Optimizing Your Mobile Test Strategy:

How to Build a Test Device List with the Device Planner

Contents

1 Introduction 3

2 Challenges of Testing Mobile Applications 3

3 How to Optimize a Test Device List 4

3.1 Optimization Method One: Market Share Approach 4

3.1.1 Market Share Approach in Theory 4

3.1.2 Disadvantages of the Market Share Approach 5

3.2 Optimization Method Two: Criteria Coverage Approach 5

3.2.1 Using the Criteria Coverage Approach 6

3.2.2 Disadvantages of the Criteria Coverage Approach 7

3.3 How to Build a List Using the Criteria Coverage Approach 7

4 Example: Building an Optimized Android Test Device List 8

4.1 OS Version 9

4.2 Manufacturer/Device Family 10

4.3 Screen Resolution 10

4.4 Operator 11

4.5 CPU 12

4.6 Physical Keyboard 12

4.7 Devices to Add 13

4.8 Devices to Delete 14

4.9 Final Android Test Device List 15

5 The Keynote DeviceAnywhere Device Planner 15

6 Conclusion—Market Share vs. Criteria Coverage Approaches to Creating a Test Device List 16

1 Introduction

Enterprises are placing increased focus on the mobile medium, as technological advances in devices allow for more full-featured mobile applications, and employees and consumers expect access to their applications anytime, from anywhere. One of the biggest challenges facing enterprises embarking on a mobile strategy is device diversity: the variety of platforms/OSes, OS versions, screen characteristics, and other hardware differences presents a seemingly insurmountable challenge for enterprises that aim to offer a quality mobile product to its customers or internal users. How can enterprises ensure that their product will work across all available target devices and platforms without busting the QA budget?

In an effort to assist enterprises developing mobile products, Keynote DeviceAnywhere has developed a methodology and tool for creating lists of target smart devices in order to maximize test coverage. This methodology is designed to minimize the set of target test devices while maximizing coverage, thus minimizing QA cost and time to market while still promoting delivery of high-quality mobile products. The strategy involves optimizing device selection based on coverage across key device differences instead of prioritizing testing based solely on device market share.

This white paper covers:

- The unique challenges of providing a quality mobile application that functions across diverse devices
- The overall theory and approach for creating an optimized device list
- An example of an optimized list, with the quantified benefit
- The Device Planner tool, offered as part of the Test Center Enterprise product suite, that implements Keynote DeviceAnywhere's recommended methodology and easily enables you to build a customized test device list

2 Challenges of Testing Mobile Applications

The mobile device market is marked by diversity along a number of axes that can affect the performance of your application. Chief among these is OS. However, many other device characteristics can have an impact on application performance.

Device Characteristic	Potential Impact
Input method	Most new devices use a touchscreen for input, but others still have keyboard-only input or keyboards in addition to touchscreens. Applications have to be designed to support multiple keyboard configurations. Potential issues can include missing input keys, pop-up touchscreen keyboards that obscure areas of the application, or touchscreen control bars that are misaligned.
Screen resolution / screen size	Unreadable text, blurred images, misalignment of screen elements and integral buttons, and items that fall off the visible screen are some of the issues that testing across screens of varied pixel density can isolate.

Device Characteristic	Potential Impact
Memory / CPU	Applications must be able to handle operation gracefully in conditions of low or insufficient memory or processing power. In addition, applications with heavy graphics or fast responses may not perform within acceptable standards.
Manufacturer / Family	Unlike iPhone and BlackBerry, Android OS devices are produced by multiple manufacturers and are customized by these manufacturers, resulting in differences in UI that can impact application functionality. Examples include on-screen and physical controls that function differently across devices, or customized handling of various input and device events. For instance, on some devices, the touchscreen keyboard layout is changed from default Android to Swype. Even the simple process of sending a text message or opening a Web page can be customized to open a third-party application instead of the native Android application. Motoblur is an example of manufacturer customization.
Operator	Certain operator restrictions can affect your application as well. For instance, Verizon does not allow data usage while on a voice call. Thus a networked app that is interrupted by a voice call must be able to gracefully handle interruption of the data connection and reconnection upon call completion, if required. Also, carriers can offer devices with customized firmware versions provided by the manufacturer. Again, this allows for the customization of certain defaults or for third-party applications to run during certain actions on a device.

All told, a prioritization strategy is needed to ensure maximum results from well-targeted tests.

