

The Essential Guide to Mobile App Testing

Tips, techniques & trends for developing,
testing and launching mobile applications
that delight your users

The Essential Guide to Mobile App Testing

Introduction	3
The Mobile Boom	4
What's at Stake	5
Why Failing Hurts	6
The Mobile Boom: Facts and Figures	7
Why It's Tough: The Soaring Expectations of Users	8
Why Failing Hurts	9
Common Issues in Mobile App Quality	11
The Increasingly Complex Testing Matrix	20
Device Fragmentation	21
OS Fragmentation	22
The Mobile Web Invasion	24
The Rise of Mobile Web	25
Testing Considerations and Differences	26
Achieving Testing Coverage	31
Conquering the Testing Matrix	32
Testing Options	33
Emulators & Simulators	34
From Surviving to Thriving: Advice for QA Leaders	35
5 Things You Ought to' Know	36
For QA & Test Leaders	37
For Developers & Engineering Leaders	38
For Product Managers & Marketing Leaders	39
For Executives & Biz Leaders	40
Tips, Tricks and Other Resources	41
Conclusion	46

Introduction: You've Developed a Mobile App...Now What?

By the time you finish reading this, the mobile app landscape will have changed. New OS versions will have been released. A bunch of new devices will have hit the market. Cool new apps will have gone viral, altered the way we interact with the world around us, and sold for millions. And mobile app testing will have become that much more complex and challenging for all of us.

Discouraged? Don't be. Despite the ever-changing mobile landscape, there *are* some constants in the realm of mobile app quality – and many of them can be found in the following pages.

Designed to help orgs of all sizes – from ten-person startups to dev agencies to in-house teams at global brands – this eBook will dispel some common myths about mobile app testing and outline several proven strategies to help you deal with mobile's exhaustive test matrix. Additional topics include:

- **Most Common Testing Challenges:** What are the most frequent hang-ups in terms of functionality, design, security and localization? We'll share them and give tips on how to avoid these pitfalls throughout the dev process.
- **Mobile Web vs. Native Apps:** They're both mobile, but the testing challenges of native apps are entirely different from those of mobile web. We'll explain which route is easier from a testing perspective.
- **Testing Coverage:** Before you can achieve thorough testing coverage, you'll need to know what the testing matrix actually looks like. We'll cover the basics of OS, device maker, carrier and other areas.
- **Tips and Tricks:** Get some helpful advice on obtaining quality crash reports, using screenshot and video capture tools, as well as other tips and tricks of mobile app testing.

Ready to learn more? Let's get started...



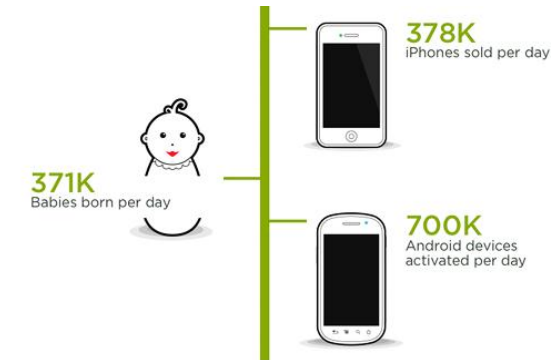
INTRODUCTION: THE MOBILE APP BOOM



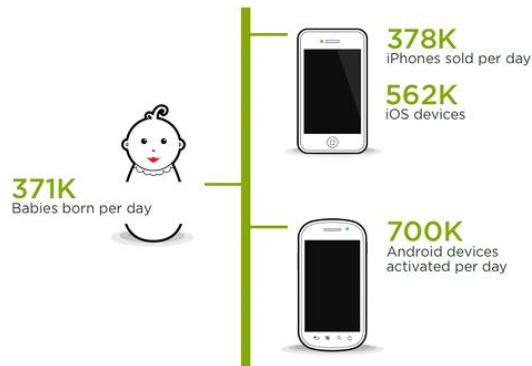
What's at Stake?

By now, the mobile boom should be obvious to just about everyone (below are some hard numbers for those who need a bit more convincing). What this means is that smartphones are rapidly becoming the primary method of interaction for consumers and businesses worldwide. So what's at stake? Just the future of your business...

377,900 iPhones are sold every day around the world

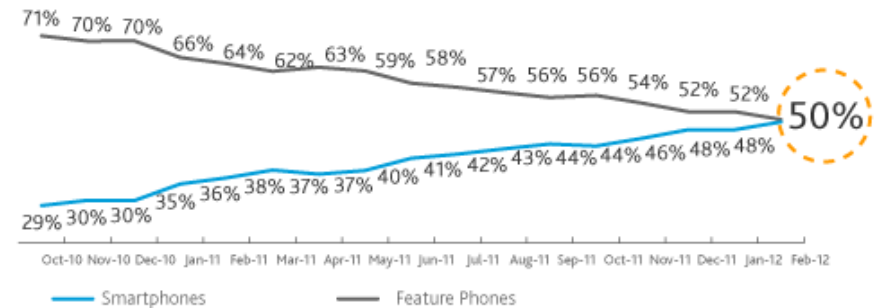


Every day 700,000 Android devices are activated across the World.



Looking at the total of iOS devices (iPhones plus iPads and iPod Touches) sold per day brings the total of Apple mobile devices sold per day to 562,000. Together with Android devices that's 1.27M mobile devices sold or activated per day compared to 371,124 children born.

U.S. Smartphone Penetration



Source: Nielsen

nielsen

Fun Fact: By 2015, International Data Corporation (IDC) predicts that 182.7 billion mobile apps will be downloaded. That's a 1600% increase from the 10.7 billion apps downloaded in 2010. ([source](#))

The Mobile Boom

The explosion of consumer apps can be seen in just about every industry, but here are a few of the more notable ones:

- **Retail:** Location-intelligent mobile commerce
- **Media:** Magazines & newspapers going 100% digital
- **Travel:** Mobile bookings, check-ins, maps, deals, etc.
- **Education:** Tablets and apps in classrooms
- **Healthcare:** Patient records, physician notes, etc.
- **Finance:** Apps for real-time trading, portfolio analysis
- **Social:** Games and social media platforms

And mobile apps are running rampant in the B2B world as well, including spaces such as:

- **Business Apps:** CRM, ERP, HR systems
- **Productivity:** Docs, spreadsheets, presentations
- **Collaboration:** Email, IM, publishing

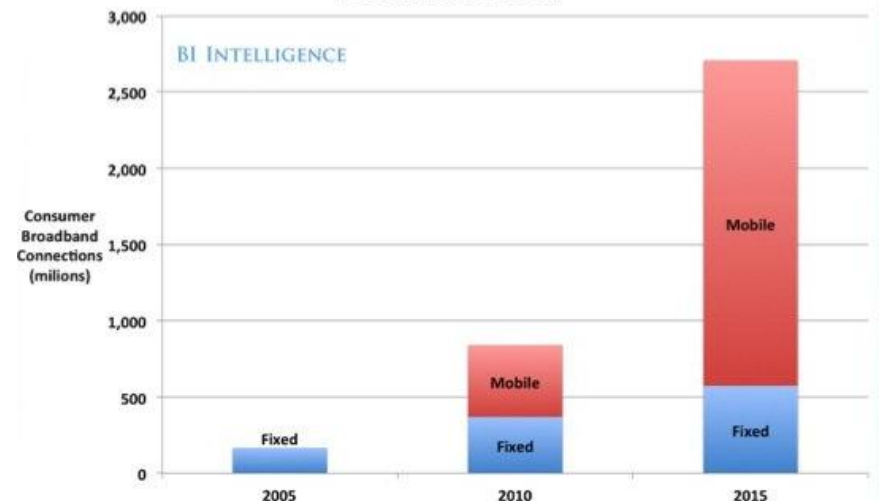
Still think that mobile apps are for games, music and “entertainment purposes” only? Do you *really* think that mobile applications won’t be relevant to your industry, your company, your team and your job a year from now? Think again.

As mobile adoption further increases, mobile apps will become ever more critical to the success of your business (and *your* career). Companies that recognize this trend for what it is – a technological revolution – and take preemptive action to make the quality, security and usability of their mobile applications top priority, will find themselves with a tremendous advantage.

Global Internet Device Sales



G-20 Internet Access

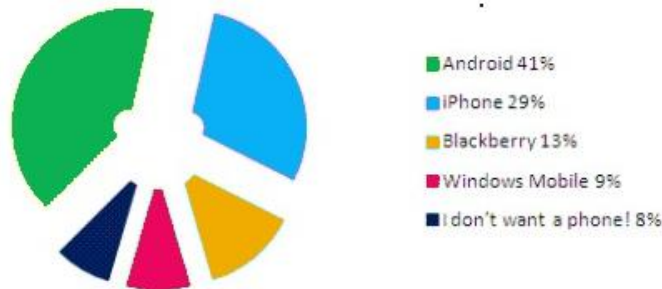


The Mobile Boom: Facts and Figures

The readers of the [SD Times](#) are a fairly representative subset of tech-savvy firms to poll when you're interested in how many companies are buying into the mobile app phenomenon. Their recent survey found that **70% of respondents planned on having a mobile app within the next 18 months**. Other findings included:

- Of those already building apps, tablet specific apps are beating out phone-centric apps 88.1% to 82.3%
- Apps for e-readers are grabbing the attention of 9.8% of companies
- Android apps barely beat out iOS at a rate of 63.9% versus 63.6%
- 1 in 5 companies are developing for Blackberry

What phone OS do you WISH your company supplied you with?



