

Тестирование мобильных приложений



Тестирование мобильных приложений - ЭТО НЕ Тестирование мобильных телефонов!



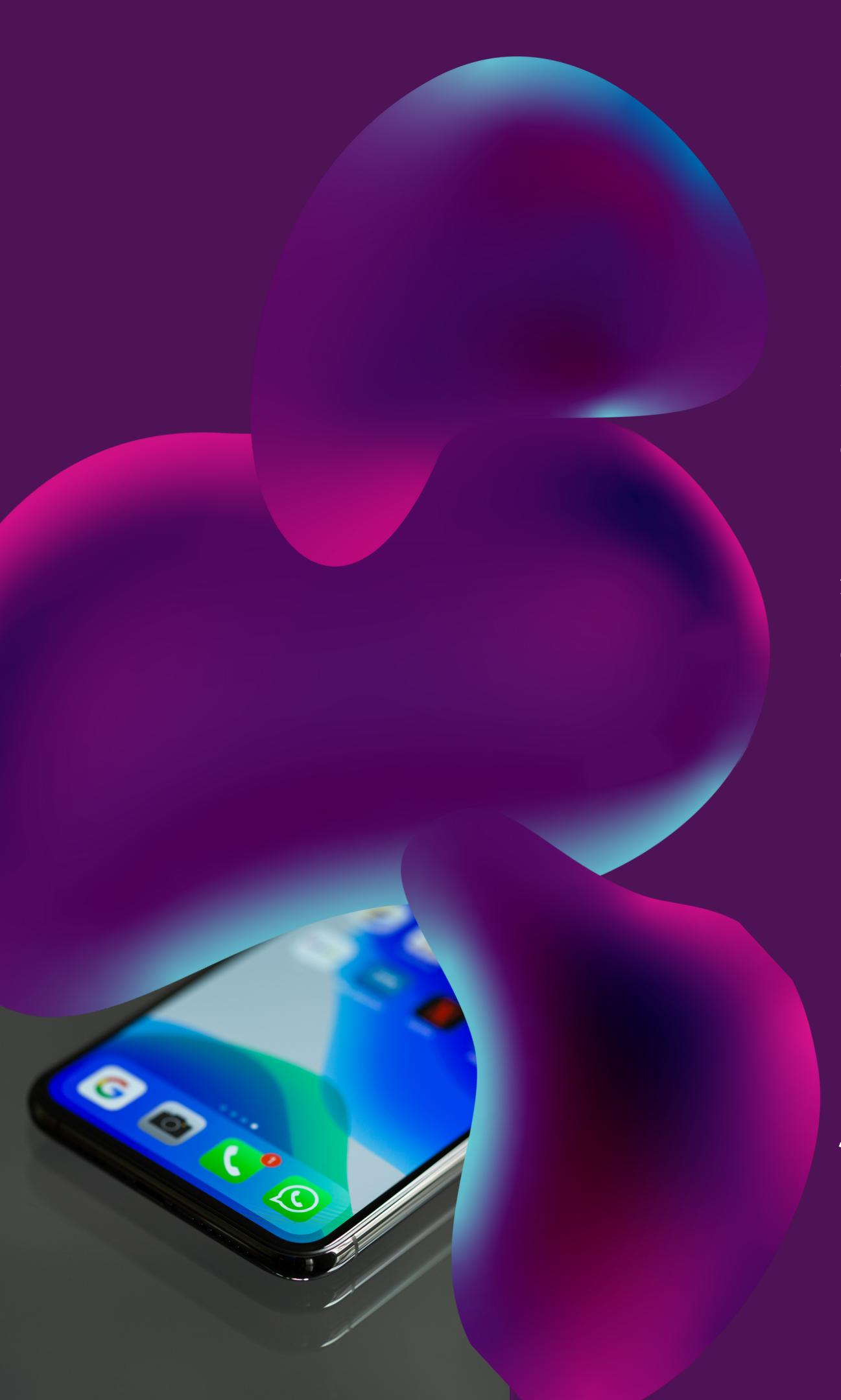
Спецификация мобильного приложения:

- 1. Мобильное устройство - не может работать как персональный компьютер.**
- 2. Мобильное устройство может менять ориентацию экрана.**
- 3. Основная задача телефона - это прием и выполнение вызовов.**
- 4. Операционная система мобильного телефона быстро устаревает.**
- 5. Мобильное устройство постоянно осуществляет поиск сети.**
- 6 Мобильные приложения должны поддерживать несколько входных каналов (клавиатура, голос, жесты), мультимедийные технологии и другие функции, повышающие их удобство использования.**

Типы приложения:
Для тестирования
мобильного приложения
большую роль играет его тип.
Существует три основных
типа мобильных приложений:

1. Нативные
2. Гибридные
3. Мобильные веб-
приложения





Нативное приложение - это приложение, разработанное специально для одной платформы (Android, iOS)

Эти приложения называются нативными от того, что они написаны на родном для определенной платформы языке программирования. Для Android этим языком является Java, тогда как для OS - objective-C или Swift. Нативные приложения находятся на самом устройстве. Они установлены через магазин приложений (Play Market на Android App Store на iOS). Они разработаны именно для конкретной платформы и могут использовать все возможности устройства. Потому как нативные приложения оптимизированы под конкретную ОС, они органично вписываются в любой смартфон, отличаясь высокой скоростью и производительностью.

