



Where the best test

# The Essential Guide to iOS App Testing

Tips for launching high-quality iPhone and iPad apps

## White Paper

June 2012

In-The-Wild Testing for Functional + Security + Load + Localization + Usability



## Whitepaper

# The Essential Guide to iOS App Testing

What You Need to Know About Testing Your iOS App

---

### Table of Contents

• <b>Introduction</b> .....	3
• <b>The iOS Matrix</b> .....	3
- iOS Hardware.....	4
- Carriers.....	4
- Other Matrix Considerations.....	5
• <b>iOS Focus Areas</b> .....	5
- UDID.....	5
- The App Store.....	5
Common Rejection Reasons.....	5
Permissions.....	6
- Memory.....	7
- VoiceOver.....	7
• <b>Helpful Tools and Tips</b> .....	7
• <b>About uTest</b> .....	9

*“An iPod, a phone, an internet mobile communicator....these are NOT three separate devices!*

- Steve Jobs  
Apple CEO, introducing the iPhone in 2007

# Introduction

Let's be honest. Saying you need to "test a mobile app" is just about as generic as saying you need to breathe gas. The reality is you need to breathe a very specific gas if you want to keep breathing. The same goes for mobile app testing. If you want your mobile app to survive and thrive post-launch, then you need to test it on the unique operating system it was designed for.

Whether you're developing for iOS, Android, Blackberry or Windows Phone, you will uncover unique design and system advantages and disadvantages during testing. While there are plenty of tips and tricks that can be used to [launch great mobile apps](#) in general, this whitepaper will take a more in-depth look at how to thoroughly test iOS mobile apps – specifically, apps for the iPhone and iPad.

When compared to other mobile operating systems, the "closed" nature of Apple has some advantages when it comes to testing. Having one manufacturer control all models of all iOS devices drastically limits fragmentation, unlike the Android matrix. There are two standout ways that mobile iOS devices have limited fragmentation:

- **Screen Size:** There are only two screen sizes to develop for with iOS – phone and tablet (the iPod Touch screen is very similar to the iPhone screen).
- **UI:** Android developers often run into unforeseen problems with custom user interfaces, which are fairly prevalent. Since Apple controls all iOS devices there is a single UI to plan for.

**"If you're a smartphone enthusiast who has had extensive playtime with the iPhone ... and Android smartphones, you'll probably have noticed how the former mobile operating system offers a much more smoother user-experience compared to the latter's general choppiness."**

- Awais Imran  
Tech Blogger, Redmond Pie

But testing for iOS still presents challenges.

**By the end of this Whitepaper you should have an idea of the iOS testing landscape and the unique issues that can arise along the way to iOS app success. We'll also point out some useful tips and tools to make the road a bit smoother.**

## The iOS Matrix

The testing matrix involves a combination of hardware, software, versions, operating systems, carriers, locations and features that make up the entire mobile ecosystem. The iOS world looks downright simple compared to the jungle of hardware-software-manufacturers that is Android. But there are still important factors to consider. The main

components within the iOS matrix are hardware (and corresponding supported software) and carriers.

### iOS Hardware

While Apple doesn't support multiple phone models, they do have three independent devices compatible with apps. The major Apple devices associated with app use are the iPhone, iPad and iPod Touch. Each device has gone through multiple generations, but since Apple stops pushing updates to older devices, you do not have to worry about supporting them all. This also limits the number of hardware/software version combinations you need to consider. Here's a breakdown of the devices and a snapshot of which support Apple's more recent system updates.

- **iPhone:** There are five different versions of the iPhone. However, only the 3GS, 4 and 4S handsets still receive software updates (earlier versions have been discontinued).
- **iPad:** Since the iPad is a relative new comer to the iOS lineup, all three generations are still supported. (Remember, the new iPad introduces a high retina screen to the equation – a new feature we're likely to see on future iOS devices.)
- **iPod Touch:** For development and testing purposes, the iPod Touch is largely the same as the iPhone. While the WiFi capable iPod Touch has gone through four iterations, only the more recent versions (the iPod Touch 3 and 4) are still supported devices.

