The ultimate guide to smartphone app testing

So the hard yards of your smartphone application design and development are complete. Now it's time to put it in the appstore, sit back, and watch the money roll in, right? Well, not so fast. At the mercy of a very public user-feedback system and with a host of complex variables to take into account, smart app developers know that thorough testing of that clever new product can mean the difference between outrageous success and painful obscurity in the marketplace...

BY JAE RUSSELL

Initially high-end specialist devices aimed primarily at business-users, the smartphone market was once dominated by no-nonsense business-focused platforms such as Blackberry, Palm, Symbian and Windows Mobile. How times have changed. The advent of the Apple iPhone, and more recently with the rise of Google's Android, smartphone ownership is now very much a mainstream practice, with prices dropping steadily and ownership well within the reach of most people.

Currently global penetration of smartphones is around 30 per cent – higher in developed economies – and growing fast. Australia has the second highest smartphone penetration in the world at 37 per cent, and at the start of 2012 there were more than 800,000 such devices in New Zealand alone. Both the number and range of devices in use is increasing quickly meaning mobile-savvy businesses potentially have ready access to a very large audience.

While many mobile services can be supplied through mobile versions of web pages, the use of mobile apps allows the full functionality of smartphone technology to be exploited. The functionality of apps is almost limitless, and constrained only by imagination, and the future potential for development wide open. The leading app sources, Apple's App Store and Google Play, currently host around 600,000 and 500,000 apps respectively. Apple hit a total of 25 billion app downloads in March this year with Google reaching 15 billion not long after. Monthly download figures are now in excess of 1 billion + apps. For many businesses producing an app can be a great step towards further growth.

An app can be a powerful tool to reach people but care must be taken to ensure development is done well. With apps being developed by a range of people from 'darkened bedroom dwellers' to professional developers working in global companies, you want to ensure your app doesn't have an amateur feel that reflects badly on your brand. Ratings and comments made in app stores can have a huge impact, with negative comments and poor ratings meaning your app is likely to be avoided altogether, or uninstalled promptly, and an alternative installed. Even the coolest app won't be recommended to the user's friends if it fails to work. One bad App could turn your whole business sour. (It is estimated that around a quarter of the currently available Android Apps are 'low quality'.)

If you do decide to go to the effort of developing an app, a number of challenges need to be faced. It may well be that your target market is primarily a user of one specific platform (e.g. iPhones), which makes your development task much simpler. However, at the other end of the spectrum, if your target is the corporate or consumer market you must develop your App for the broadest range of devices to get good coverage of the market. For example, developing for Android, iOS and Blackberry will give you a potential market coverage of over 80 per cent *(see Table 1)*.

TABLE 1:

SMARTPHONE MARKET SHARE

Market share of the various Smartphone operating systems as at the last quarter of 2011.

Platform	Units shipped (millions)	Market share (%)
Android	81.9	51.6
iOS	37	23.4
Symbian	18.3	11.6
Blackberry	13.2	8.3
bada	3.8	2.4
Window Phone	2.5	1.6
Others	1.8	1.1

It's not possible to develop one app for 'The Smartphone'. You must consider the divergence of OS platforms in your target markets - for example iOS, Android, Blackberry, Windows Mobile, Windows Phone, Symbian, Bada, Hp WebOS. And then, even within each platform, different OS versions and smartphone models can add development challenges.

It is also not as simple as developing once and churning out a build for each platform. Imagine taking a document written in

»

PLUK CASE STUDY

An innovative new use of mobile app technology is the pluk app. Using the devices microphone to record sound bites from TV or radio, users can get additional information and specials offers delivered immediately to their phone.

To achieve maximum efficiency in development, pluk elected to use different development teams for the initial target platforms – iOS and Android. As a result, one challenge was to ensure consistency in both the product and testing. For this, and other, reasons the pluk team elected to have the app testing performed in one place by an independent testing agency – in this case Synapse. The desire was to achieve rigorous QA, testing and fault diagnosis in a consistent way across both platforms.

They also knew the value that non-functional testing added to ensuring they delivered a robust app. Given pluk is a consumer app, it was recognised that pluk would be used on a wide range of devices and it was important to ensure the users experience was optimal across them all. This included considering real world scenarios that could not be defined in a requirements specification. They could access proven experience in non-functional mobile app testing as well as get a fresh set of eyes reviewing the end to end product.