3 How to Optimize a Test Device List

The goal of any testing strategy is to cover as many potential use cases as possible, and given limited resources (which is almost always the case), prioritize test cases that provide the most coverage. This is true of both product functionality coverage and target platform coverage. With the device diversity in today's market, adequate coverage of target platforms for mobile products is particularly challenging.

3.1 Optimization Method One: Market Share Approach

One method of maximizing coverage is to determine the set of devices with your target OSes that will have the highest incidence of accessing your application. So if you support both iOS and Android, and your application will be used across millions of iPhone 4s but only thousands of Motorola Droids, you prioritize testing on the iPhone 4 above the Motorola Droid. Your test plan will consist of testing on as many devices as you have time for out of a set of devices prioritized by market share. This is called the Market Share Approach.

3.1.1 Market Share Approach in Theory

If you are developing a consumer application, you can estimate market share of device models from purchased data or by using analytics. Purchased data is often imperfect for a number of reasons: your application's target market might not represent the market as a whole, and most purchase data is only approximated through surveys. Data from analytics embedded within your application provide the closest approximation to the breakdown of devices accessing your application, although no collected data

can accurately predict the future, and what you really want to know is which devices will access your application once it is available in the market.

Even for internal enterprise applications, unless your application is intended for only a small number of company-provided device models, your user base probably looks a lot like the consumer market as a whole: your employees have a variety of personal iPhone, Android, and BlackBerry devices through which they want to access your enterprise application.

Nonetheless, you can typically gather approximate market share data and utilize it to prioritize your test device list. You choose devices with the highest market share and test on as many as you have time and resources for. So if your QA cycle allows for testing across 10 devices, you test on the 10 devices with the highest market share.

3.1.2 Disadvantages of the Market Share Approach

The Market Share Approach ensures that you test on the most popular devices. However, by focusing only on market share, you could be missing a whole class of devices that are less popular, but when combined, still comprise a significant portion of the market. And you could be wasting resources testing on two devices that are very similar—your application is extremely likely to work on both of these devices if it works on either one. The next section describes an alternative approach that takes into account both market share and device characteristics to optimize your test strategy.

3.2 Optimization Method Two: Criteria Coverage Approach

To explain why targeting testing based solely on device market share does not provide optimal test coverage and why a different approach is better, we must explore the reasons for testing on diverse devices and platforms at all. The fundamental reason for testing on multiple devices and device platforms is that there is a risk that an application that works on one device might not perform as well on another. So the goal in testing across devices is to minimize the risk that the application will fail in the field, causing a poor customer experience.

The best way to minimize this risk is to test your application across all devices that it will ultimately run on. Unfortunately, most companies do not have the bandwidth or budget to do so. There are over sixty Android devices launched to date in the United States. Even iPhones, produced by only one manufacturer, have four hardware versions (five if you count the CDMA version) and many more hardware/OS version combinations. As a result, companies must turn to creating a *representative* device list that maximizes test coverage while remaining within budget.

As mentioned above, one method of creating such a representative list is the Market Share Approach. You choose the devices with the highest market share and test on as many as you have time and resources for.

A second method of creating a device list is the Criteria Coverage Approach. You identify differences across devices along key characteristics that might cause incompatibility with your application and target your testing on devices that exemplify those characteristics. Chief among these characteristics is OS. But even within an OS, key characteristics that should be considered include, but are not limited to: OS version, screen size/resolution, manufacturer/device family (Android), operator, CPU, and input method. So if your QA cycle allows for 10 devices, you choose 10 devices that are popular but also different from each other along OS and key criteria, and maximize coverage across your chosen criteria.

So, in the Market Share approach, you create a list strictly based on market share of a model. What can be missed in this approach is adequate coverage of key device variables that can interfere with your

application's performance. The Criteria Coverage Approach improves on the Market Share Approach by replacing some more popular devices with somewhat less popular devices to get more coverage across important criteria, thus minimizing the risk of issues arising on devices you have not tested. The more limited your test time, the more important it is to use this approach to maximize coverage.

3.2.1 Using the Criteria Coverage Approach

The rationale behind optimizing your device selection strategy by criteria is that there is a higher chance that an application that works on one device will work correctly on a similar device vs. a dissimilar device. In other words, the greater the difference between devices, the greater the risk that an application that works on one device will encounter issues on another.