**“If making it open would give us not the quality of Apple products all working together like they do, I would say keep it closed.”**

- Steve Wozniak  
Co-Founder, Apple

### Carriers

Wireless carriers come into play mainly when dealing with iPhones, but carriers are playing an increasingly large role with the iPad. The iPhone and/or iPad are available through more than 100 different carriers worldwide. Here's a quick (approximate) region-by-region breakdown:

- **North America:** 11 carriers
- **Europe:** 44 carriers
- **South and Latin America:** 19 carriers
- **Asia:** 35 carriers
- **Middle East:** 12 carriers
- **Africa:** 7 carriers

Each carrier presents a unique coverage footprint with varying signal strengths throughout to consider. How does your app perform with weak or overloaded signals? There is also the increasing divide between 3G and 4G coverage.

### Other Matrix Considerations

There are few other considerations to keep in mind when addressing the iOS matrix. One is that these devices are offered with several different levels of memory – including 8GB, 16GB, 32GB and 64GB. Remember to have your app only use as much memory as strictly necessary – people have a lot of music these days and if they have a 16 gig iPhone your app could be the thing that goes when memory space gets tight.

With the introduction of the New iPad (generation 3) the displays on Apple devices are also beginning to vary. If you develop an app specifically for the heightened display capabilities of the new iPad it will not perform well on earlier hardware. Likewise, your pre-existing app may not look so hot (or run as smoothly) on the new retina display. This can potentially alienate a user segment or require you to develop additional versions of the app.

## iOS Focus Areas

There are two sides to every coin (unless you're cheating), so with the positive aspects of testing on iOS (that limited matrix we just discussed), you also run into a few areas that are a bit more difficult than if you were working with a different system. Here are some key areas to focus on with your iOS app.

### UDID

A unique feature of Apple products is the device specific UDID ("unique identifier"). If you want to test your app on live devices *without* going through the app store, you need the UDID of each device involved in the test.

### The App Store

The most famous difficulty associated with iOS apps is the Apple store's notorious vetting process. There are a few things you can do to give your app a better shot at clearing that huddle without incident. Avoid these major iOS faux pas for a better shot of getting your App to the big show.

### Common Rejection Reasons

Apple will often give developers one or more reasons for why an app was rejected. Here are a few rejection reasons as explicitly stated by Apple or reported by developers:

- **Limited Audience:** Apps intended for limited audiences are sometimes rejected. The key to the rejections seems to be trying to limit the sale/download of the app. Apps that are targeted toward a limited audience but available to all iOS users seem to fair better.

- **Repetitive Apps:** Apps that are not “useful or [does not] provide some form of lasting entertainment” – in other words, an app that is already over done in the App Store.
- **Unrefined/Unprepared Apps:** An app that looks like it was poorly created, not tested or rushed through production. The Apple App Store prides itself on offering quality apps.
- **Violence:** Apps depicting realistic violence or realistic weapons used in an irresponsible way. (Angry Birds type “violence” is OK.)
- **Improper API/Background Services Use:** Trying to beat the system by using one of the iOS APIs or background features in a way other than its intended use. (i.e. Running a silent audio clip in the background to enable the app to remain running.)
- **Poaching Keywords:** You cannot use the names of pre-existing apps for the keywords for your new app.
- **Limited Features/Little Usefulness:** Joke, novelty and brand showcase apps can be caught by this rule if they do not offer users any actual usefulness. (i.e. An app that only displays one image and has no additional features.)

A more detailed list of reasons for app rejections can be found in the App Store Review Guidelines (see **Helpful Tools and Tips**). Bear in mind, new reasons often crop up in rejection letters to developers.

### Permissions

Apple has strict rules on what apps can and cannot do. Here are a few key things that you should avoid if you want your app to pass.