Достоинства:

Нативное приложение работает в автономном режиме
Оно может использовать все функции своего устройства
Продвинутый пользовательский интерфейс
Push-уведомления для удобства пользователей

Недостатки:

Разработка в сравнении с мобильными веб-приложениями
обходится в разы дороже
Необходимы большие финансовые затраты на техническое
обслуживание



Мобильные веб-приложения

Мобильные веб-приложения, представляют собой сайт, который адаптирован и оптимизирован под любой смартфон.

Чтобы им воспользоваться, достаточно иметь на устройстве браузер, знать его адрес и иметь интернет-соединение.

При запуске мобильного приложения, пользователь выполняет все те действия, которые он выполняет при переходе на любой веб-сайт, а также получает возможность установить их на рабочий стол, просто создав закладку страницы сайта.

Веб-приложения отличаются кросплатформенностью, они способны функционировать, независимо от платформы девайса.

Веб-приложения не используют программное обеспечение девайса и не заполняют место в памяти смартфона.

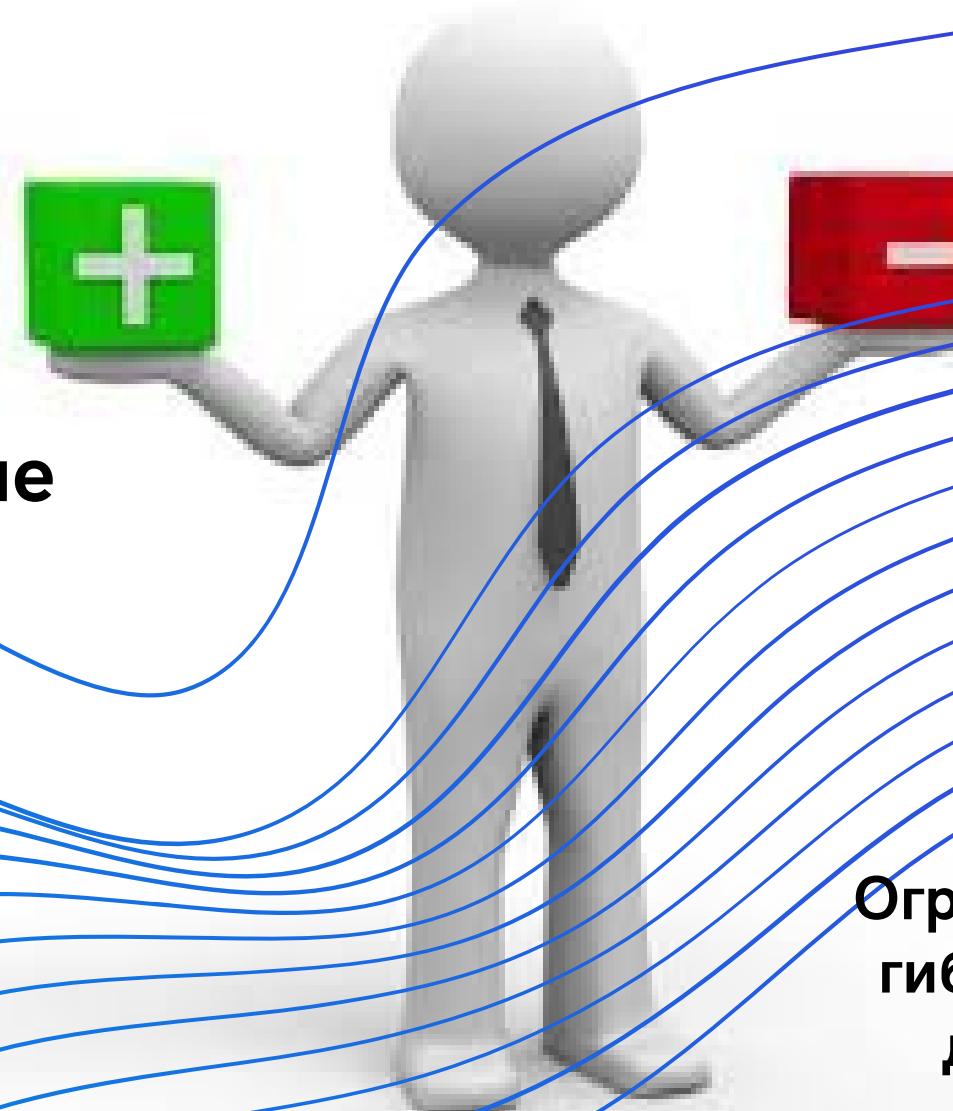
Разрабатываются они с помощью инструментов и фреймворков, что способствует высокой скорости их разработки.

Мобильное веб-приложение - это фактически веб-сайт, открытый с помощью мобильного браузера.