A simple illustration of this theory follows. The below table lists characteristics of two popular and similar Android devices, the HTC Incredible and the HTC EVO 4G:

Device	HTC EVO 4G	HTC Incredible
Operator	Sprint	Verizon
Current OS Version[1]	2.2.1	2.2
Device Family	HTC Sense	HTC Sense
Screen Resolution	480x800	480x800
CPU (MHz)	1000	1000
Processor Type	Single Core	Single Core
Device Memory	512	512
Touchscreen	Yes	Yes
Physical Keyboard	No	No

While one of the above devices operates natively on the Verizon network and the other on Sprint, and they sport different firmware versions, these devices are otherwise very similar in UI (HTC Sense for Android), screen resolution, CPU, processor, and memory. They also have similar form factors.

So, although they represent some of the highest market share among Android devices, testing on both devices is redundant across many axes, including input method, screen resolution/screen size, memory/CPU, and manufacturer/family. A better test strategy is to replace the HTC Incredible with the Motorola Droid 2 Global. This is another popular device, also on the Verizon network, with the Android 2.2 OS version. However, the Motorola Droid 2 Global has a different screen resolution, a more powerful CPU, a physical keyboard, and of course, a different UI and manufacturer. See the table listing characteristics of all three devices below:

Device	HTC EVO 4G	HTC Incredible	Motorola Droid 2 Global
Operator	Sprint	Verizon	Verizon
Current OS Version[2]	2.2.1	2.2	2.2

[1] As of April 30, 2011
[2] As of April 30, 2011

Device Family	HTC Sense	HTC Sense	Motorola Droid
Screen Resolution	480x800	480x800	480x854
CPU (MHz)	1000	1000	1200
Processor Type	Single Core	Single Core	Single Core
Device Memory	512	512	512
Touchscreen	Yes	Yes	Yes
Physical Keyboard	No	No	Yes

With the Criteria Coverage Approach, it is as important to decide what you should *not* test as what you should. You choose devices that are less similar for your test list, and you do *not* choose devices that are very similar along the identified criteria, as less is gained by testing them.

3.2.2 Disadvantages of the Criteria Coverage Approach

No test plan is foolproof. There is always a possibility that a handset that you decide not to test on has a defect that adversely affects your application. Net of such anomalies, if you cover the differences across devices that are likely to impact your application, you will have *minimized the risk* across more of the market that your application will not work on a specific device. The converse of this concept is that testing on two like devices affords you less value than testing on two unlike devices, even if the unlike devices have a lower market share combined than the two like devices.

How does this work in practice?

3.3 How to Build a List Using the Criteria Coverage Approach

Start by identifying target platforms (e.g., Android, iPhone, BlackBerry) and OS versions you intend to support (e.g., iOS 3.1.3 and higher). Then identify any criteria you want to cover in testing. Keynote DeviceAnywhere recommends including all of the criteria listed in Using the Criteria Coverage Approach above, as well as any additional key functionalities that could impact your application.

Next, determine the number of devices to test on. This is typically the number of devices you have time to test in your test cycle. If you are not sure, make a best guess; this can always be adjusted later based on your criteria coverage analysis.

Then, create a list of all devices in your target market with values for each of the criteria in question. (Keynote DeviceAnywhere maintains a database of smartphone devices and their characteristics that we utilize for this purpose—see The Keynote DeviceAnywhere Device Planner below.)

Next, prioritize these devices by market share (highest to lowest) using available data and then create your initial list starting at the top and continuing until you have reached your target device list number.

Now the difficult part: identifying the set of possible values for each criterion. Refer to your initial list, and for each criterion, identify the values that are currently *not* covered. Of these uncovered values, determine which are of the highest priority for your test coverage. These high priority values will be the ones for which a significant number of relatively popular devices must be added to your list. Identify a few devices for each of these values, prioritizing devices with higher market share. These are candidates for inclusion in the test device list. Repeat for each criterion, and develop a list of candidate devices.

Once finished, look over your candidate devices and find any devices that meet multiple criteria values that you are lacking. Add these devices to the original list.

Now look for devices in the original list that can be deleted because they are similar to other devices on the original list. Delete similar devices. Now you have a list that contains some of your original devices, while some original devices have been replaced by other devices that cover more criteria values.

