

Real device testing:

Web and Mobile Device Testing Best Practices



Introduction

There are over 14 billion mobile devices in operation worldwide. Companies seeking to take advantage of the global online consumer base must constantly optimize their offering for web and mobile devices.

Network testing on real devices ensures that mobile apps, mobile websites, web and desktop apps remain functional and dynamic across operating systems, network connections, browsers, devices, and a global user base.

Explore how to leverage real, in-the-wild web and mobile device testing for consistently flawless user experiences.

Device Testing Options

Software development companies have two major options for device testing — real and virtual. Each solution has its advantages and drawbacks depending on the application's lifecycle stage.

Virtual device testing solutions like simulators or emulators are great for early-stage testing, while real device testing is crucial to ensure an app's success before it gets into the hands of end-users.

Often, utilizing a combination of testing options produces the best end product.

For example, a large e-commerce company might have its employees test a feature on its in-house device lab, hire a networked testing company for skilled manual testing, and utilize a digital device farm during development sprints.



In-house mobile device testing lab

Some companies opt to build and maintain an in-house device library for employees to quickly test on a variety of devices.

While this strategy might suffice for a quick unit test of a new or a recently updated feature, maintenance of an in-house device lab is difficult. Acquiring devices and updating their operating systems isn't always the most cost-efficient or scalable solution. An internal QA team is normally focused on testing strategy, management, and automation and too lean to employ enough manual testers to cover all devices. Using internal device labs is becoming increasingly outdated due to their inability to handle localization tests, device coverage, and the drastic rise in remote work.



Emulators

Emulators have limited potential applications, and work best for lower priority devices that still need coverage, or during

the initial stages of testing. This technology allows QA engineers to test unique devices on their PCs by working on various device profiles. As part of a strategy, emulators can be an effective tool, but there are disadvantages to relying on device emulators as a primary testing option including:

- Extensive setup
- Slow test time
- Inaccurate screen resolutions and image rendering
- External conditions (like the effect of loud noise on sensors or slow network connections on task completion) can't be tested
- Internal hardware differences (CPUs and GPUs) aren't tested
- Touchscreen issues cannot be emulated



Simulators

Device simulators are often used for regression testing and maintaining quality in existing functionality.

Simulators can test the base quality and functionality of core features, but apps often run differently in a simulated environment. Simulators can't cover the full extent of the mobile environment because they have more processing power than mobile devices, and an inability to test external conditions like the device battery.



Device farms

With device farms, QA teams can test app performance on a variety of devices hosted in the cloud.

Although these digital device labs provide easy access, the simplified cleansing methods they use to present the devices and the centralized location of the tests don't match real-world environments. The device screen size and responsiveness are accounted for but the OS and other physical elements such as battery life or device storage are not.



In-the-wild devices

Optimal testing includes deploying and testing on real hardware

with hundreds of device/OS combinations in the hands of human testers. Validation is more accurate when it includes the device's responsiveness, how a real user interacts with that device, battery life, network connectivity issues, and location considerations.

Device Testing Options

	In-house Device Lab	Emulator / Simulator	Device Farm	In-the-wild devices
Cost	Upfront & ongoing	Generally free	Pay-by-use, subscription	Included with third party testing service
Advantage	Hands-on testing, real-time	Quick to switch between devices	Many devices available	Thousands of devices/OS available Powered by real people doing real testing
Disadvantage	Requires ongoing maintenance Limited access for remote employees	Location, UI via mouse, solid network Doesn't simulate hardware	Higher price point Location, UI via mouse, solid network Only the most popular devices supported	Individual devices not on a connected network
Best Use Case	Developer testing during sprints, real-time	Quick feedback for CI builds Developer checks	Parallelized test automation Specific devices for debugging	Function, regression, UX, location-based, localization, real-device testing Remote, burstable teams



Two real-life examples

Large enterprise, millions of monthly users, lots of automation	Emerging growth company, small team, no automation
In-house Device Lab • 2 devices per person, split between iOS & Android & data plans vs. wifi only • Drawer with older phones and wearables for testing	In-house Device Lab • QA person with 5-6 devices • Dev & QA devices spread across models/OS
Emulators/Simulators • Build verification tests, concentrating on functionality • Quick feedback	Emulators/Simulators • Developer quick checks
Device Farm • Extended automated tests concentrating on functionality • Functional coverage: 2 iOS versions, 5 Android versions	Device Farm • Available for specific tests (low-level subscription/ pay by use)
In-the-wild devices • 47 Device/OS permutations	In-the-wild devices • 12 Device/OS permutations

Why Test on Real, In-the-Wild Devices?