Достоинства:

- Простая разработка
- Простое обновление
- Легкий доступ

Мобильные веб-приложения не требуют установки



Недостатки:

- Нет поддержки автономных функций
- Ограниченнaя функциональность в сравнении с гибридными и нативными приложениями (нет доступа к файловой системе и локальным ресурсам)
- Проблемы с перераспределением: Google Play и app Store не поддерживают перераспределение мобильных веб-приложений.

Гибридные приложения

Представляют собой сочетание веб и нативных приложений. В особенности имеется в виду их кроссплатформенность и доступ к функционалу смартфона.

Такие приложения могут быть загружены исключительно из маркетов вроде Google и App Store.

Гибридные приложения располагают опцией автономного обновления информации, а для их работы необходимо интернет-подключение.

Эти приложения способны соединять достоинства нативных с технологичной актуальностью, которая обеспечивается последними веб-технологиями.

Однако, в отличии от нативных, стоимость создания гибридных на порядок ниже, а его скорость - выше.

Сходство с веб-приложениями, в свою очередь, дает возможность легко и оперативно вносить корректизы (разработчикам не приходится, как с нативными, повторно размещать приложение в магазине ради устранения ошибок предыдущей версии).

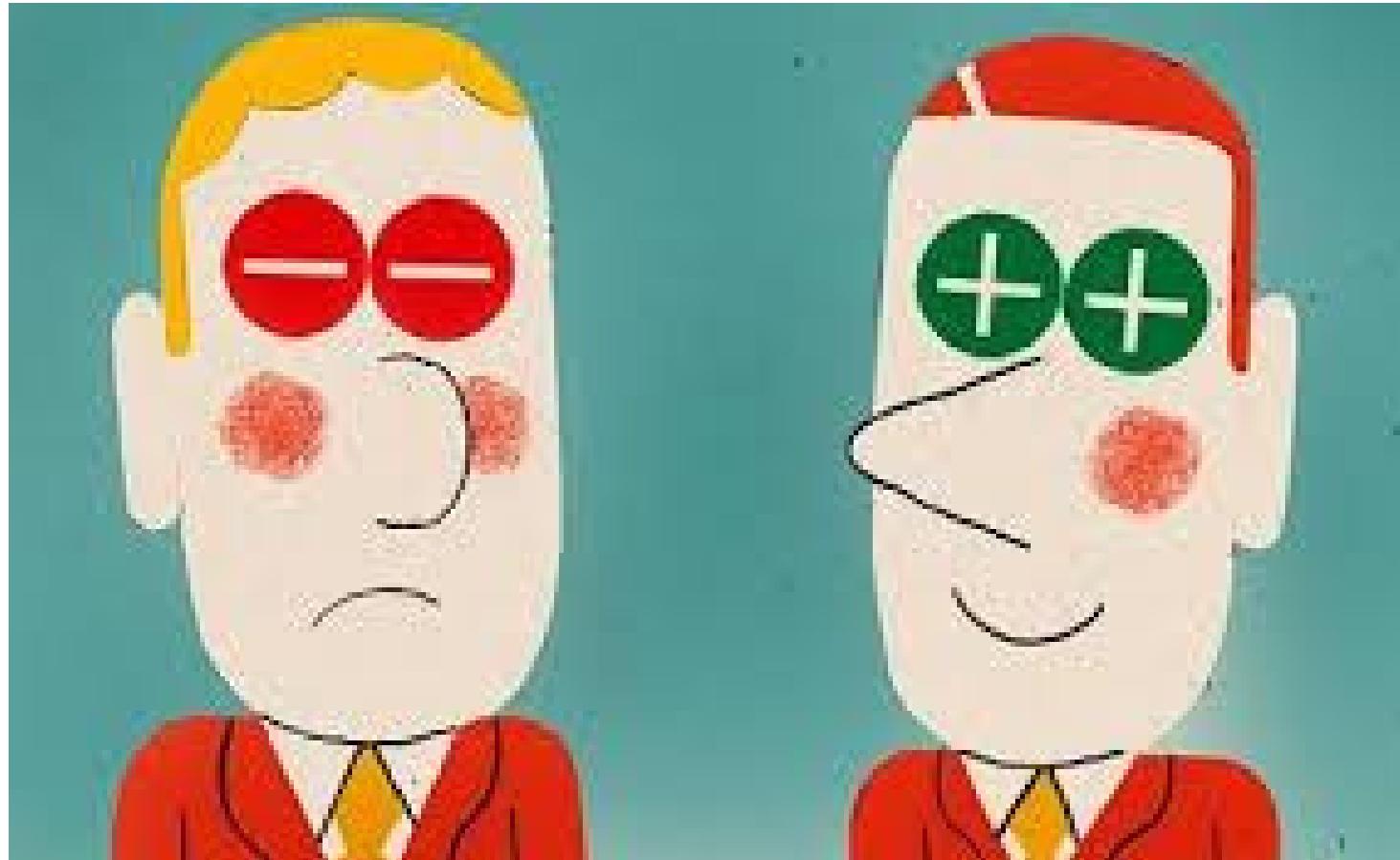
Разработка гибридного приложения перспективна еще и потому, что она подразумевает создание сразу под две платформы.