This is a time consuming and iterative process, and there might be more than one right answer. Fortunately, Test Center Enterprise product offers the Device Planner tool, which utilizes access to the Keynote DeviceAnywhere features database, and based on your priorities, automatically creates an optimized device list. While the algorithms used to create this list are quite complex, the following section presents a simplified example of how this works in practice and outlines the creation of an optimized Android test device list.

4 Example: Building an Optimized Android Test Device List

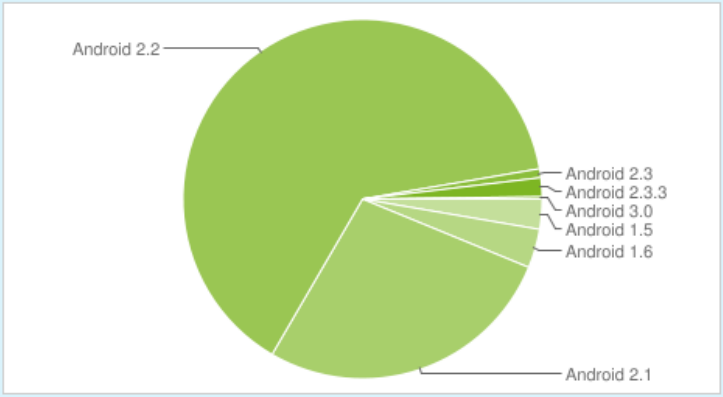
There is a lot of variety in the Android OS because multiple manufacturers build these devices. The ten most popular Android devices (not including tablets, to simplify the analysis) according to Amazon.com as of April 30, 2011 are:

Manufacturer	Model
HTC	Thunderbolt
Motorola	Droid X
Motorola	Atrix
HTC	Inspire 4G
HTC	EVO 4G
HTC	Incredible
Samsung	Continuum
Motorola	Droid 2
Samsung	Epic 4G
HTC	Evo Shift 4G

This seems at first like a pretty good list of devices to test on, as these devices have all enjoyed good sales or are being heavily marketed and are likely to have high future sales. However, let's look at coverage across a number of different axes of device characteristics.

4.1 OS Version

From Google, the distribution of Android OS version in the market is as follows:



Platform	API Level	Distribution
Android 1.5	3	2.7%
Android 1.6	4	3.5%
Android 2.1	7	27.2%
Android 2.2	8	63.9%
Android 2.3	9	0.8%
Android 2.3.3	10	1.7%
Android 3.0	11	0.2%

Data collected during two weeks ending on April 1, 2011

Here are the OS versions covered by our ten popular Android devices:

Manufacturer	Model	Original OS Version	Current OS Version[3]
Motorola	Atrix	2.2	2.2
Samsung	Continuum	2.1	2.1
Motorola	Droid 2	2.2	2.2
Motorola	Droid X	2.2	2.2
Samsung	Epic 4G	2.1	2.2.1
HTC	EVO 4G	2.2	2.2.1
HTC	Evo Shift 4G	2.2	2.2
HTC	Incredible	2.1	2.2
HTC	Inspire 4G	2.2.1	2.2.1
HTC	Thunderbolt	2.2	2.2

Not surprisingly, the majority of the devices on the list are on Android version 2.2 or a derivative, as that is what comprises the majority of the market. Ideally, however, a test list should have at least one 2.3.3 device, as many devices moved to that version over the summer. There was only one device model with this OS version being sold commercially as of April 30 2011: the Samsung Nexus S. This device might not have sales as high as some of the other devices on this list; however, testing on a newer Android version helps future-proof your application.

In addition, testing one device on an earlier Android version might be of interest; version 1.6 still has 3.7% of the market, although that share will dwindle over time. The HTC G1 is still on Android 1.6.

So we have now identified some possible candidates for the Android test list: the Samsung Nexus S and HTC G1.

[3] As of April 30, 2011

4.2 Manufacturer/Device Family

As you can see from the above list, the manufacturers HTC, Samsung, and Motorola are well represented. However, LG, Sony Ericsson, Huawei, and Kyocera have also launched Android devices in the US market. The LG devices, the Optimus S, Optimus T, and Vortex, all belong to the Optimus One family, and over 1.3 million units of all these devices had been sold in North America as of the end of 2010[4]. So, it might be important to consider one of these devices. The other manufacturers have only launched one Android phone to date, so you can consider testing on those devices only if time and resources permit.