“The biggest motivator for the surveyed audience was to offer mobile access to hosted online services. This was the business driver for 38.4% of respondents, followed by making money by selling apps directly to consumers (36.3%), being mobile to stay competitive (33.2%) and offering mobile versions of desktop software (32.9%). Only 8.7% said that making money from embedded mobile ads was a business driver.” ([source](#))



Mobile First: “From a consumer perspective, the top mobile-first companies are all app-centric. These companies provide services that could be done from a PC but provide the service specifically for mobile platforms. Right now, **the top tier of mobile-first companies are the ones that understand that the mobile experience is fundamentally different from a PC experience** – and then develop and design their services specifically with mobile interactions in mind.” ([source](#))

Why It's Tough: The Soaring Expectations of Users

According to a recent study, 60% of mobile users will abandon your app or site if it doesn't load within three seconds. Three seconds! Of those users, 43% said they would ever return to your app or site – and that's just scratching the surface of user expectations.

Other areas include:

- **OS & device fragmentation:** Having an app available on some devices and not others is a death sentence in the mobile world. If you're aiming for widespread adoption, then you need to provide a seamless user experience regardless of the device. More on this momentarily.
- **Security:** Companies who inadvertently violate their own privacy policies run the risk of making headlines for all the wrong reasons. Users expect their personal data to remain personal – and they expect their apps to be 100% free of viruses, malware and other threats.
- **Connection speed:** Does your app run the same on 3G, 4G and Wi-Fi coverage? If the answer is no, expect low app store ratings. In some instances, such as installation, a poor connection can make the difference between a five star review and one star review.
- **Data usage:** Users are wising up to the fact that some apps drain battery faster than others – and they're abandoning those apps in favor of the more economical ones. When developing an app, stay mindful of how it functions in all of these situations.
- **Installation & launch:** Is your application easy to install and launch? If it is, you're ahead of the majority of app developers. Users have shown little patience with regard to apps that are not easy to install and run.

The point is this: Right or wrong, users have incredibly high expectations when it comes to mobile app quality. As if testing wasn't challenging enough already!



Why Failing Hurts

Back in the day, one bad customer experience – whether it be with a business, service or product – would spread via word-of-mouth and reach a few prospective customers. Thanks to **social media** platforms like Twitter, Facebook, Yelp and others, however, that same word-of-mouth now travels at the speed of light and reaches hundreds, thousands, or even more users.

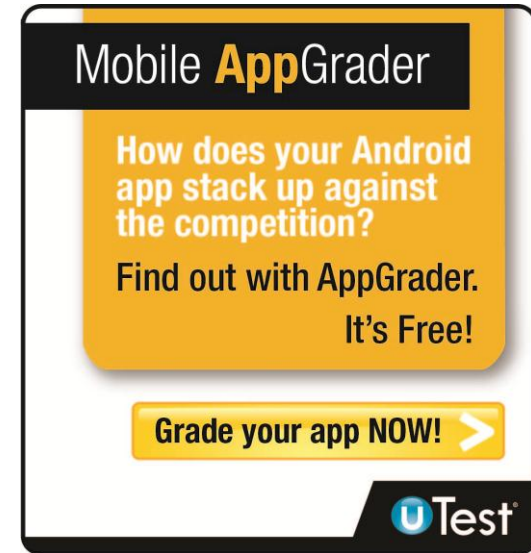
The other major difference is **app store ratings**. Imagine if a poor review were stamped directly on the packing of a product. With app store ratings, that's exactly what's happening.

In the world of apps, the meaning of “quality” can often be ambiguous. This is not the case with *USA TODAY*, which takes a very quantitative approach to the term. You would be wise to follow their example.

*“App store ratings are extremely important to us,” said Michael Croghan, Mobile Solutions Architect. “You can launch a beautifully designed native application, but if it crashes, then it will receive a poor rating and users will go elsewhere. Our goal is to launch nothing short of a 4.5 star app every time - no exceptions. Anyone can read your app store rating. **There’s no way to hide poor quality in the world of mobile.**”*

That last line is worth repeating: “There’s no way to hide poor mobile app quality in the era of social.”

And that’s why failing hurts.



Key Questions

Before addressing the challenges of mobile app testing, it would be smart to first understand the key questions. Here are a few questions you've likely asked yourself and your company:

- How's a company supposed to ensure their multiple mobile apps work on multiple OS platforms, carriers, devices and across countless locations?
- What are some of the most common bug issues?
- Should testing rely on manual testers, automated tools or a combination of both approaches?
- Should testing be performed in-house, by an outsourcing firm or by a loosely affiliated beta testing group?
- What are the testing challenges of native apps vs. the mobile web?
- Is mobile fragmentation actually a good thing?
- Can a company actually increase its app store rating?
- Is extensive testing necessary even in a closed ecosystem like iOS?
- Are some types of testing, like functional and security, more critical than other types of testing, like localization and usability?

Now let's get to some answers...



The Mobile Mantra

"Much as we love freedom and choice, we also love things that just work, reliably and seamlessly. And if we have to pay for what we love, well, that increasingly seems OK. Have you looked at your cell phone or cable bill lately?"

- Chris Ander

COMMON ISSUES IN MOBILE APP QUALITY



Common Functional Issues for Mobile

Functional testing covers the broadest range of criteria, so any checklist will be somewhat incomplete. That said, here are a few areas that should probably always be covered in your [functional testing](#) strategy:

- **Sign-up & Login:** This may seem like a no-brainer, but if users cannot easily access your application, your efforts will have been wasted. If your app or mobile site requires password and username (not recommended), pay close attention to the fields and make sure that it's easy for users to enter their information.
- **Menu Options:** Often times, menu options can be difficult to access and decipher. Make sure that menu items like Help, About, etc. are easy to find and navigate.
- **Keys:** Any problems related to scrolling, text selection, the back button, etc. are bound to lead to trouble, so make sure your key functionality is clear and consistent. Also, be sure to cross-check if your app will run with a keyboard and/or touch-screen.
- **Data Handling:** An important consideration for all mobile apps (but for games in particular) is data handling. Does your mobile app store saved information properly? What about data deletion – or worse – unintended data deletion? Make sure to include this in your testing scope.
- **Connection Speed / Carrier:** With the iPhone now running on Verizon, Sprint and others, testing on only one carrier is no longer an option. Make sure that your most basic app tests include some of the most popular (and likely) carriers, as performance can vary greatly from one carrier to another.



- **Operating System:** Like carriers, the mobile operating system running your app will have a huge impact on its relative performance. For some operating systems (like Android), you will not be able to test across every possible combination with in-house resources, but you should identify 3-5 of the most popular ones and test accordingly.
- **Screen Size:** While this applies mainly to tablets, screen size discrepancies should be a top consideration while testing your mobile app. For instance, how much should users have to scroll from left-to-right or top-to-bottom, if at all?
- **Interruptions:** How does your mobile app behave when the device battery is at full strength, medium strength and low strength? This is just one of the many variables to consider when testing, but it's also one of the easiest items to check for. Others include incoming calls, SMS, MMS and video calls.
- **Error Messages:** Your error messages should be clear, concise and actionable. Do this, and you're a step ahead of virtually all mobile applications on the market today.