While emulators, simulators, and device farms have use cases that prove their effectiveness, they can't be the singular testing strategy. In-the-wild device testing is the only strategy that accounts for browsing, network connection, and unlimited device and operating system combinations experienced by real users.

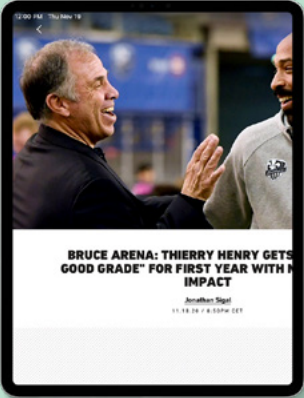
Some applications require extensive real device testing to ensure that the product is interfacing with sensors properly and the graphics fit each user's screen. Responsiveness can't be sacrificed during testing, and the multitude of consumer screen sizes require real device testing for proper functionality.

Example of device-specific bug

What it should look like on iPads



What it looks like on iPads



Portrait



Landscape



For software development teams building consumer-facing products, the ability to test apps on the same real devices used by their target users is critical for several reasons:

1. Network realities

As apps depend increasingly on mobile internet plans and wifi connection to access the network for data or third-party services, network glitches impact functionality.

2. Location-based testing

Apps that offer services to users based on their geographic location, using GPS data from cell towers or Wi-Fi, require testing in different locations for enhanced accuracy and stability.

3. Device realities

In most consumer devices, multiple applications are running on the same device and are consuming hardware resources, such as CPU, memory, network bandwidth, etc. In-the-wild device testing can also account for Bluetooth connectivity and camera features (AR, scanning, photography).

4. Fit and finish

Human touch plays a significant role in measuring how easy and user-friendly a software application is and whether it meets design expectations. Structured exploratory testing helps uncover issues missed from automated scripts while usability (UI/UX) testing revolves around the entire app-driven experience – something that requires real people testing on real devices and in real locations.

5. To find more bugs pre-release

The more devices you use for testing, the more likely you are to find a device-specific issue.



“Testing an application across multiple, in-the-wild, hardware and network configurations is crucial because in the real world, users report issues on real devices, not on simulators or emulators.”

Kristi Kaljurand,
Testlio Engagement Manager

5 Reasons to Test on Real Devices

Network Realities

- Wifi, data, 5G, 4G, etc.
- Bandwidth with connection

Location

- Location-based capabilities
- Movement
- Spoofing can be difficult

Device Realities

- Bluetooth connectivity
- Camera (AR, scanning, photography)
- Other apps running
- Resources like battery, memory and CPU

Fit & Finish

- Smooth & expected rotation
- Usability, Accessibility Gestures
- Pixel peeping

Find more bugs pre-release

- “human in-the-wild” devices reveal more device-specific bugs

Which Real Devices/OS Combos to Use?

Testing and QA teams must be strategic when choosing which device and device/OS version combinations to cover when catering to a target user base, keep an eye on potential growth, and balance risk tolerance with deadlines.

Based on a comprehensive analysis of client release data and device-OS configuration tests, Testlio found that leading mobile app companies in most industry segments follow a best practice of testing 24 or more unique device/OS combinations before each update.

24

device/OS combinations

Average number of device/OS combos tested based on aggregate client data

41

device/OS combinations

Average number of devices tested per week for a top-tier shopping app

A Testlio client, a top-tier shopping app that shares the highest App Store rating (4.9) with only three other Top 30 shopping apps, uses 41 device/OS combinations—almost twice the median number because they have a large user base and greater than average risk.

→ **Check out the full report: State of App Testing – Commerce and Retail.**

3 ways to determine your real device needs

User analytics

When strategizing device coverage for an existing application, nothing is more important than usage analytics, including:

- Operating systems: do users typically upgrade to the newest OS version, or do you need to support past versions? If so, how far back?
- Devices and brands: which devices are commonly used? Are users on the most current versions? If there are multiple versions of one device, how similar or different are those versions, and can they be combined?
- Phone vs. tablet: what is the breakdown of use? How much of the device coverage should be allocated to phones vs. tablets?

User profile

User profiles can provide focus on target consumers and what types of device they use.