"Device family" refers to Android customizations. Most manufacturers put out both customized and non-customized devices. The table below shows our popular device list with the corresponding device families.

Manufacturer	Model	Device Family (Android)
HTC	Inspire 4G	HTC Sense
Motorola	Atrix	Motoblur
HTC	Evo Shift 4G	HTC Sense
HTC	EVO 4G	HTC Sense
Samsung	Epic 4G	Samsung Galaxy S
HTC	Thunderbolt	HTC Sense
HTC	Incredible	HTC Sense
Motorola	Droid 2	Motorola Droid
Motorola	Droid X	Motorola Droid
Samsung	Continuum	Samsung Galaxy S

NOTE Motorola Droid is a version of Motoblur.

The most common device families are represented here; however, devices with the stock Google experience (i.e., without a customized UI) are not represented. Many Android devices on the market, including LG Optimus One devices, the G1, the G2, the Nexus One, and Nexus S, are built using the standard Android UI. These devices are also possible candidates for the device list. Note that the G1 and the Nexus S are already being considered for OS version coverage, and an LG Optimus for manufacturer coverage, so these devices can add representation for two criteria at once.

4.3 Screen Resolution

Let's look at our popular Android device list again and examine a new characteristic, screen resolution:

Manufacturer	Model	Current OS Version[5]	Screen Resolution
Motorola	Atrix	2.2	540x940
Samsung	Continuum	2.1	480x800
Motorola	Droid 2	2.2	480x854
Motorola	Droid X	2.2	480x854

[4] "LG 'Optimus One', Global 2 million mark 2010/12/10" (in Korean). lge.co.kr. LG Electronics, Inc. 2010-12-10.
[5] As of April 30, 2011

Manufacturer	Model	Current OS Version[5]	Screen Resolution
Samsung	Epic 4G	2.2.1	480x800
HTC	EVO 4G	2.2.1	480x800
HTC	Evo Shift 4G	2.2	480x800
HTC	Incredible	2.2	480x800
HTC	Inspire 4G	2.2.1	480x800
HTC	Thunderbolt	2.2	480x800

Three different screen resolutions are included in our list of ten most popular devices. However, the over fifty devices launched in the US market represent seven different screen sizes. Some of the sizes not represented are supported only on one device. But one screen resolution not represented, 320X480, is utilized by twenty Android models offered in the US market. Both LG Optimus Series devices and the HTC G1 mentioned earlier have 320X480 screen resolution, so adding either or both to your test list will give you improved coverage of screen resolution.

4.4 Operator

The table below presents our original Android device list again with operator information included:

Manufacturer	Model	Operator
HTC	Inspire 4G	AT&T
Motorola	Atrix	AT&T
HTC	Evo Shift 4G	Sprint
HTC	EVO 4G	Sprint
Samsung	Epic 4G	Sprint
HTC	Thunderbolt	Verizon
HTC	Incredible	Verizon
Motorola	Droid 2	Verizon
Motorola	Droid X	Verizon
Samsung	Continuum	Verizon

Note that there are no T-Mobile devices on this list. This is understandable, as T-Mobile is the operator with the smallest subscriber base. Ideally, however, at least one test device should be on the T-Mobile network, to ensure coverage of operator-specific customizations. There are many Android devices offered by T-Mobile, but the Nexus S, the LG Optimus T, and the G1, all devices considered for other criteria coverage above, are T-Mobile devices.

Also, as you can see, Verizon is perhaps over-represented. We will revisit this issue in Devices to Delete.

4.5 CPU

The table below presents our original Android device list with CPU information included:

Manufacturer	Model	CPU (MHz)
HTC	Inspire 4G	1000
HTC	Evo Shift 4G	800
HTC	EVO 4G	1000
HTC	Thunderbolt	1000
HTC	Incredible	1000
Motorola	Atrix	1000
Motorola	Droid 2	1000
Motorola	Droid X	1000
Samsung	Epic 4G	1000
Samsung	Continuum	1000

As you can see, the devices in this list are all relatively high powered. Looking at all device models in the market, the CPU values are:

CPU (MHz)	Count of Device Models
500	1
528	10
600	10
720	2
800	7
1000	25
1200	1

There are quite a few devices with far less powerful CPUs than those on our original list, and testing on lower-powered devices, depending on your application, might be important. Two devices that have been mentioned consistently above to provide coverage of other criteria can do so here: LG Optimus devices have a 600 MHz processor, and the G1 has a 528 MHz processor.