From The Experts: Functional Testing Tips

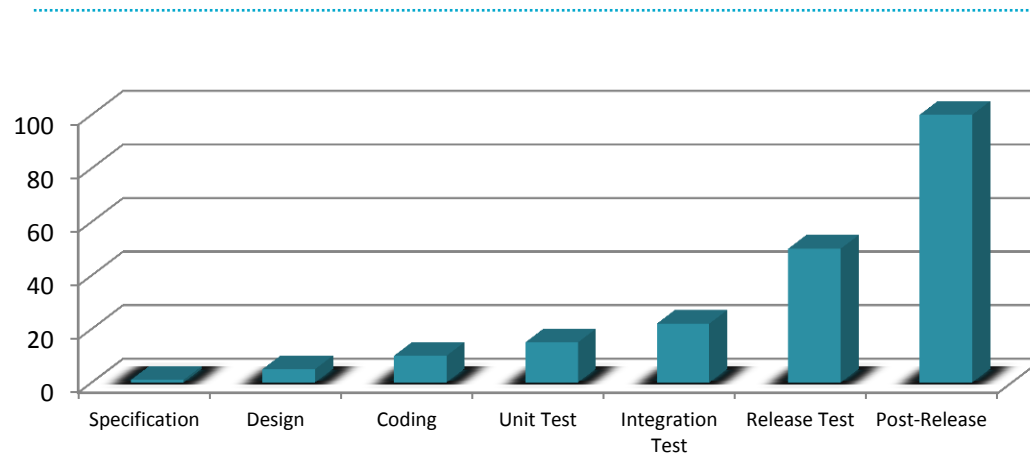
- **Screen real estate:** You might use a mobile device just like a regular 1024×768 pixel application, but your users probably won't. Try to actually use the application on a number of devices — just use it. You'll likely come away suggesting a mobile interface, perhaps an automatic re-direct on login when your application senses a mobile device. Even then, you'll want to explore the application in a number of devices, looking for usability problems. (*Matt Heusser*)
- **Finding device specific bugs:** If, for example, you discover links that are not functional on certain devices, then you should first determine whether it works on the desktop browser. If it is reproducible on the desktop browser, then it is obviously not a device-specific bug, but rather a bug in the basic application. If a link is functional in the desktop browser and it doesn't work in the device, then it should be identified as a device specific bug. (*Jigar Patel*)
- **Normal use test:** Start on a full battery and use the application for 6-12 hours and measure the battery level at the end of each ½ or 1 hour. You may use an automated testing tool to do this so as to keep the test running for the required time interval. This test will tell you how quickly your application drains the battery when in 'normal' use, with all the foreground and background features of the application running normally. (*Sachin Date*)
- **Idle run test:** Turn off the screen lock and power saver modes on the device. Then start on a full battery and keep the application running on its main, home or dashboard screen as appropriate, and measure the battery level at ½ or 1 hour intervals. This test will measure the battery drain due to such things as intentional or unintentional automatic screen refreshes, and due to the background threads or services running in your application. (*Sachin Date*)



Functional Issues to Consider in the Dev Phase

With all the time we spend on tools and techniques designed to catch mobile bugs post-dev, we should point out that many of these defects can be prevented by following some key steps *pre-dev*. Here are a few points to consider, courtesy of 53days.com:

- Context is critical for mobile applications. A thorough understanding of the user's context and objectives is a must requirement.
- Conduct ongoing user acceptance testing (UAT) throughout the design and development process, including testing on real devices.
- Use simple navigation structures pointing to one specific task at a time.
- Store data on the device selectively and archive data that is less frequently accessed on the server.
- Effectively use multi-threading wherever possible to improve performance.
- Use high contrast text color and select typefaces for maximum readability.
- Provide clear status and feedback based on progress of task completion.
- Manage content wisely. Wherever possible, crop large images and reduce the size of data files.
- Use a consistent User Interface design that helps users maintain a familiarity with the application.
- Get involved in real-time interaction via social media. This could include live Facebook or Twitter streams.



Relative cost of defect, by time of discovery



Common Mobile Usability Issues

The goal of [usability testing](#), simply put, is to make sure that a user can complete the tasks they are expected to complete. More importantly, they must be able to do so easily and without becoming frustrated. A good user experience can be thwarted by a number of issues, including:

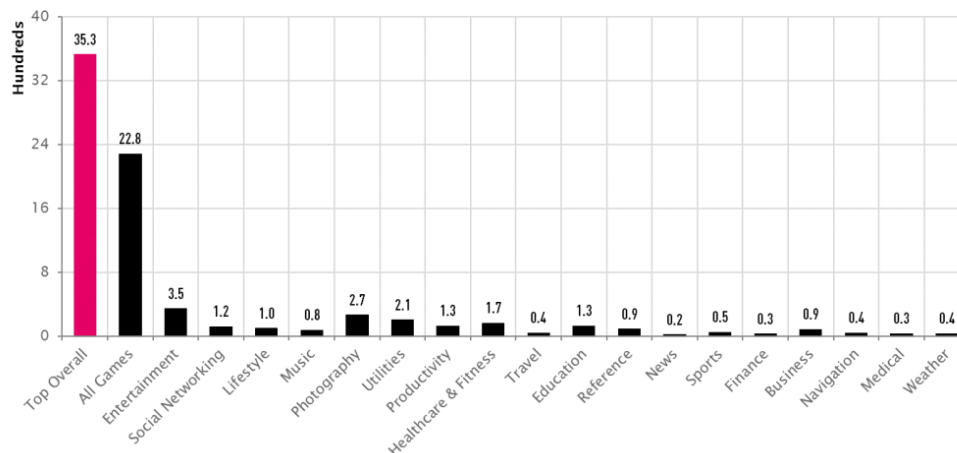
- **Functionality:** A mobile application must present the user with the appropriate functionality. If the functionality of an application is perceived as incomplete or inadequate, customers may be lost.
- **Layout & Design:** Good layout and design allow a user to easily complete tasks. If a button is placed in the perceived 'wrong' place, users will get frustrated and might look for an alternative product.
- **Interaction:** The flow of an application must be natural and allow the user to easily complete tasks. If a user believes they have been sent to the wrong page and have to manually backtrack in order to find the page they were looking for, they can become frustrated and abandon the application.



NUMBER OF PAID DOWNLOADS TO HIT TOP 25 PER CATEGORY

APPLE APP STORE FOR IPHONE, UNITED STATES, DAILY DOWNLOADS

DISTIMO



Reminder: To get to the top of the App Store, you'll either need to launch an amazing game (see sidebar), or be among the best in your field in terms of usability.



Usability Testing: Tablet Apps

It's quite common to see companies taking their current mobile app, increasing the graphic fidelity, and releasing it as an "original" tablet app. This is a big mistake – but don't take our word for it. Here's usability guru [Jakob Nielsen](#) on the UI problems of tablet apps:

*"The biggest problem in our recent tablet studies has been TMN: **too much navigation**. Also, too many inconsistently scrolling fields. Some tablet apps cram in so many weird features that **users get overwhelmed and flail around** without gaining mastery of the content.*

I think that most designers of phone-based apps have recognized the need to limit the number of features and the number of wildly scrolling areas. The small screen imposes useful discipline that keeps out the worst excesses that still dilute usability on larger tablets.

*While tablet UIs need to quiet down and **become more consistent**, that doesn't mean that they should be phone designs with prettier graphics. The bigger screen allows for more features, and more focus on immersive use over longer periods of time than the quick hits that are most useful on phone-sized devices."*



In-The-Wild Testing
Think outside the lab

**Is your app ready for
what lies beyond the firewall?
Find out with in-the-wild testing.**

Get the paper here >

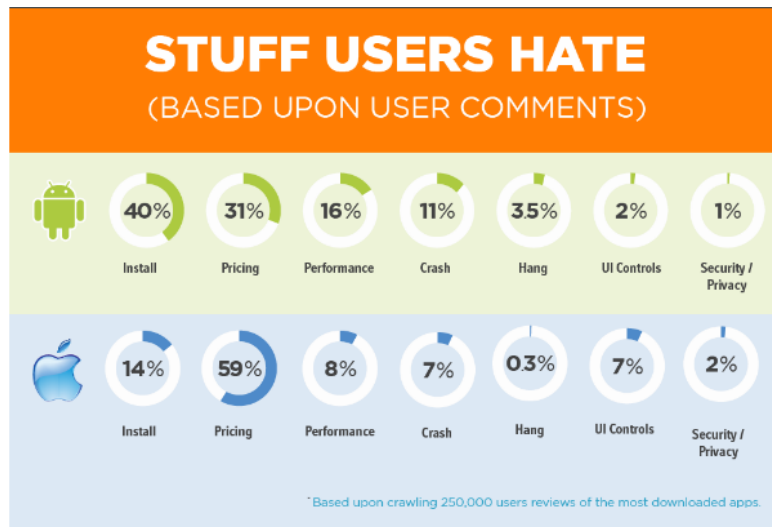
uTest



Common Mobile Load & Performance Issues

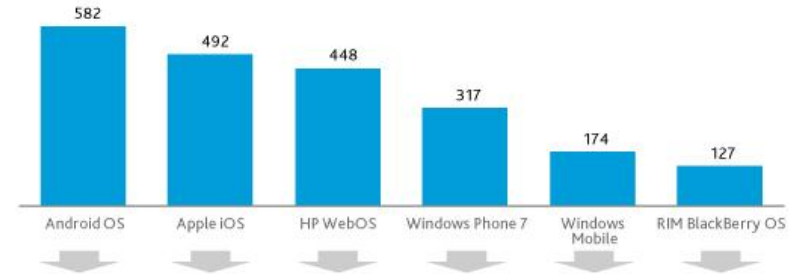
Load testing is another critical step in the successful launch of a mobile application. The key factor to be aware of here is mobile carriers and data usage, which can effect the speed and ease-of-use of the application. If the application performance is poor for the end user, they will be likely to divert to other apps and services. The application must therefore be tested with different devices and carriers depending on the county and region.

Performance issues, crashes and hangs accounted for roughly 15% to 30% of user complaints in the iOS and Android marketplaces.