- **UDID:** Remember that unique device identifier? Some developers would have apps access this ID for tracking and advertising purposes. In early 2012, because of privacy concerns, Apple began rejecting apps that accessed the UDID without alerting users. For the time being developers can still access the UDID but must first ask user permission.
- **Collecting Personal Data:** Apple will reject apps that collect personal user data without first notifying and obtaining a user's permission. Notification is often incorporated into the app installation process, with a user's continued installation considered consent.
- **Address Book Access:** Remember the debacle when it was revealed that Path was accessing users' address books? Since it came out that this was a common practice for apps, Apple has disallowed apps from accessing, transmitting and storing contact information without user permission.
- **Excessive Permissions:** Before CarrierIQ and Path hit the news for accessing data without permission it wasn't uncommon for apps to access far more data and applications than really necessary. Those days are gone and developers now need to be transparent about what data and applications their apps will be able to access.

## Memory

Memory plays a big role when it comes to iOS devices. Apple devices range from 8GB to 64GB and your app will be expected to perform on each device regardless of memory capacity. (iPad exclusive apps only need to plan for 16GB, 32GB and 64GB.)

Be sure to test the memory usage of your app and for memory leaks (see “Instruments” in the **Helpful Tools and Tips** section of this whitepaper for more on how to test for leaks). Keep in mind that an app may perform differently when there is limited memory available compared to when there is ample free memory. How does your app perform when there are other apps running simultaneously?

## VoiceOver

VoiceOver is Apple’s accessibility feature that allows users to interact with their devices via spoken commands. VoiceOver will also describe an app’s UI to users. With iOS 3.0 Apple introduced a UI Accessibility programming interface to help developers ensure their app is VoiceOver compatible. Apple expects applications intended for iOS 3.0 or higher (which is not all applications) to be VoiceOver compatible.

# Helpful Tools and Tips

Here are a few resources that might make iOS development and testing a bit easier.

- First and foremost review the [App Store Review Guidelines](#) – available to registered Apple developers. The guidelines “provide rules and examples across a range of development topics, including user interface design, functionality, content, and the use of specific technologies.” Even if your app is pristine and passes every test you throw at it, if it doesn’t fit within these guidelines it was all for nothing.
- The [iOS Human Interface Guidelines](#) will help ensure your app’s UI meets Apple’s expectations. Topics include: Platform Characteristics, Human Interface Principles, App Design Strategies, User Experience Guidelines and Custom Icon and Image Creation Guidelines, among others.
- **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,

*“The app approval process is in place to ensure that applications are reliable, perform as expected, and are free of explicit and offensive material. We review every app on the App Store based on a set of technical, content, and design criteria.”*

- Apple App Store Review Guidelines

<DEVICE\_NAME> will appear at the end of the file name, before the extension – making it easier to keep track of each report.

- A useful companion to the crash report is the device **console log**. 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 the 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).
- Testing on an emulator is fine, but it’s always good to also test on real devices before launch. Emulators differ than real devices when it comes to CPU type, screen interaction (mouse versus touch screen), memory capacity and usage, zooming functionality, network connectivity (emulators tend to be stationary while devices can test while mobile), sleep mode and accelerometer response to name a few.
- **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.
- Even if your app works flawlessly on iPhone and iPod Touch, be sure to test it again before porting it to the iPad.

It is important to regularly review all of Apple’s developer guidelines – from App Store review policies to Human Interface expectations – as they are continuously changing and updating. Knowing Apple’s policies and being aware of changes before they take effect can save you time and effort.

**“Apps that worked well on the iPhone may become unstable when they migrate to the iPad. Only thorough testing can reveal bugs before you release your app.”**

- Debi Zylbermann  
Writer, Ezine Articles

## About uTest

uTest provides in-the-wild testing services that span the entire software development lifecycle – including functional, security, load, localization and usability testing. The company's community of 50,000+ professional testers from 180 countries put web, mobile and desktop applications through their paces by testing on real devices under real-world conditions.

Thousands of companies -- from startups to industry-leading brands – rely on uTest as a critical component of their testing processes for fast, reliable, and cost-effective testing results.

More info is available at [www.utest.com](http://www.utest.com) or [blog.utest.com](http://blog.utest.com), or you can watch a brief online demo at [www.utest.com/demo](http://www.utest.com/demo).



**uTest, Inc.**

153 Cordaville Road  
Southborough, MA 01772  
**p:** 1.800.445.3914  
**e:** [info@utest.com](mailto:info@utest.com)  
**w:** [www.utest.com](http://www.utest.com)