4.6 Physical Keyboard

A majority of Android devices follow the iPhone form factor closely, with touchscreen-only input. But there are still over fifteen Android models available in the US market with a physical keyboard. Both the Evo Shift and the Droid 2 have physical keyboards. So does the HTC G1.

4.7 Devices to Add

Additional criteria can be analyzed, but let's assume for the purpose of this exercise that all of the criteria relevant to your application's performance have been included above. Based on the preceding analysis, adding the following three devices can vastly improve your test coverage:

Device	Key Criteria Values Represented	Importance of Criteria to Market
T-Mobile Samsung Nexus S	OS Version: 2.3.3	Currently only one commercial device has this OS version[6], but most devices launched in the next 3 – 6 months will be on variants of 2.3.
	Device Family: Stock Google UI (G1 and Optimus also provide this benefit.)	Over a dozen Android device models in the US market use the stock Google UI.
	Carrier: T-Mobile (G1 and Optimus also provide this benefit.)	While T-Mobile is the smallest operator of the big four, it still has millions of subscribers using Android devices.
T-Mobile HTC G1	OS Version: 1.6	3.7% of worldwide Android devices (installed base) are still on this version.
	Screen Resolution: 320X480 (Optimus also provides this benefit.)	20 device models in the US market have this screen resolution.
	CPU: 528 MHz	10 device models in the US market have a 528 MHz processor.
T-Mobile LG Optimus T	Manufacturer: LG	Behind HTC, Samsung, and Motorola in the Android market for now—the big four operators combined have launched 6 LG Android devices in the US market to date.
	CPU: 600 MHz	10 device models in the US market have a 600 MHz processor.

[6] As of April 30, 2011

4.8 Devices to Delete

If your test device list is flexible in size, you can easily add the above three devices to your list. But if you only have time to test across 10 devices, which ones do you eliminate to accommodate the three additional devices?

In the same way that we explored differences across all devices in the market, we can look at similarities. Here again are the ten original devices with values for key criteria:

Manufacturer	Model	Operator	Current OS Version[7]	Device Family (Android)	Screen Resolution	CPU (MHz)	Physical Keyboard
HTC	Thunderbolt	Verizon	2.2	HTC Sense	480x800	1000	No
HTC	Evo Shift 4G	Sprint	2.2	HTC Sense	480x800	800	Yes
HTC	Inspire 4G	AT&T	2.2.1	HTC Sense	480x800	1000	No
HTC	EVO 4G	Sprint	2.2.1	HTC Sense	480x800	1000	No
HTC	Incredible	Verizon	2.2	HTC Sense	480x800	1000	No
Motorola	Droid X	Verizon	2.2	Motorola Droid	480x854	1000	No
Samsung	Epic 4G	Sprint	2.2.1	Samsung Galaxy S	480x800	1000	No
Motorola	Atrix	AT&T	2.2	Motoblur	540x940	1000	No
Motorola	Droid 2	Verizon	2.2	Motorola Droid	480x854	1000	Yes
Samsung	Continuum	Verizon	2.1	Samsung Galaxy S	480x800	1000	No

The Samsung Continuum and the Epic 4G are similar devices, both belonging to the Galaxy S family. Removing the Epic 4G has a smaller impact as the Continuum is the only device on the original list with the 2.1 OS version. While the LG Optimus T (one of the three devices to be added) also runs on Android 2.1, this OS version comprises over 25% of the market (see OS Version above) and should be represented by more than one device.

Other candidates for removal are one of the two Verizon Droids, and one or more of the HTC Sense devices.

- The Droids are similar, but the Droid X is slightly more popular than the Droid 2, thus we choose to retain the Droid X and delete the Droid 2 solely on basis of approximate market share. The Droid 2 contains a physical keyboard, but we still have two devices on the list, the Evo Shift and the G1, with physical keyboards.
- The AT&T HTC Sense device should remain—with only two devices, AT&T is already somewhat sparsely represented in this list. The HTC EVO 4G and the HTC Incredible are similar devices, and since the HTC Thunderbolt both is extremely popular currently and is the successor to the Incredible, the Incredible can be safely deleted.