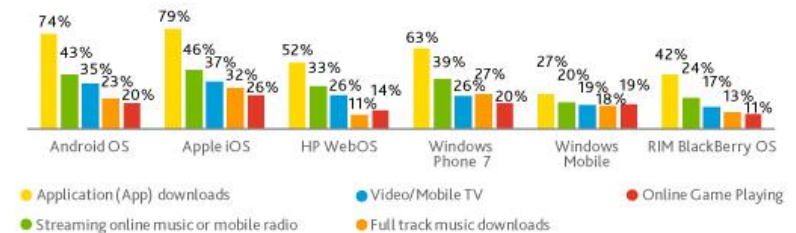
Average Monthly Data Usage (MBs)

Q1 '11 Nielsen Customer Value Metrics, National



Heavy Data Use Activities, Last 30 Days

Q1 Feb - Apr 2011, Mobile Insights, National



Source: Nielsen

nielsen



Common Mobile Security Issues

The explosion of mobile applications presents an entirely new set of security challenges. While most of the tools and practices of traditional web and desktop applications are equally applicable to mobile, there are some unique concerns to keep in mind, including lost or stolen devices, mobile malware, targeted attacks on devices and more.

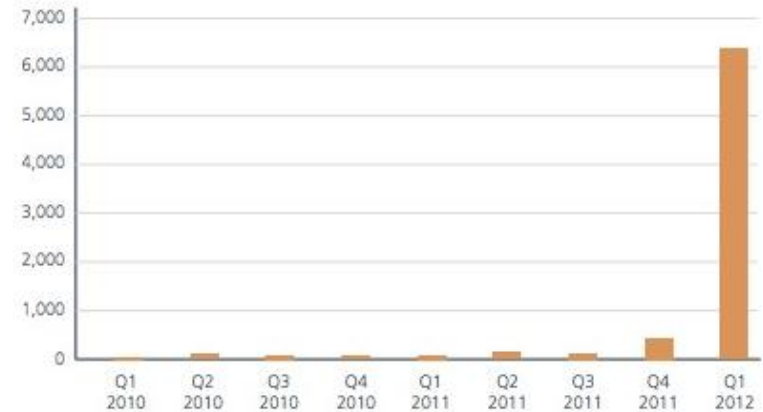
No application can ever be 100% secure from threats, but as developers, these are questions that you should always be asking yourself when it comes to [security testing](#) for mobile apps:

- **Confidentiality:** Does your app keep your private data private?
- **Integrity:** Can the data from your app be trusted and verified?
- **Authentication:** Does your app verify you are who you say you are?
- **Authorization:** Does your application properly limit user privileges?
- **Availability:** Can an attacker take the app offline?
- **Non-Repudiation:** Does your app keep records of events?

“Software can be correct without being secure. Indeed, software can meet every requirement and perform every specified action flawlessly yet still be exploited by a malicious user. This is because security bugs are different from traditional bugs. In order to locate security bugs, testers have to think differently too.”

– James Whittaker, Microsoft

New Mobile Malware



(Source: McAfee)

[Sidebar]: “These threats, as usual, mostly target Android. While Google and other major stores have made great strides in keeping malware out of their stores, third-party stores and forums remain a problem. Among the areas of mobile malware that saw major increases in threats was mobile backdoor malware and the always popular premium-rate SMS-sending malware.” [\(source\)](#)



Common Mobile Localization Issues

The worldwide proliferation of mobile applications (and the mobile web) requires your products to be accessible and contextual for users in all markets and regions. But with the localization process exceedingly difficult to verify, many companies are excluding entire sub-sets of users *without even knowing it*. Here are a few of the more common issues related to [localization testing](#) mobile applications:

- **Content** – Static & dynamic content like catalogs, search results, metadata
- **Dates** – Is the date January 1 or 1 January?
- **Characters** – Different languages have different set of characters
- **Postal codes** – In some countries postal codes contain letters
- **Phone numbers** – Different formats for different markets
- **Direction** – Some languages are written left to right, others are right to left
- **Currency conversion** – Especially important for internet retailers
- **Tax calculation** – VAT, sales tax and others vary from country to country



Why is Localization So Important?

“Our global customers have different demands of our products. We want products to ‘feel local’ and to support features that may be unique to specific markets. For instance, in India-based languages using a standard keyboard is difficult, so we develop strategies like virtual keyboards or category browsing for search. As we specialize our products for certain markets, it introduces more challenges for testing (e.g. requiring special cultural knowledge).”

– Patrick Copeland
Sr. Engineering Director, Google

Why is Localization So Difficult?

“The main reason is the timing. QA is always the last line of defense and often does not get enough time to do the job properly (because other teams slip the schedule). Well, in the case of localization it’s even more complicated because the translators must do their part before it can be tested. Hence, more wait time.”

–Galina Kramer
QA Manager of Localization, Zynga



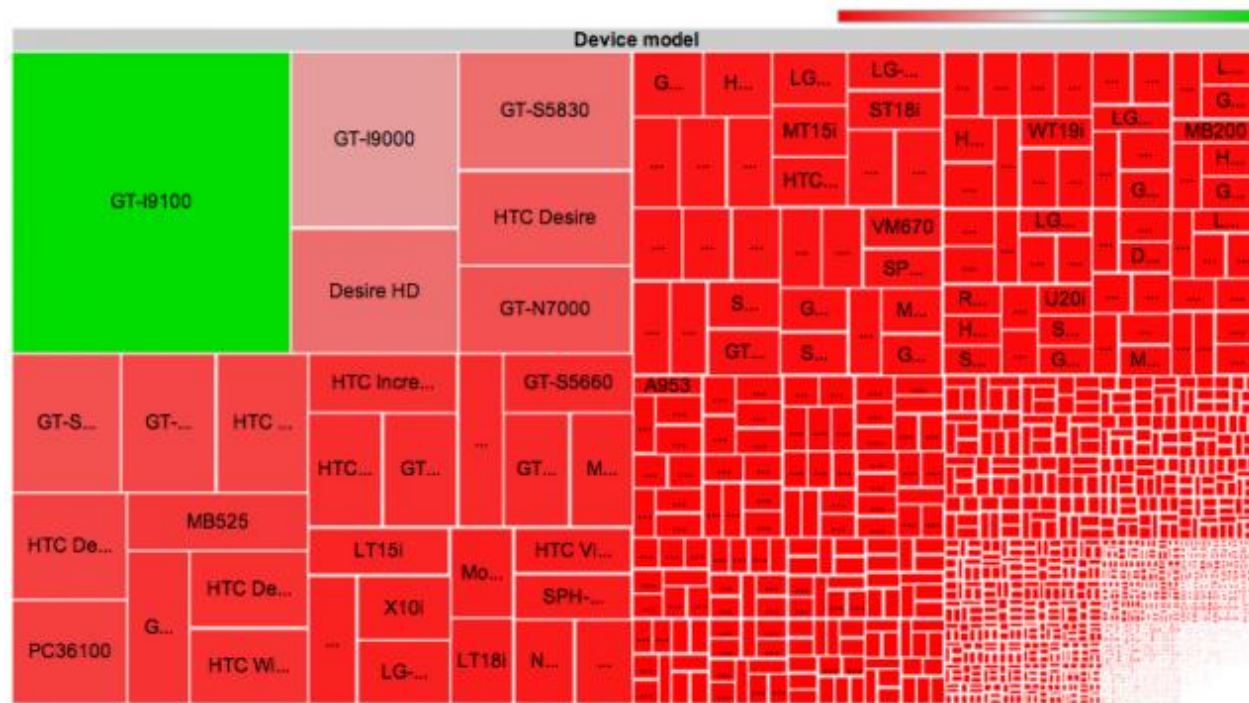
THE INCREASINGLY COMPLEX TESTING MATRIX



Device $\mathbf{F} \mid \mathbf{R} \mid \mathbf{A} \mid \mathbf{G} \mid \mathbf{M} \mid \mathbf{E} \mid \mathbf{N} \mid \mathbf{T} \mid \mathbf{A} \mid \mathcal{T} \mid \mathbf{I} \mid \mathbf{O} \mid \mathcal{N}$

Perhaps the most difficult aspect of the testing matrix is device fragmentation. This is especially true of the Android operating system, with its seemingly countless permutations.

