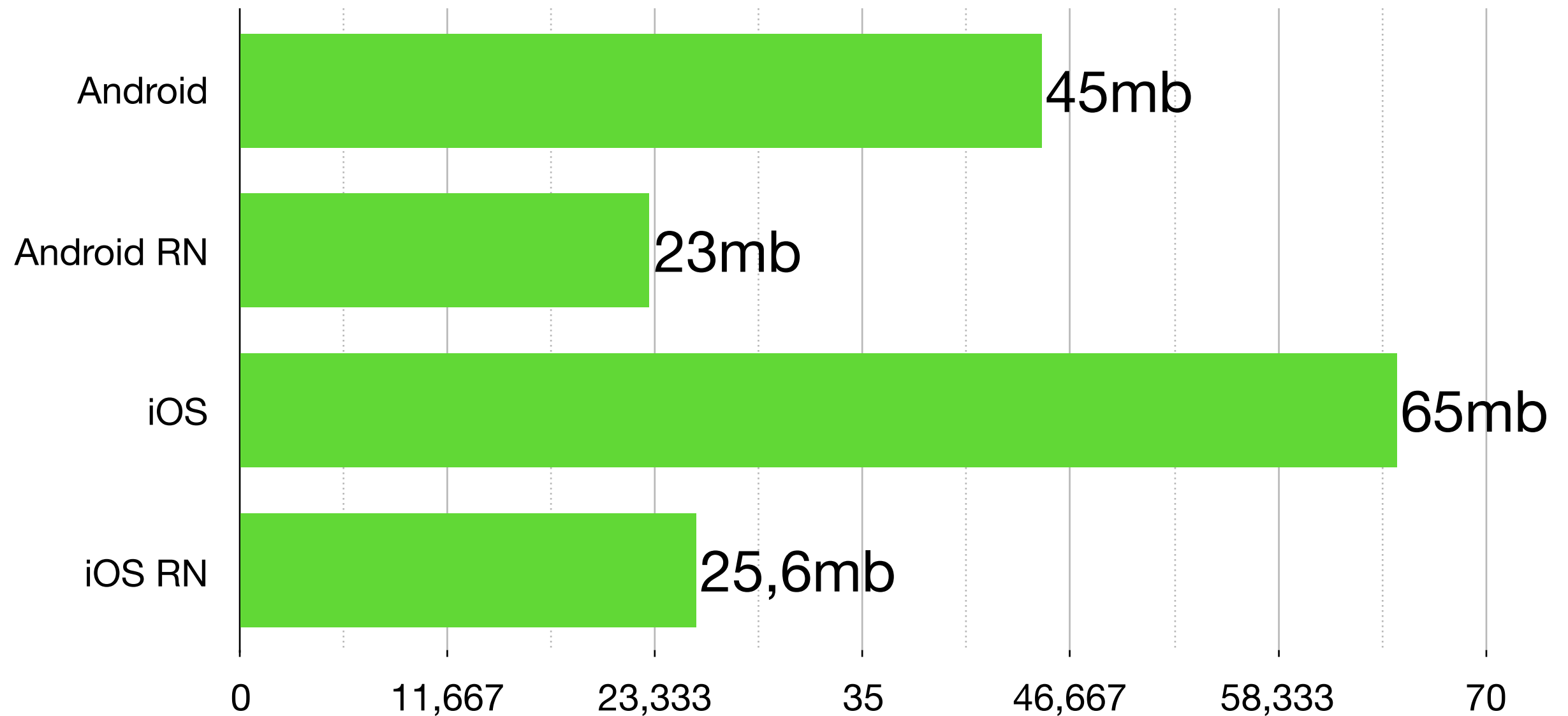
- 2 месяца - готовый функционал
- 3 месяца - релиз

Кодовая база

Swift Java Kotlin JavaScript Obj-C



Размер приложения



Покрывѳіе тестами

64%

Трудности: сеть

- Отмена запросов (пришлось патчить)
- MQTT - пришлось реализовывать
- Сжатие - не включено по умолчанию
- Кеширование картинок (не было)

Трудности: Скорость старта приложения

- Реорганизация модулей
- Отложенные запросы к Realm
- Отложенная инициализация навигации

Экзотические баги на андроид устройствах

- Перевод каретки как сабмит (исправлено, обновлением RN)
- Не отображаются отраженные картинки на Huawei P9
- Layout анимация - исчезновение элементов

Трудности: Realm

- Скорость timestamp vs Date
- Живые коллекции и объекты
- Нет асинхронных запросов
- Ломает/тормозит дебаг

Трудности: сбор ошибок

- Crashlytics - не умеет RN
- Sentry.io не показывал стэк на Android. Пришлось патчить raven.

Трудности: выбор библиотек

- react-navigation vs wix/react-native-navigation
- mqtt - нет для двух платформ
- старые версии RN
- статусная нотификация - не подобрали (написали свою)
- Swift (вес)

Трудности: производительность

- Скорость навигации на iPhone 5 и ниже
- Больше потребляемой оперативной памяти (50mb vs 90mb)

Плюсы

iOS

- Поддержка iPad
- Поддержка горизонтальной ориентации
- Поддержка iOS версий 8.0 и выше

Android

- Сохранение состояния при навигации

Плюсы

- Единая кодовая база
- Размер приложения
- Скорость сборки
- Скорость разработки интерфейсов

Плюшки

- Огромный выбор редакторов
- Разработку можно вести прямо на устройстве (Blink)
- Прототипировать можно прямо в браузере (Expo)
- StoryBook!
- Prettier!

Требования

- Понимание ReactJS, Redux, Redux-Saga, FlowType
- Умение читать чужой код (мало документации)
- Опыт нативной разработки
- Компромиссы

Выводы

**Если еще не используете -
начинайте**

Спасибо за внимание

yury.korolev@gmail.com