From the testing a number of issues were identified prior to launch, particularly around the UI and stability on a range of devices and on server interactions. Identifying these problems prior to release allowed pluk to deliver a more robust app to the market. Subsequent updates are also tested to ensure that new features haven't introduced new issues.

> “Developing a rigorous testing protocol is no simple task but you ignore doing this at your peril...Don't let this final step in development undermine your whole development effort.”

Chinese and running it through a translation tool. The output will still require a significant amount of work to make it a useable document in English. Even for a very simple app you will ideally customise the User Interface (UI) for each platform.

Critical to your app's success is good development and excellent testing across a range of target devices. I spoke with app developers about some of the development issues they experience when building multi-platform apps. While there are tools available to facilitate multi-platform app production, they are not the ideal solutions.

DEV Tip # 1: *"Cross platform development tools such as PhoneGap might do the job for the most basic of apps. If you have something that you can display as a website you can use HTML and PhoneGap to provide it as an app. You miss a lot of functionality using a tool like this (can't use platform specific code and access to some device functionality is hit and miss) and it runs a lot slower than a native app."*

Even development tools like Mono, which are designed to maximise your development effort and provide multi platform development support, are likely to require you to build your own device interfaces for some platforms as well as the UI – and simply don't support many, including BlackBerry.

When developing multiplatform apps there is also the challenge of managing a range of development environments:

DEV Tip # 2: *"There are different development tools for each platform and some of these have restrictions on how they can be used and what desktop environment you need."*

"You can sometimes split off and share most of the backend/ business logic, but you really need to build the UI at least and maybe device interaction for each platform."

So with the right tools and knowledge some of the initial development effort can be shared across the diverging platforms, but there will still be a need for customisation. Getting the UI right across multiple platforms can be a challenge in itself and thorough testing here helps uncover unexpected issues on each format:

***DEV Tip # 3:** "You get a lot of people who want a consistent look and feel between the various platforms. This tends to be a big headache, as you end up trying mimic another platform's UI - writing it from scratch. That can take a lot of time to code, and you only get a 90 per cent match. The UI experience can vary dramatically across devices and platforms and it is therefore important to test across a range of representative devices. You have to be careful about what the app does on different screen sizes. It is a good idea to check the tablet variants, and landscape devices and devices that switch orientation when you slide the keyboard out."*

One way to make the app development process easier and more efficient is to secure developers with existing relevant skill sets for the different coding platforms and the various target smartphones and their peculiarities. As with many things there is no substitute for experience here. Working closely with experienced developers in the early stages is likely to be valuable as they'll have some idea of what can be accomplished and how tricky the implementation is going to be on each platform. As the project progresses they'll have valuable knowledge to solve issues that arise:

***DEV Tip # 4:** "If you are coding on a new platform, sometimes you just start heading off in the direction you normally would and get half way through only to learn that a critical feature that you thought was standard wasn't available – or it wasn't the right way of doing things on that platform.*

"For example, Android does not enforce strong rules around device hardware specifications and formats which results in different methods for interacting with hardware functions. In addition manufacturers tend to apply their own UI interpretation over the core Android OS.

"Something that is considered standard on one platform might not be available at all on the other platforms – this can sometimes be filled in using 3rd party APIs, sometimes you need to work around it yourself.

"In addition, generic bug fixes might need to be ported across all platforms. But because each has been written slightly differently sometimes you have to go a completely different way about fixing things. Then there are all the platform specific bugs...."

Once these development challenges have been met, thorough testing is essential to identify issues in aspects that vary across platforms and devices. One of the issues in producing a robust multiplatform app is that developers don't know what they don't know. Rigorous and consistent app testing and fault diagnosis across all target platforms is critical in ensuring you don't damage your image or brand by releasing a malfunctioning app. App testing falls into two broad categories – functional (does it do what it's supposed to do and look how it should look?) and non-functional (does it work in a variety of 'real world' situations and target devices?).

Functional testing is usually well defined against a specification and covered by the development team – they won't release an app until it meets all the specifications listed. Non-functional testing can be more of a Pandora's box and needs to be performed under a number of conditions on a representative range of devices. This is a critical phase of development, extending the functional testing and optimising the experience for the greatest number of users. The app should display and work well (and intuitively) on all of the target devices under real world conditions.