Качество и возможности гибридных приложений зависят от фреймворка, который использует разработчик.

Гибридное приложение - это сочетание нативного и мобильного веб-приложений. Его можно определить как отображение содержимого мобильного сайта в формате приложения.

Недостатки:

Работает не так быстро, как нативное приложение
Графика менее адаптирована к ОС в сравнении с нативным приложением



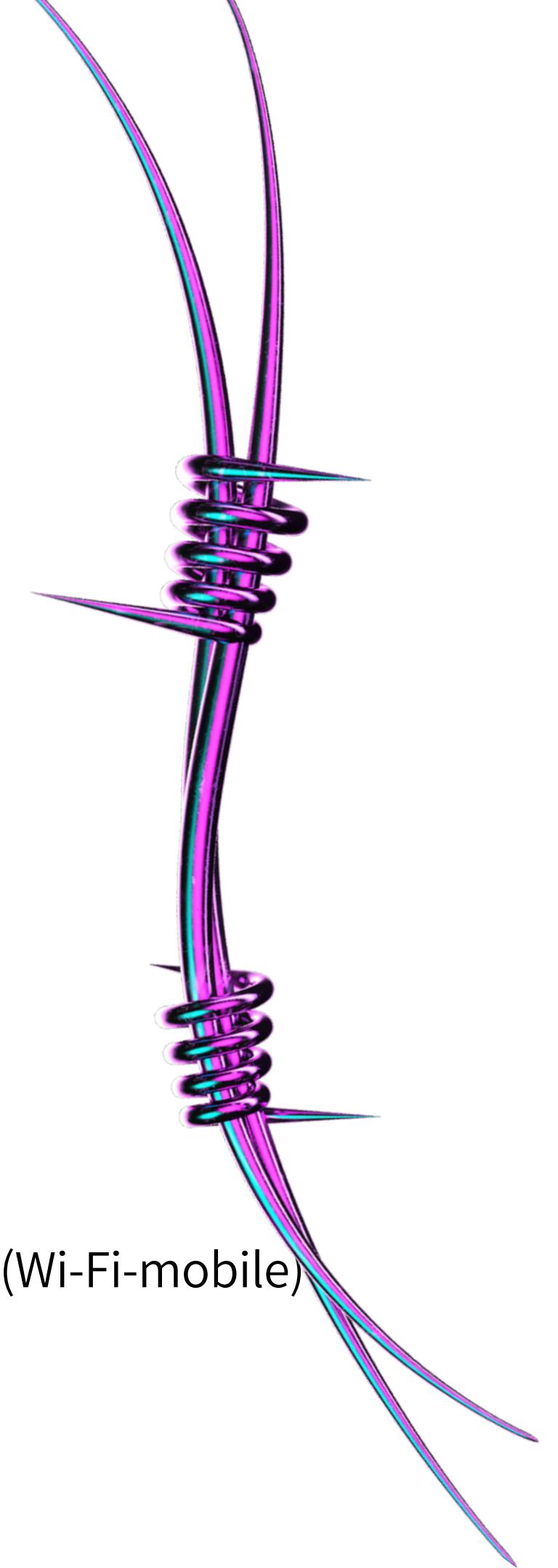
Достоинства:

Особенности устройства
Встроенный браузер
Простое распространение
Более рентабельно по сравнению с нативным приложением

Особенности мобильного тестирования



- 1) 2G/3G/4G/5G
 - слабый сигнал сети
 - прерывание сети
 - работа без сети
- 2) Wi-Fi
 - слабый сигнал сети
 - медленное соединение
 - прерывание сети
 - работа без сети
- 3) Переключение между сетями (Wi-Fi-mobile)
- 4) VPN сервисы



5) Входящие звонки, смс, push

6) Подключение зарядного
устройства или кабеля

7) Изъятие батареи

8) Низкий заряд батареи

9) Спящий режим

10) Многозадачность

11) Состояние:

-активное

- свернутое

- неактивное

- разделенный экран

12) Сенсоры:

- акселерометр

- гироскоп

- тачскрин

- датчики (GPS, NFS и т.д.)

13) Ориентация и
жесты:

- портретный/
ландшафтный

- мультитач
-тридетач

- физические кнопки
(громкость, питание)

14) Геолокация и
GPS:

- перемещения/
отслеживания

- точность
сигнала

- оповещения



15) Реклама:

- реклама не должна перекрывать кнопки управления приложением
- реклама должна иметь доступную кнопку закрытия, потому что чаще всего пользователь ее не ищет, а просто удаляет приложение

16) Локализация:

- верстка при переводе на другой язык
- соответствие формату установленного региона
- временные настройки должны быть соблюдены

17) Память и батарея:

Нагрузка и потребление:

- активный режим
- фоновый режим
- спящий режим
- работа с большим количеством данных

18) Установка и настройка:

- установка
- настройка
- удаление
- повторная установка
- подготовка тестовой среды и настройка мобильных девайсов

Основные виды мобильного тестирования



Функциональное тестирование
Нагрузочное тестирование
Конфигурационное
тестирование
Тестирование безопасности
Юзабилити тестирование

Функциональное тестирование

Чтобы правильно определить, какие кейсы необходимо проверить, необходимо понимать бизнес-функциональность приложения(банкинг, соцсеть, игровые и т.д.), а также целевую аудиторию.