Without any impact to testing the criteria identified as important to our application, we can remove the following devices from the original list:

[7] As of April 30, 2011

- Epic 4G
- Droid 2
- HTC Incredible

4.9 Final Android Test Device List

Manufacturer	Model	Operator	Current OS Version[8]	Device Family (Android)	Screen Resolution	CPU (MHz)	Physical Keyboard
HTC	G1	T-Mobile	1.6	Stock Google Experience	320x480	528	Yes
HTC	Thunderbolt	Verizon	2.2	HTC Sense	480x800	1000	No
HTC	Evo Shift 4G	Sprint	2.2	HTC Sense	480x800	800	Yes
HTC	Inspire 4G	AT&T	2.2.1	HTC Sense	480x800	1000	No
HTC	EVO 4G	Sprint	2.2.1	HTC Sense	480x800	1000	No
LG	Optimus T	T-Mobile	2.1	LG Optimus	320x480	600	No
Motorola	Droid X	Verizon	2.2	Motorola Droid	480x854	1000	No
Motorola	Atrix	AT&T	2.2	Motoblur	540x940	1000	No
Samsung	Continuum	Verizon	2.1	Samsung Galaxy S	480x800	1000	No
Samsung	Nexus S	T-Mobile	2.3.3	Stock Google Experience	480x800	1000	No

With a switch of three devices, this list provides more coverage across different OS versions, Android experiences, screen resolutions, CPU strengths, manufacturers, and operators than the original list. While it covers less of the market share across all Android smartphones, it provides excellent coverage of significant differences among Android devices.

5 The Keynote DeviceAnywhere Device Planner

We have provided a relatively simple example of building an optimized test device list in this paper; however, building a list is typically not so easy. You might have multiple target platforms and need to ration testing across all of them. You might have a set of criteria in mind, but you want to weight them differently as some are more important to test across than others. You might already have a set of devices in house that you would like to use and want to know what other devices would best complement them. And the market is always changing; it's a constant effort to stay abreast of new devices launched and OS updates announced (and sometimes not announced). In fact, as this paper goes to print, the Motorola Droid 2 Global, a device referred to in Using the Criteria Coverage Approach, already has a new firmware version available, and the Sprint version of the Nexus S with Android 2.3 has been launched.

Keynote DeviceAnywhere has developed the Device Planner, a component of the Test Center Enterprise product suite, which utilizes Keynote DeviceAnywhere's device features database and the logic described in this paper to build a customized and optimized test device list. The Device Planner is available for free

[8] As of April 30, 2011

at http://tce.deviceanywhere.com. We keep this system updated at least on a monthly basis, adding new devices and updating existing devices as they are upgraded to new OS versions. The Device Planner also allows you to enter your own existing devices you plan to test on and will fill in the remainder of your list, utilizing both your existing devices and new devices to optimize your test list. You can also easily change criteria weights and device list sizes, and build as many lists as you like.

Optimizing test effectiveness is an important component of providing a quality mobile application. The Test Center Enterprise tool set, designed to optimize mobile product quality, encompasses tools for test planning, test execution and automation, and post-release monitoring.

You can learn more about Keynote DeviceAnywhere and the Test Center Enterprise product suite at www.deviceanywhere.com.

6 Conclusion—Market Share vs. Criteria Coverage Approaches to Creating a Test Device List

The least risky way to prepare a mobile application for launch is to test it across all target devices. Absent this very expensive if not impossible proposition, one strategy is to test as many devices as you have time for, prioritized by market share. But as noted in Market Share Approach, this can leave large holes in functionality coverage. It may lead to ignoring entire classes of devices that, while not making it to the top of the market share list, might as a group encompass a significant portion of the market. And testing dissimilar devices (devices with differences across key criteria that are important to the functioning of your application) provides more test coverage than testing across similar devices.

Keynote DeviceAnywhere's Criteria Coverage Approach of identifying key criteria and testing across the more popular devices that cover relevant values for these criteria is an optimal strategy to maximize test coverage for your mobile application and minimize risk of issues arising in the field. The Device Planner exemplifies this approach in a simple Web-based tool to enable you to quickly build and make adjustments to a test device list based on a combination of device popularity and coverage of criteria that are important to your mobile application.

All testing strategies involve trade-offs, and by removing devices that have a higher market share from your test list, you add to the risk that defects or issues specific to those particular models are not discovered. However, by adding devices with characteristics that would otherwise have not been addressed in your test plan, you reduce the risk that you will experience issues in the field with an entire class of device.