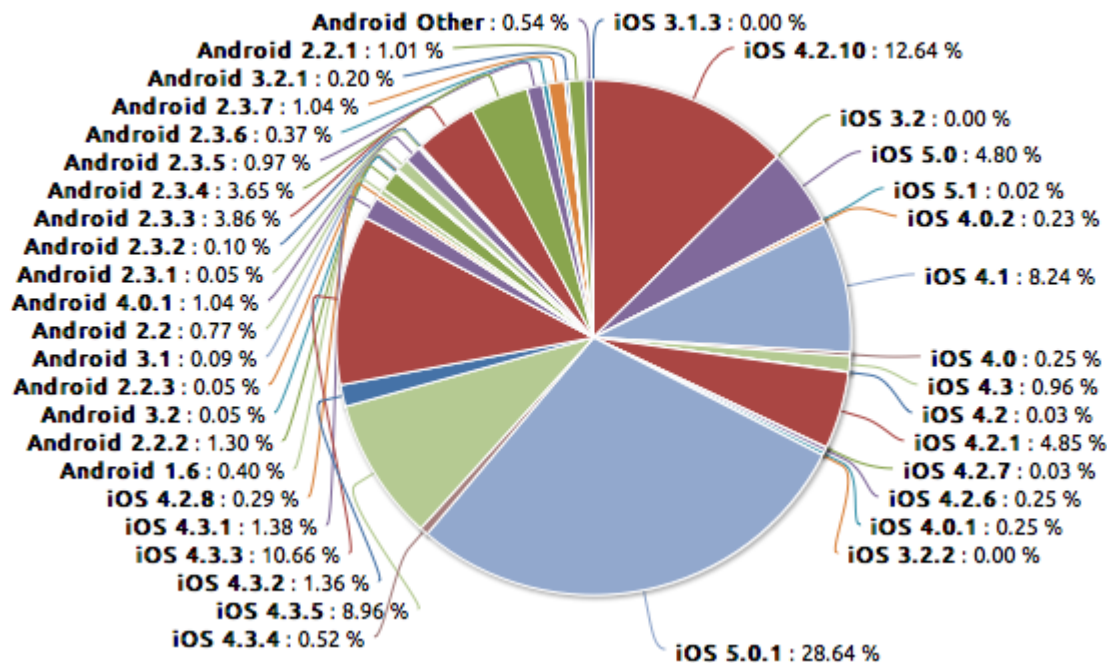
Take, for example, the image below. This is a data chart of nearly four thousand separate Android device models encountered by the dev team at [OpenSignalMaps](#) over the course of a six month period. If you're concerned with quality on cross-platform apps, chances are your matrix will look somewhat similar.



Operating System F | R | A | G | M | E | N | T | A | T | I | O | N

Although less daunting than the hardware matrix, the variety of mobile operating systems also poses a challenge for dev and engineering teams whose goal is to provide a consistent user experience across platforms. The pie chart below illustrates the fragmentation that exists in just the Android and iOS ecosystems. The chart outlines each operating system's required programming language.

Crashes by OS Version Normalized (12/1 - 12/15)



Platform	Language(s)	Remarks
Android	Java, C, C++	Open Source OS (based on Linux) developer.android.com
bada	C, C++	Samsung's mobile platform running on Linux or RealTime OS developer.bada.com
BlackBerry	Java	J2ME compatible, extensions enable tighter integration na.blackberry.com/eng/developers
iOS	Objective-C, C	Requires Apple Developer Account developer.apple.com/iphone
MeeGo	Qt, Web Apps, C++, others	Intel and Nokia guided open source OS (based on Linux) meego.com/developers
Symbian	C, C++, Java, Qt, WebApps, others	Open source OS built from the ground up for mobile devices www.forum.nokia.com/symbian
webOS	HTML, CSS, JavaScript, C	Supports widget style programming, (based on Linux) developer.palm.com
Windows Mobile	C#, C	.NET CF or Windows Mobile API, most devices ship with J2ME compatible JVM developer.windowsmobile.com
Windows Phone	C#, VB.NET	Silverlight, XNA frameworks create.msdn.com

Carriers and Other In-The-Wild Testing Problems

If you've ever experienced unexplained inconsistencies with the performance of your mobile app, there's a good chance that carriers and connections are the culprit. Perhaps more than any other criteria, issues associated with carriers are almost always found outside of the lab environment.

So how can anyone:

- Replicate diverse locations?
- Test on-device in NYC, SF, LA, DFW, CHI?
- Test abroad?
 - SIM cards?
 - Carriers?
- Test GPS-based intelligence?
- Test IP-based localization?

Without moving a portion of your testing out of the lab and [into the wild](#), there's literally no possible way to ensure quality across carriers and locations. Here's a good story from a European tech exec that pretty much sums it up:

"We found, for instance, that one of our UK servers was not using the closest node. It was sending information to Australia for no apparent reason. **Without testers physically present in those areas** – telling us how long it took to search, stream and play music – there's absolutely no way we would have known about those issues."



THE MOBILE WEB INVASION



The Rise of Mobile Web

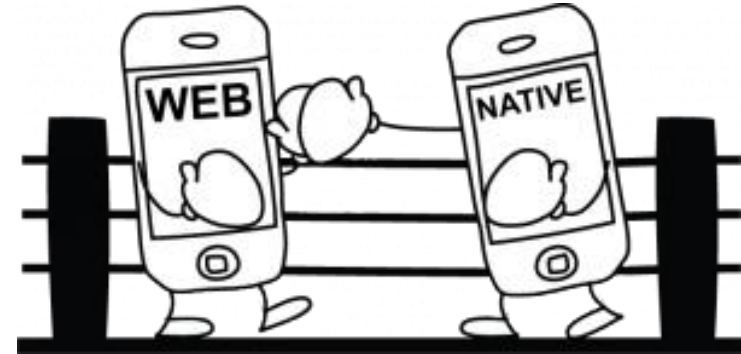
Native apps broke onto the smartphone scene running full steam ahead. But lately some companies have been opting for mobile web apps and mobile optimized websites. The advance of HTML5 has been speeding this trend along.

Mobile web is attractive for several reasons. It is often cheaper and easier than creating and maintaining native apps because mobile web offers developers the advantage of creating one Java-based source code that will function across platforms. Native apps need to be tailor-fit to the desired OS – requiring development using different coding languages, which, as we saw earlier, can be challenging and a hassle.

So if mobile web is so much easier, why hasn't it replaced the native app? Well, for all the perceived ease of mobile web development, it also presents a unique set of testing challenges. It also offers different features than native apps. For this reason mobile web is growing as an addition to, not replacement for, native apps. Which is where the advent of the hybrid app comes in.

Ultimately, neither native apps or mobile web is going to win out any time soon. So it is important to know the features and issues of both. And don't forget to keep an eye on mobile web trends so you can stay ahead of the curve.

Pros and Cons: “Apps have the obvious downside of requiring more development resources, especially to be truly optimized for each device. If a company doesn't have enough resources to do this right, it's better to have a nice mobile site than a lame app.” – [Jakob Nielsen](#)



By the Numbers: A recent [survey by SD Times](#) found that “only 50% of small and medium sized businesses check the appearance or functionality of their website for smartphone users. Of those that have, 43% admit their sites have a reduced appearance and reduced functionality. Further, 57% of these businesses have not optimized their websites for mobile use and have no plans to do so.

Mobile Web vs. Native Apps: Differences in Testing

Native Apps

- Testing begins with app installation and launch
- Testing on some mobile devices requires access to a device ID
- Functionality and Usability need to be tested on multiple devices. Consider:
 - Operating System and version
 - Screen size
 - Custom themes
 - Interruptions
- Many native apps have access to additional APIs. Those connections need to be tested.



Mobile Web

- No installation required
- How does the site render in different browsers
 - iOS runs Safari
 - Android runs a stock browsers, but also supports Chrome, Dolphin, Opera Mini and third party browsers
 - Windows Phone runs IE
 - Blackberry runs a native browser, Opera Mini or Bolt
- Requires an internet connection,
 - Connectivity varies by location
- Load time is extremely important to mobile web users ([source](#)):
 - 60% of users expect sites to load within **three seconds**
 - 74% of users will only wait **five seconds**



The Future: Mobile Web *Plus* Native Apps

Native apps still have a higher user rate and remain the only option if your app needs to access specific APIs (such as the camera or address book). Native apps also offer a highly controllable custom experience because they are developed specifically for a designated operating system – so you can control exactly how it looks on each device. Not to mention the built-in exposure of being included in an app market.

For these reasons, native apps are not going away any time soon. But mobile web does reduce app maintenance and allows for more content. As a result, mobile web is emerging not necessarily as competition to the native app, but as a potential ally. Companies are now beginning to embrace either both types of apps or hybrid apps which combine elements of the two into a single app.



NATIVE APP OR MOBILE WEB?

- “44 percent of respondents only offer a native application, as compared to 22 percent that offer a Web application, or 35 percent that offer both.
- When we asked them about future plans, the percentage of those offering a Web app or both remained the same, while many of those with native app alone said their app strategy choice remains to be seen and may change over time.”

– Lie Luo
Global Intelligence Alliance

Pandora’s CTO Tom Conrad on hybrid HTML5-native apps:

“It’s the best way to get the best of both worlds with the technology that’s available right now. That gives you integration with the OS and really, really high performance and really fluid user experiences. But integrated with some HTML5 content, whose strong suit is uniform platform dynamics, and rapid turns on user interface development.”