For instance, corporate finance/expense approval apps have a user base that skews towards latest and greatest devices, while users of a family-oriented app are likelier to use older devices and OS.

Analyzing users' geographical distribution can give insight into product growth and the type of user who might be adopting it in the near future.

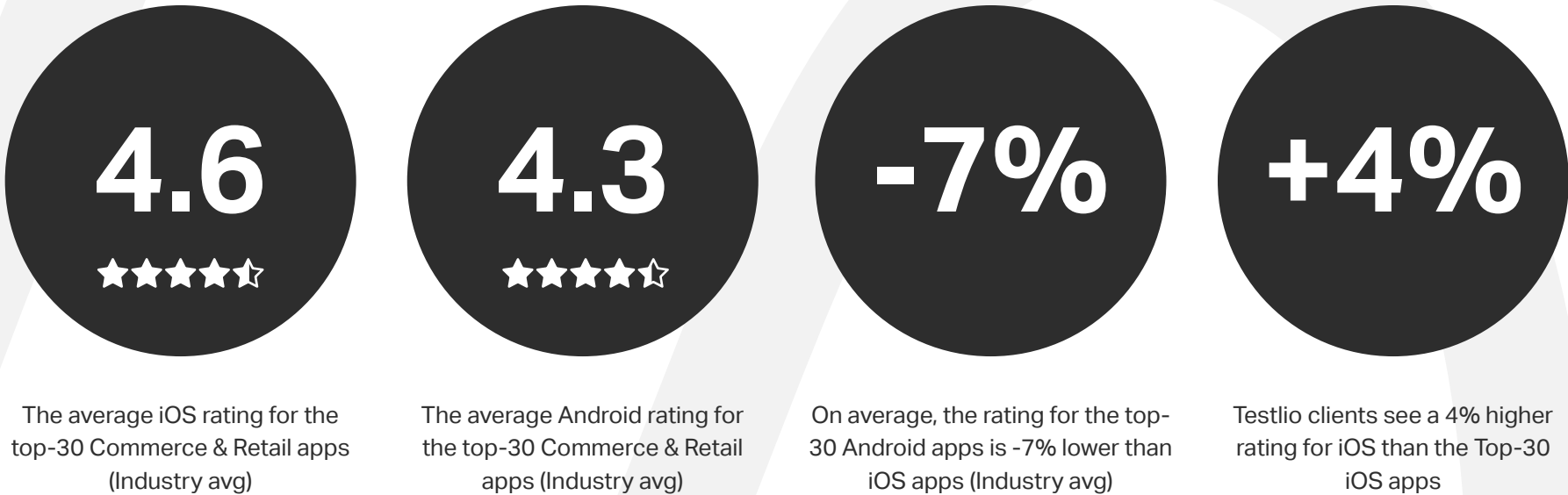
Technology innovation

When testing on brand new devices/ OS versions, focus on getting a good breadth of coverage. iPads and iPhones come in various sizes and models, but they have a sole distributor. With Android, each manufacturer tweaks the OS slightly to fit with different offerings that differentiate their brands, such as auditory or motion sensors or a curved screen.

Because Apple is pretty assertive with upgrading its OS, the latest two versions are generally sufficient to provide test coverage. For the latest versions of Android, the Google Pixel series is generally a good bet as other Android phones tend to stick with the operating system in use at the time they were built.



Testlio’s research shows that iOS apps have higher ratings than Android apps on average in the Commerce & Retail space.



While testing methods like emulators and in-house device labs have niche use cases, in-the-wild device testing provides a holistic approach for companies wanting to build a better app, update, or site. Look for these important features in any real device testing solution:

- Vetted testers
- Available in the global markets you need
- Ability to shrink and expand testing resources on-demand
- Can cover popular devices from the past four years
- Test cycle management
- Collaboration between freelancers to reduce duplicate issues
- Ability to utilize in-house QA teams alongside experienced testers in one platform

Learn **more** about Web and Mobile Device testing

[Learn More](#)

About Testlio

Testlio is the originator of networked testing. Our award-winning combination of platform, freelancers, and services enables you to deliver great customer experiences—fast. In any location. On any device. In any language. Testlio

clients include athenahealth, CBS, Equinox, Microsoft, the NBA, and Netflix. Collectively, they have awarded us an industry-leading 4.7 G2 rating and an NPS of 75.