Учесть, по какому каналу приложение будет распространяться: через сторы или напрямую к пользователям.

Сценариев функционального тестирования очень много.

Например:

- проверка корректности работы полей
- проверка логики переходов по экранам
- проверка корректности работы всех кнопок
- проверка взаимодействия с соцсетями
- проверка поддержки транзакций через системы онлайн-оплат
- проверка корректности установки приложения и его работы на всех предусмотренных устройствах

Нагрузочное тестирование

Цель нагрузочного тестирования - проверить, корректно ли функционирует приложение при разном количестве записей и при переходе из Wi-Fi в мобильную сеть.

Найти участки приложения, которые могут тормозить его работу.

Убедиться, что приложение не съедает весь заряд батареи телефона.

Важность этого тестирования переоценить невозможно - если приложение не справится и начнет тормозить или вовсе вылетать.



Конфигурационное тестирование

Конфигурационное тестирование показывает, корректно ли работает мобильное приложение (а именно, его клиентская часть) на разных устройствах.

Обычно перед конфигурационным тестированием готовится матрица покрытия, куда заносят все нужные конфигурации. После конфигурации приоретизируют и проверяют в первую очередь важные варианты.

Потому как проверить функционирование и отображение на всех устройствах и при всех условиях не возможно - мы должны иметь представление, как найти оптимальный компромисс.

Для этого и нужны приоритеты конфигураций.

Приоритеты обусловливаются нуждами конечных пользователей.

Тестирование безопасности

Данные пользователей обязаны быть в безопасности.

Тестирование помогает понять, все ли мы сделали, чтобы защитить данные (и не только пользователей) от возможных угроз.

Происходит проверка приложения на устойчивость к различным угрозам безопасности: Dos-атакам, вирусам, воровству данных.

Процесс тестирования безопасности приложения очень не прост и многогранен.

Для различных целей используются разные методы.

Например:

- проверка защиты данных пользователя от сетевых атак
- проверка обязательной аутентификации при доступе к секретному контенту
 - проверка корректности работы всех кнопок
 - поиск и устранение неуправляемого кода
- проверка защиты бизнес-логики приложения и т.д.

Юзабилити тестирование

Юзабилити - это свойство интерфейса, которое либо помогает взаимодействию пользователей с приложением, либо усложняет его.

Тестирование помогает выяснить, как пользователи взаимодействуют с приложением.

Понять заранее впечатления пользователя довольно трудно.

Проверка юзабилити может проводиться различными методами.

Это большой пласт тестирования мобильных приложений, сайтов и сервисов.

Сценариев тестирования довольно много.

Например:

Экспертный анализ - используются накопленные знания о пользовательском опыте.

Тестирование с помощью веб-аналитики - можно понять, какой путь выбирают пользователи, но этот вид анализа не даст понимания, что именно мешает пользователям.

Тестирование с участием живых пользователей с помощью специальных сервисов.