The Distant Future: Hybrid Apps

When companies decide that want to marry the native app icon and interface with the ease and content richness of the mobile web, they turn to hybrid apps. Hybrid apps give developers access to the necessary APIs when needed, but also room for additional content that does not need to be specifically formatted according to OS.

Hybrid apps take the shape of a native app with an OS-specific user interface. This native app can offer limited information and navigation options but also gives the user the ability to dig deeper by connecting to the mobile web. Alternately, developers may just make a native icon that connects directly to their mobile website. Having a hybrid app has the added bonus of giving the app visibility within app markets while also maintaining a mobile optimized website for casual browsers.

Examples of Hybrid Apps:

- **ESPN ScoreCenter**
 - A native app with updated scores within the app
 - Directs users to a mobile website for further articles
- **Lotte**
 - 100 pages written in HTML and used across platforms
 - A smaller number of custom developed native pages
- **Bank of America**
 - A native app icon pushes users directly to the bank's mobile site



A Side-by-Side Comparison: Native Apps vs. Mobile Web vs. Hybrid Apps

	Device Access	Speed	Development Cost	App Store	Approval Process
Native	Full	Very Fast	Expensive	Available	Mandatory
Hybrid	Full	Native Speed as Necessary	Reasonable	Available	Low Overhead
Web	Partial	Fast	Reasonable	Not Available	None

Source: [Worklight](#)



Ask an Expert

Have testing questions?
Let one of our testing experts guide you.

Ask an Expert ➔

“A second downside of apps is that users have to install them. Our testing shows poor findability and usability in Apple’s Application Store, and many users won’t even bother downloading something at all for intermittent use. So ask yourself whether you’re really offering something within the hardcore mobile center of need: time-sensitive and/or location dependent, and whether your offer is truly compelling in this crowded space. Most companies are never going to make it big in mobile. In some cases all they need is to make their main website somewhat mobile-friendly. Many others should deliver a dedicated mobile site but not bother with apps.”

- Jakob Nielsen

What The Near-Term Future Holds

Neither native apps nor mobile web are going to go away any time soon and with that in mind, here are a few trends to keep an eye on.

- In recent months, roughly 50% of smartphone users accessed native apps while 49% of used the mobile web ([comScore](#))
- Strides are being made to increase the speed of mobile web, which would make it more competitive with native apps. [See sidebar]
- Mobile web versus native app use fluctuates according to industry. For example, in retail 51% of shoppers use mobile sites while only 28% use native apps. Remember, users have different expectations for apps based on industry.
- Some companies (like [CNET](#) and [Kayak](#)) are redesigning their websites to mimic mobile sensibilities.



GOOGLE'S NEW WEB PROTOCOL

“The net result [of tests loading 77 webpages across 31 domains] is that using SPDY results in a mean page load time improvement of 23% across these sites, compared to HTTP. This is equivalent to a **speedup of 1.3x** for SPDY over HTTP. Much more work can be done to improve SPDY performance on 3G and 4G cellular networks, but this is a promising start.”

– Matt Walsh,
Ben Greenstein,
Michal Piatek

Google's Mobile Web Performance Team

ACHIEVING REAL-WORLD TESTING COVERAGE



Conquering the Mobile Testing Matrix

By now, it should be obvious that traditional testing methods are no longer sufficient on their own to combat the unique challenges presented by mobile app testing. So what's a tech leader to do? Hire more testers in key geographic markets? Construct an elaborate QA lab? Spend more on simulators and emulators? Cross your fingers and hope for the best?

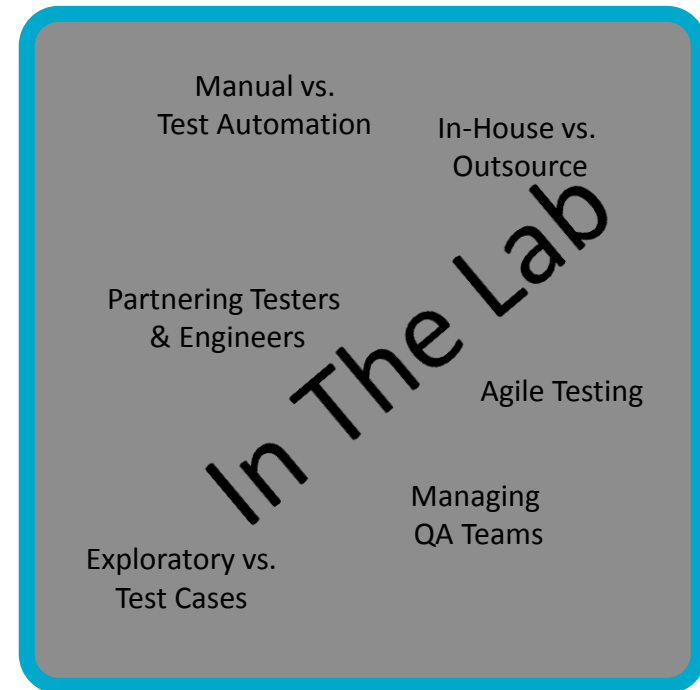
For years, as new testing challenges have emerged, new schools and approaches were debated. Choices and alternatives included:

- Manual testing vs. automated tools
- In-house teams vs. outsourced partners
- Guided testing vs. exploratory testing
- Emulators vs. remote access

In each instance, these innovations took place inside the confines of the QA lab, either behind the company firewall or in a QA lab halfway around the world. Starting to see the problem?

Historically, when companies wanted to improve their testing, they did so within this somewhat sterile environment – far removed from where their users work, live and play. The evolution of mobile eliminates that option as an effective solution on its own. If testing for modern applications is conducted exclusively within a central location, how can one be sure that it will work in the hands of actual users?

One solution: Hundreds of top companies – from global enterprises to early startups – have moved a portion of their mobile app testing outside the lab and into the wild. This enables applications to be tested by professional testers, on real hardware, with real software, using imperfect connectivity. In short, *under real-world conditions*. That's not to suggest that other methods are insufficient. Let's take a closer look....



Test Sourcing Options: Lab + Wild

In-The-Lab Testing

- **In-House:**
 - **Pros:** In-house team makes communication easier; professional testers who are tightly integrated into SDLC
 - **Cons:** Capacity constraints; testing may not match user demographics; maintaining adequate collection of mobile devices is prohibitively expensive
- **Outsourced Testing**
 - **Pros:** Professional testers; access to range of devices
 - **Cons:** Hidden costs; not easily scalable; challenges in communication and time zone; still lab-based testing

In-The-Wild Testing

- **Beta Testing**
 - **Pros:** No direct costs; lives outside the test lab environment; range of mobile devices, carriers, OS, locations
 - **Cons:** Users are not professional testers; provide subjective feedback that is not diagnostic, which raises the cost for the company; also puts unfinished product in front of customers
- **Crowdsourced Testing**
 - **Pros:** Lives outside the test lab across real devices, carriers, OS and locations; professional testers who mirror end users; access to range of mobile devices & connections
 - **Cons:** There is a direct cost; requires careful partner vetting; requires communication efforts between testers & developers



Emulators & Simulators

In many ways, the use of mobile emulators and simulators is essential for developing and testing a quality application. Such technology enables developers to verify certain functionality that is not specific to any device, carrier or operating system.

They are also very useful in terms of usability, and especially design, including data input, screen size, button use, etc. – all from the convenience of their own laptop. However, pointing and clicking with a mouse on a desktop monitor isn't the same as using a finger on a small screen.

Remember, the very nature of emulators and simulators means testing of applications is occurring in an environment far removed from the real world, where actual users run and interact with those applications ON their devices.

Consider, for instance, an application that is data entry intensive. To assume that it can be fully tested on a simulator (i.e. with full keyboard and mouse access) and be ready for release is a mistake mobile app developers have continually made. Other actions (like scrolling) differ greatly from a mouse to a touchscreen, and features like accelerometers and geo-location simply cannot be replicated. Not to mention that people hold and use phones in different ways – sometimes entirely one-handed.

The convenience of simulators and emulators has made it easier than ever to be lured into a false sense of security. **But the advantages of such tools are limited in scope, and should never be considered a substitute for real-world, on-device testing.** Last we checked, there weren't any emulators or simulators dealing with rush hour in LA or testing while in the fields of Iowa.

USERS USE REAL DEVICES

“So there I was, on my iPod Touch, trying to get to a list of users whose name started with the letter “I.” It worked great on the simulator with a mouse, but with the actual iPod, my finger was too fat to click the single line of pixels.

Suddenly it hit me: ***This is different.*** Sure, all of the old GUI rules apply, but suddenly we have a new set of ways the application can fail.”

– Matt Heusser

Reminder: Some companies offer labs filled with real devices for testing. But these tests still rely on computer generated reports. Just because an app didn't crash doesn't mean all the graphics loaded correctly or there were no GUI bugs. For that, you need real users with real devices.

From Surviving to Thriving: Your Strategy Moving Forward

Now that we've covered the testing challenges (and solutions) associated with mobile apps, you may feel as though your current testing strategy is hopelessly flawed. It isn't.