The basics of non-functional app testing

Non-functional testing of apps should cover a whole range of other real-life scenarios that a smartphone and its user may encounter. You need to get inside the behaviour of the user and try to anticipate a range of scenarios that might occur and ensure the app will continue to function as it should and not cause a major malfunction of the phone. A comprehensive and often complex testing schedule is required to ensure as many potential issues as possible are discovered by the testers rather than the customers.

One of the more obvious areas to require testing is device compatibility. This includes rigorously testing the app across a range of device formats ensuring that varying screen sizes, processing capabilities, keypad types, touch screens, cameras, GPS and other types of device functionality are covered. This should include UI interaction, graphics and text presentation as well as device functions.

Other areas such as connectivity issues also need to be rigorously tested. For example, what happens when the device loses its internet connection while the app is synchronising with an online server? Does it recover gracefully and resume synching when the connection is re-established? End-to-end system testing may also be performed here to ensure the over-

»

“Anyone who has worked on a project for some time can be vulnerable to ‘inattentional blindness’ where errors are missed due to over-familiarity with the app. Have you ever proof-read a document you’ve written numerous times only to hand it over to someone else and have it returned with what should have been glaring errors highlighted?”

all solution works as expected in addition to the app in isolation. If the app accesses online data there can be a range of issues around providing a robust experience on a device that may be on and offline at any time.

Performance and stress testing is done to check that the app functions correctly, retains stability and responds in an acceptable manner under normal use including prolonged periods of spamming on key inputs (touchscreen, buttons, virtual keyboard and other incoming external signals such as Bluetooth) and interactions with other functions of the app. You don’t want the app to enter an unrecoverable state if the user’s cat walks across their device!

It’s also important to investigate how the app performs under load. That is, in devices with low memory, when running for long periods of time or when numerous other apps are running. Does it perform well and not result in aberrant operation?

Adversarial testing (does the app recover gracefully when power to the device is suddenly removed?), boundary condition (are any required restrictions on user inputs enforced?) and application profiling (measuring and comparing key measures of the apps functionality across various releases) should also be considered for testing, amongst other things.

Some challenges of app testing

Developing a rigorous testing protocol is no simple task but you ignore doing this at your peril. If you want a professional and useful app you must be thorough in this final stage of development. Don’t let this final step in development undermine your whole development effort. If you outsource testing to a specialist agency you can benefit from their existing knowledge and familiarity with performing these tests. They will have a proven structured approach to testing and can apply a consistent and rigorous test process. This will probably save you time and money.

Performing these tests across a range of devices also poses its own challenges. With smartphone manufacturers regularly releasing new and improved OS versions and devices, keeping current with this and maintaining a suite of representative devices can be costly and quite a commitment. Again, one solution is to outsource app testing to a dedicated testing unit meaning the cost of maintaining the suite of devices is spread across a number of clients. In this case you’ll also benefit from their intimate knowledge of a range of devices and their nuances, which also takes time to acquire.

Anyone who has worked on a project for some time can be vulnerable to ‘inattentional blindness’ where errors are missed due to over-familiarity with the app. Have you ever proof-read a document you’ve written numerous times only to hand it over to someone else and have it returned with what should have been glaring errors highlighted? The more familiar a person is with a project, the more likely they are to see what they expect to see when checking it rather than seeing what is actually there. This is a strategy the brain uses to filter out information – what has been seen several times already is paid less attention and subsequently, errors are often skimmed over. A ‘fresh set of eyes’ can reveal a number of faults that might otherwise be missed, as being new to the project these eyes will see what is actually there, errors and all, and pay more attention to the finer details than those who have been over them several times before. So it can be useful to isolate a structured testing process from the development environment for this reason also.

While developing for multiple platforms can be a challenge it will become increasingly essential as the uptake of smart devices, in a range of forms, continues apace and as more app providers wish to offer cross platform support. Whatever approach to testing is taken, it is critical non-functional testing be completed extensively to protect your image and brand.

ABOUT JAE RUSSELL//

Jae Russell has over 24 years experience working in the Telecommunications and IT industries, including the testing of desktop software, mobile applications (Apps) and mobile devices. Jae is experienced in the development and documentation of test and evaluation procedures and test plans, and in deploying and supporting software solutions. He currently runs Synapse Mobile and focuses on providing comprehensive App testing services to App developers.