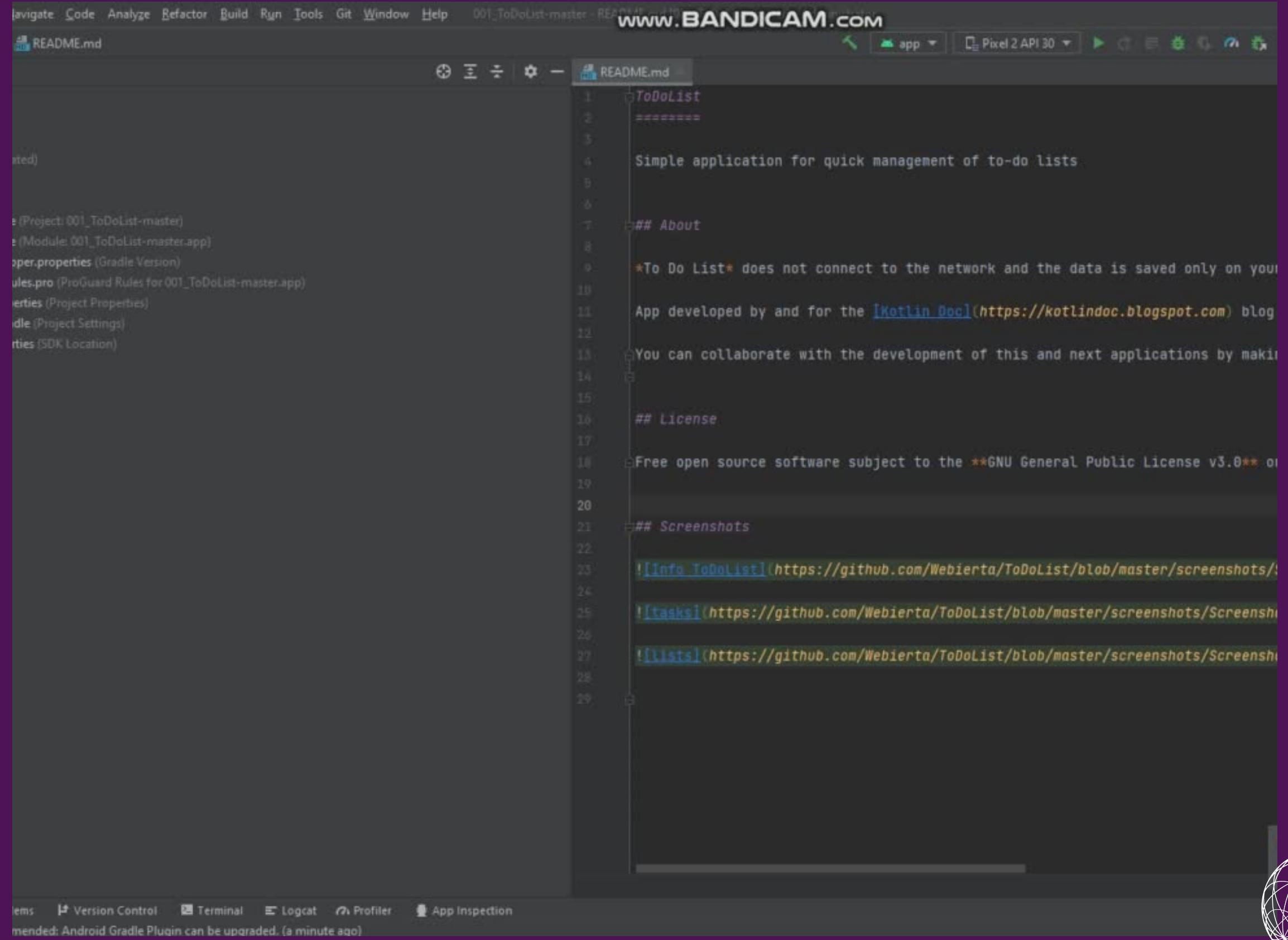
В итоге получается запись экрана с помощью которой уже становится понятнее, насколько удобен интерфейс для пользователя.

Симуляторы и эмуляторы

Симуляторы и эмуляторы отлично подходят для ранних стадий, но тестирование на реальных устройствах перед выпуском имеет решающее значение для обеспечения успеха приложения.



Эмуляторы Android и iOS



The screenshot shows the Android Studio interface with the README.md file open. The file contains the following text:

```
Simple application for quick management of to-do lists

## About

*To Do List* does not connect to the network and the data is saved only on your device.

App developed by and for the [Kotlin Doc](https://kotlindoc.blogspot.com) blog.

You can collaborate with the development of this and next applications by making a pull request.

## License

Free open source software subject to the **GNU General Public License v3.0** or later.

## Screenshots

![Info TodoList](https://github.com/Webierta/TodoList/blob/master/screenshots/Info%20TodoList.png)
![Tasks](https://github.com/Webierta/TodoList/blob/master/screenshots/Screenshots%20-%20Tasks.png)
![Lists](https://github.com/Webierta/TodoList/blob/master/screenshots/Screenshots%20-%20Lists.png)
```

At the bottom of the interface, there are tabs for Version Control, Terminal, Logcat, Profiler, and App Inspection.

Эмуляторы дублируют каждый аспект поведения исходного устройства, как аппаратный, так и программный.

Они в основном имитируют все оборудование, используемые реальными устройствами, позволяя точно так же приложениям работать на них без изменений, и все программные обеспечения.



Преимущества:

- имитирует как программное, так и аппаратное обеспечение;
- приложение работает без изменений;
- помогает выявить неожиданное поведение;
- относительно недорогое решение;
- часто используют при раннем тестировании во время разработки.