To achieve success in the mobile world, you'll still need to:

- Keep testing in the lab
- Keep hiring and training
- Keep automating
- Keep innovating

But recognize that this approach will never again be sufficient on its own. For reasons already discussed, you'll need to move a portion of your app testing efforts out of the lab and into the wild to mirror your user base:

- Technologically: OS, browser, device, carrier
- Geographically: Continent, country, city, language
- Demographically: Age, gender, education, employment, industry



“Don’t be fooled by the word ‘wild’ when it comes to testing mobile. When you think of the term ‘[in-the-wild testing](#)’ think of it as ‘real-world vs. lab conditions.’ **This is not outsourcing or beta testing, and it’s definitely not suggesting you replace your existing QA or the solid processes you have in place within your test lab.** Rather, this is about complementing, scaling, and ensuring that *a portion* of your testing mirrors your end users as closely as possible. The key is to align your in-the-lab testing with your in-the-wild testing.” – Matt Johnston, CMO, uTest

5 THINGS THAT YOU OUGHTA' KNOW



For QA & Test Leaders

- “Bite the hand that feeds you (a little). Don’t be a jerk, but make sure you don’t go passively along with bad ideas. A weak willed tester is a bad thing. Make your concerns known...and know when to fold.” – **Patrick Copeland, Senior Engineering Director, Google**
- “What doesn’t work is assuming that because you are the test manager, or the tester, that you *know* how your testing adds value to the company, that you *know* what the most important testing is, and that you *know* the managers and executives are idiots because of what they are telling you to test. It’s not up to us to judge them for either being idiots or for making clumsy requests that don’t actually help them resolve their concern. It is up to us, however, to help them figure out what they actually need, help them figure out how to meet that need, and then guide our testing to best serve that need.” – **Scott Barber, CTO, PerfTestPlus**
- “The top reason we should care about diversity in our testing teams is because the demographic of a computer user is more diverse than ever before.” – **Lanette Creamer, Owner, Spark Quality**
- “I suggest taking two weeks and actually measuring how the team is spending its time. Oh, not for reporting – it is very important the team stop the time-tracking after two weeks and not hand individual metrics into management for evaluation. Instead, we want to use the numbers for improvement. For example, many of the people I talk to can spend 80% of their time in meetings, working on documentation, working on compliance activities, doing email, and so on. That only leaves 20% of the time to test! Just pushing those numbers from 80/20 to 60/40 will double the amount of time the team spends actually doing testing.” – **Matt Heusser, Writer and Consultant**
- “Our job is vigilance. To lose vigilance is to abdicate our responsibility. Vigilance, in testing, means being a good skeptic. We must reject certainty in any form. We’re the Knights of May Be. To believe is to cease questioning; to fall asleep at our posts.” – **James Bach, Author and Consultant, Satisfice**



For Developers & Engineering Leaders

- “Are developers and testers friends or foes? I’m sure this varies from company to company and team to team, but I’d wager a sum of money that teams where the two are friends *dramatically* outperform teams where they’re foes.” – **Andrew Muns, President, Software Testing & Performance**
- “If test can work with development to improve the quality of the software prior to code complete, that will improve the speed of the whole system. Realizing this, and having a little bit of data to “prove” it, can help the entire system improve.” – **Matt Heusser, and Consultant**
- “When people make a repeated error using my code, instead of asking why these people are idiots, I learned to ask what’s wrong with my software that causes the nice people to look like idiots.” – **Cem Kaner, Professor of Software Engineering, Florida Institute of Technology**
- “In mobile app markets today, users have an instant, strong, and public vote on the quality of an app through star ratings and comments. Interestingly, they don’t often complain about failed unit tests – what matters is any impact on their experience. Be sure to focus on integration, network fault tolerance, the device, other apps, and most importantly, user perception and expectations.” – **Jason Arbon, Engineering Director, uTest**
- “Engineering and quality best practices defined during the PC and Web eras are not well suited to the agile and mobile environments of today. Only dinosaurs will blindly port the practice they are comfortable with to mobile – it requires a re-imagining of engineering to be competitive, efficient and relevant.” – **Jason Arbon, Engineering Director, uTest**



For Product Managers & Marketing

- “Find the right mixture of testers who are close enough to the product to become real experts with it and specialist testers who are expert breakers. Call it whatever you like but product expertise and testing expertise are the two ingredients that I find make for a good product.” - **James Whittaker, Partner Development Manager, Microsoft**
- “But even then, we don’t *assure* quality; we *question* it on behalf of our clients. The people who are building and managing the product *assure* quality.” – **Michael Bolton, Principle, DevelopSense**
- “If you change a core feature, you need to test everything again – even if it was working and should still be working because your other tests show it all fine.” – **Michelle Sullivan, Founder, SORBS**
- “Study your actual users: invite a handful of representative customers to your location and run them through simple usability studies of your software. One day in the lab is worth a year in university lecture halls, in terms of actionable lessons learned. (And remember that your “usability lab” can be a regular office or conference room — as long as you shut the door.)” – **Jakob Nielsen, Usability Expert and Principle, Nielsen Norman Group**
- “Expectations can be managed once you have established a credible track record of delivering product. Here data and process are invaluable. If you can prove your points with data, then it makes it very hard to refute it, whereas hand waving just sets you up for constant failure. From a process perspective, setting up a framework so that other teams know what is required to be completed to allow the product to ship or what is the real definition of Feature Complete, for example, puts everyone in a better position.” – **James Sivak, Director of QA, Unidesk**



For Executives & Business Leaders

- “I want software projects as a whole to run more smoothly and more predictably. I really think that’s what software testing is all about — reducing the uncertainty of software development and finding ways to muscle errors out of the process. A process in which mistakes are harder than doing the right thing is the ultimate goal. We can’t eliminate them, but we can make doing the right thing to be the easiest thing to do.” – **James Whittaker, Partner Development Manager, Microsoft**
- “I don’t consider performance testing optional if you’re making money online. Here is how I would think about it: “What is the cost of an outage to my company?” Think about it in terms of dollars lost as well as customers lost.” – **Dan Bartow, VP of Product Management, SOASTA**
- “Teams need to be able to set aside time to look at everything from the proverbial forty thousand foot view. Only then can they make true progress with goals that may take a year to implement but that have huge ROI opportunities.”– **James Sivak, Director of QA, Unidesk**
- “I urge you not to use expensive tools, even if they work. Never let your manager buy them. Because expensive tools become something you MUST use, even if they don’t work. A free tool may be freely abandoned. This gives you flexibility.”—**James Bach, Author and Consultant, Satisfice**
- “To me, the biggest weakness [in the way companies test software] is not considering software testing anything but a (barely) necessary evil. Testing is seen as something that could be done by a troop of monkeys, so serious testers are treated like third-class individuals. You treat people as if they are stupid, then they will wind up acting stupid.” – **Gerald Weinberg, Author and Consultant**



TIPS, TRICKS AND OTHER RESOURCES



Tips and Tricks

Are you a niche tester holding on to a Samsung Galaxy Mini or running Cupcake so you can ensure new apps work under all conditions? Do you ever wonder if it's worth it? Well here are a few great sites that let you monitor phone, platform and browser popularity. Check these out if you're wondering if upgrading will hurt your testing matrix or if you're holding on to a dinosaur that might not be worth testing with.

- [Android Developers](#) shows you graphs of data collected about devices that have access to the Android Marketplace over a recent 2-week period. The site breaks the information down into platform popularity and percentage of distribution, then combines the most recent stats with previous studies to show you trends over the past few months. They also have a graph for screen size popularity, but it's limited to "small, medium, large and extra large."
- [Stat Counter](#) looks at trends from January 2011 to January 2012 and offers stats by browser, browser version, mobile browser, operating system, mobile OS and a few other, non-phone specific options. The charts are easy to read and offer both a line chart or a bar graph version. Plus, you have the option of saving the chart as a JPEG, downloading the data into a CSV spreadsheet or embedding the chart into your HTML. It's really a neat site, especially if you're researching new hardware or wondering whether you should update your operating system.
- AppBrain offers visitors a look at the [most popular Android phones](#) and the [most common SDK versions](#). Perhaps more interesting than the charts themselves (which cover from a one year of data to only a few months) is the feature at the bottom of the page that shows changes over the past 30 days. It's an easy to read chart that tells you just how much market share (in terms of percentage) a phone or version gained or lost over the last month.

Read more: www.mobileapptesting.com/is-your-testing-doing-any-good/2012/02/

APPGRADER FOR ANDROID

What is AppGrader?

It's a free online tool where developers quickly test their Android application on some of the most popular devices.

What does AppGrader provide?

Within minutes developers receive a customized report with:

- An app grade on a scale of 1 to 100
- Issues identified during download, installation and basic usage
- Comparison of your overall grade to the most popular apps in the same category

Why should developers use AppGrader?