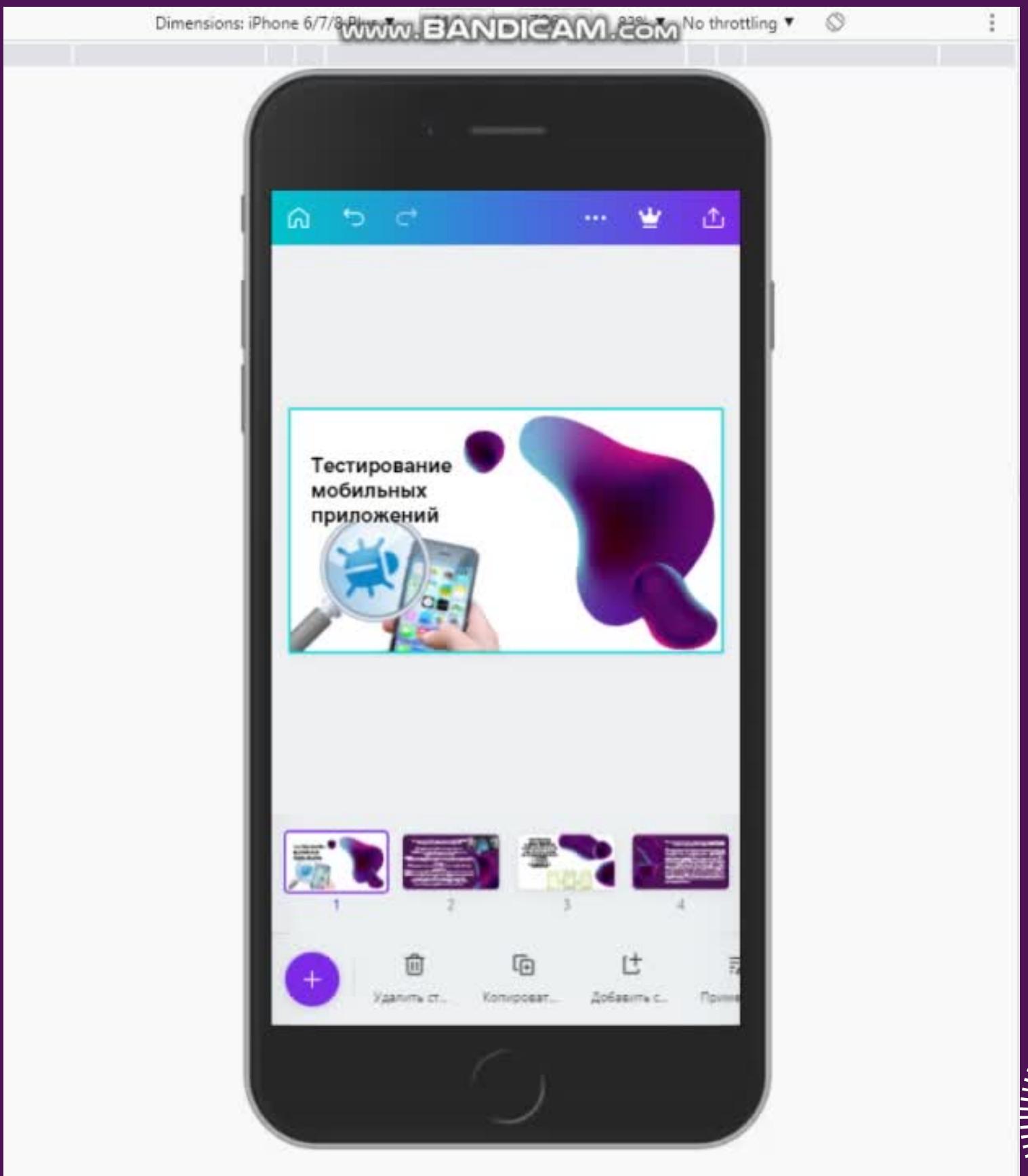


Недостатки:

- Эмуляторы работают очень медленно (из-за симуляции аппаратного и программного обеспечения);
- Эмуляторы не учитывают важные факторы - перегрев, разрядка аккумулятора или конфликт с другими приложениями;
- настроить хороший эмулятор - очень долго и дорого;
- Эмуляторы могут оказаться не совместимы с приложением, что порождает дополнительные действия для решения проблемы;
- Эмуляторы могут поддерживать только определенные версии ОС.



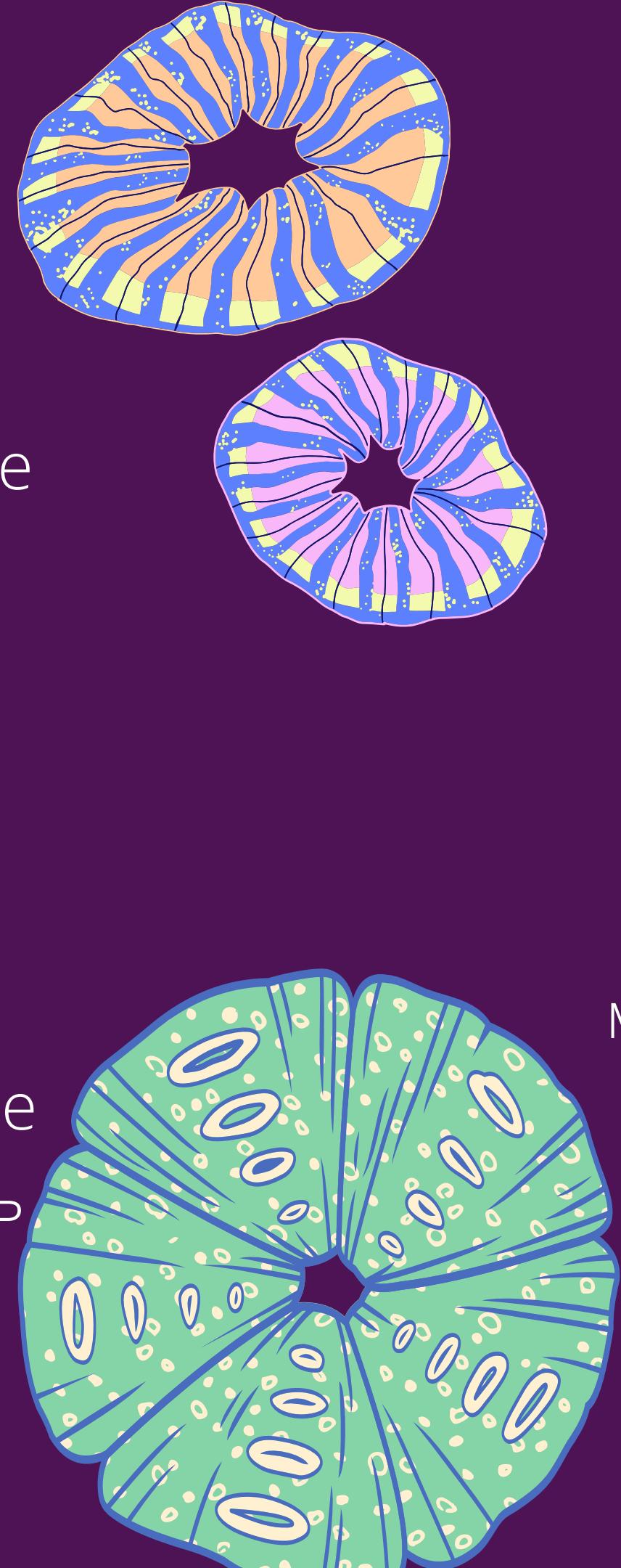
Симуляторы Android и iOS



Симулятор настраивает безопасную среду, аналогичную операционной системе оригинального реального устройства, но он не пытается симулировать аппаратное обеспечение реального устройства. В итоге, вы увидете ОС и интерфейс устройства, но вы не столкнетесь со всеми проблемами, которые может вызвать программное обеспечение. Некоторые приложения могут работать иначе, и это главная причина ненадежности симуляторов.

Преимущества:

- симуляторы быстрые (моделируют только программное обеспечение);
- относительно простые в настройке;
- может использоваться для изучения поведения приложения;
- поможет выявить неожиданное поведение, но также может дать ложные реакции.



Недостатки:

- не принимает во внимание аппаратное обеспечение;
- приложения могут работать немного по-другому;
- результаты моделирования могут быть трудными для анализа из-за неполных данных.

Charles Proxy

Charles Web Debugging Proxy - это инструмент мониторинга HTTP и HTTPS трафика.

Он выступает в роли прокси-сервера между тестируемым приложением и сервером на бэкенде, позволяя не только видеть, но также перехватывать и редактировать запросы.



Основные возможности:

- Breakpoint - это настройка точек остановки, которые позволяют перехватить запросы
- Map remote - подмена ответа сервера. Подмена ответа одного хоста на ответ другого
- Rewrite - перезапись данных, которые отправляются в Charles Proxy
 - Позволяет задать правила изменения и работать в автоматическом режиме.
 - Можно изменять и добавлять заголовки, искать и заменять текст в теле запроса или ответа.
 - Можно менять статус ответа.
- Throttling - помогает тестировать сервис на плохой связи, искусственно ограничив через настройки пропускную способность канала
- Repeat Advanced - позволяет нужное количество раз повторить тот же запрос (автоматически)

Подготовка и тестирование мобильных приложений

1. Планирование (основной и дополнительный функционал, с чем взаимодействует, набор девайсов)
2. Документация (чеклист, тест кейсы)
3. Подготовка и настройка тестового окружения (девайсы, установочные файлы, инструменты для тестирования)
4. Ручное тестирование:
 - тестирование процесса установки;
 - тестирование важности обновлений;
 - тестирование процесса регистрации и авторизации;
 - тестирование всех функций приложения;
 - тестирование функций, специфических для устройства;
 - тестирование отправки и получения сообщений об ошибках;

- низкоуровневое тестирование ресурсов: использование памяти, батарея, автоматическое освобождение ресурсов и т.д;
- тестирование сервисов: состояния приложения, прерывания, функционирование как в режиме онлайн, так и в автономном режиме, переключение между сетями;
 - тестирование верстки;
- тестирование процесса удаления и повторного установления.

5. Автоматизированное тестирование (API и UI тесты)

6. Подготовка документации о тестировании (заполненный чеклист, комментарии и т.д)



Ссылки:

<https://habr.com/ru/post/570784/>

<https://itvdn.com/ru/blog/article/mob-test-blog>

<https://qalight.ua/ru/baza-znaniy/testirovanie-mobilnyh-prilozhenij/>

<https://software-testing.ru/library/testing/mobile-testing>

http://testbase.ru/?post_type=skill&p=462

<https://www.performance-lab.ru/blog/mobilnoe-testirovaniye-fitness-device>

<https://qa-academy.by/qaacademy/news/testirovaniye-mobilnyx-prilozhenij-dlya-nachinayushhix/>