- Catch problems before users do
- Benchmark an app against the competition
- Ensure a good first impression **It's F**

How it Works

- Upload the Android APK file
- Send your Mobile App
- If the mobile app crashes on a device, AppGrader provides the device-specific crash log for details

[Check Out AppGrader for Android Now >>>](#)

Tips and Tricks: Android

We write a lot on the subject of mobile testing challenges specific to Android – with its countless devices, OS versions and other permutations. But for some reason, it means a lot more coming from a company like Netflix.

Amol Kher – the Netflix Engineering Manager in Tools for the Android, iOS and Apple TV teams – recently penned a terrific blog post on their first-hand experience testing their app on Android. I highly suggest you [read the post in its entirety](#), but here are my highlights:

- **On their testing challenges:** “When Netflix decided to enter the Android ecosystem, we faced a daunting set of challenges: a) We wanted to release rapidly every 6-8 weeks, b) There were hundreds of Android devices of different shapes, versions, capacities and specifications which need to playback audio and video and c) We wanted to keep the team small and happy.”
- **On automation:** “You probably guessed that automation had to play a role in this solution. However automating scenarios on the phone or a tablet is complicated when the core functionality of your application is to play back videos natively but you are using an HTML5 interface which lives in the application’s web view.”
- **On device priority:** “To put it other way, when it comes to watching Netflix, any device other than those ten devices can be classified with the high priority devices based on their configuration. This in turn helps us to quickly identify the class of problems associated with the given device.”
- **On their testing team:** “We keep our team lean by focusing our full time employees on building solutions that scale and automation is a key part of this effort. When we do an international launch, we rely on crowd-sourcing test solutions like [uTest](#) to quickly verify network and latency performance. This provides us real world insurance that all of our backend systems are working as expected. These approaches give our team time to watch their favorite movies to ensure that we have the best mobile streaming video solution in the industry.”

[Read the rest >>>](#)



Tips and Tricks: iOS

Even though the iOS operating system doesn't present as large of a matrix as Android doesn't mean testing iOS apps is easy. iOS devices are supported by more than 100 carriers worldwide, and the introduction of high res screens with the New iPad added yet another dimension to the matrix. Here are a few tools (most of which are already on the device) that will make iOS testing on real devices easier and more effective:

- **Crash reports** are logged automatically on Apple devices. These reports can be accessed by syncing the test device to a computer. The .crash files begin with the application name and contain date and time information. In addition, <DEVICE_NAME> will appear at the end of the file name, before the extension – making it easier to keep track of each report.
- The **console log** is an iOS feature that includes information from every application on the device. This log can help pinpoint if your app is being adversely affected by other apps/software on the device. The console log does not last very long so be sure to access it quickly following a crash, otherwise the details about an issue may be lost. The console log can be saved by connecting the device to a computer and accessing the "Console" tab within device configuration utility.
- The **built-in screen shot command** is a good tool to use to document bugs. Holding the home and power button simultaneously will send a snapshot of the device's screen to the iCloud (and subsequently to all your connected devices).
- **Instruments** is an application that traces and profiles iOS code. It is available as part of Xcode Tools. One of the most helpful instruments is the Leaks template which monitors memory usage of the app and detects memory leaks.

ESSENTIAL GUIDE TO iOS APP TESTING

Want to learn the ins and outs of mobile app testing for the Apple ecosystem?

uTest's *Essential Guide to iOS App Testing* details the iOS testing matrix, UDID provisioning and important areas of focus for testing iOS apps (including common hang-ups and issues), plus a lot of tools and tips for developers and testers.

[Get The Essential Guide to iOS App Testing](#)



Tips and Tricks: Windows Phone

It's important to remember that Android and iOS aren't the only operating systems out there. Windows is making a big push with Windows7 and soon, Windows8. The company has been actively courting mobile app developers. In late 2011 the Microsoft Windows Phone Marketplace hit the 50,000 app mark – a feat that took iOS 12 months and Android 19 months. It took Windows 14 months. Less than a month after that the Marketplace had jumped by another 10,000 apps.

What to Know about Windows Phone

- The recent introduction of low-end (cheaper) smartphones has hit a tiny testing snag ... not all apps will work on them. But as things stand now, 5% of existing Windows Phone apps won't work with the new, lower system requirements. ([Source](#))
- Windows8 apps can be tested in an iPad using Win8 Metro Testbed. This gave Windows developers a jump start on the new market before Windows8 was released. It also opens the market to developers without access to a Windows8 device. ([Source](#))
- Microsoft strictly adheres to their Metro style. Leading up to the release of Windows 8, the company published *Performance Tips for Metro Style XAML Apps*. The paper provides performance tips for metro style XAML apps. It provides guidelines for developers to improve performance of their apps in the scenarios most important to users. It assumes that you know basic XAML programming in C#, C++, or Visual Basic." Download the paper at <http://www.microsoft.com/en-us/download/details.aspx?id=29886>



“ Just for looks and beauty, I definitely favor the Windows 7 phone over Android. ”
– Steve Wozniak

AND IN CONCLUSION



Conclusion

Those companies who neglect testing in their [mobile application](#) development lifecycle, do so *at their own peril*. There was a time when the testing matrix was indeed too complex, too burdensome to be completed using standard means. But thanks to the rapid evolution of [in-the-wild testing](#), all that has now changed.

Those companies that are leveraging in-the-wild testing - particularly in the mobile apps and mobile web space - are gaining a competitive advantage with each new release. As the mobile market continues to grow sharply, those brands that make real-world testing coverage a priority will enjoy ROI in terms of increased market share, profitability and above all, user loyalty. Those who neglect testing will struggle to keep up in a world filled with app stores, social media and increased user expectations. Simple as that.

Either way, the future of mobile applications remains bright. Once reserved for a tech-savvy niche, the use of mobile applications is now firmly entrenched in the mainstream. There's no turning back when it comes to mobile apps, and that means the testing for mobile must rapidly evolve to keep up.



For more on how in-the-wild testing can complement your mobile app testing efforts, and help you launch apps that work in the hands of end users, chat with one of our testing coaches by [clicking here](#) or by calling 800.445.3914.

About uTest

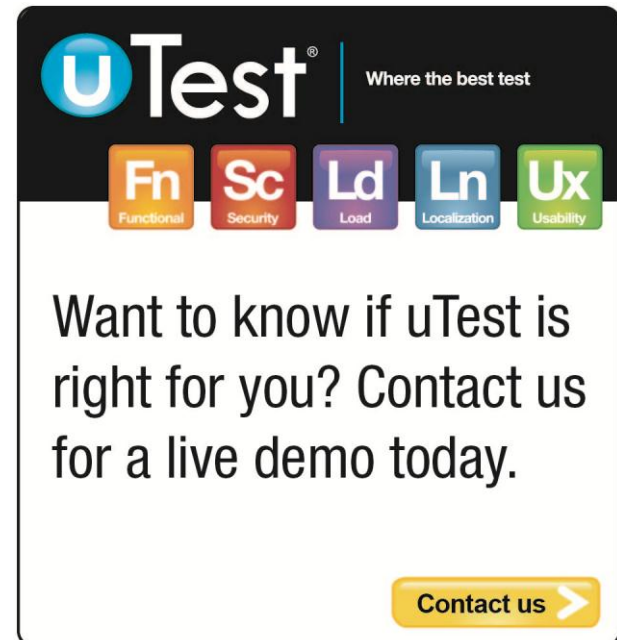
uTest provides real-world testing services for web, desktop and mobile applications. By leveraging a community of 60,000+ professional testers from 190 countries, uTest helps companies test their products under real-world conditions. Thousands of companies – from startups to global enterprises such as Google, Microsoft, HBO, Amazon and *USA TODAY* – turn to uTest to complement their in-the-lab testing, and to help them launch better apps. uTest's services span the entire software development lifecycle, including [functional](#), [usability](#), [security](#), [localization](#) and [load testing](#).

The company is headquartered near Boston, with offices in Silicon Valley, London and Israel. uTest has raised more than \$37MM in funding and consistently generates triple-digit annual revenue growth. The company won the American Business Association's "Most Innovative Company of 2011" award, and was named a "Best Place to Work" by the Boston Business Journal two years in a row.

More info is available at www.utest.com or blog.utest.com. For more info on mobile app testing, visit www.mobileapptesting.com

uTest, Inc.

153 Cordaville Road
Southborough, MA 01772
p: 1.800.445.3914
e: info@utest.com
w: www.utest.com

A graphic for uTest featuring the logo and tagline "Where the best test" at the top. Below the tagline are five colored boxes labeled Fn (Functional), Sc (Security), Ld (Load), Ln (Localization), and Ux (Usability). The main text reads "Want to know if uTest is right for you? Contact us for a live demo today." and a yellow "Contact us" button with a right arrow is at the bottom right.

uTest® | Where the best test

Fn Sc Ld Ln Ux
Functional Security Load Localization Usability

Want to know if uTest is right for you? Contact us for a live demo today.

Contact us ➔

