

The Cross –Platform Challenge

"When you port between different platforms, you have to have expertise in all of them; you have to be expert in everything. That's the real challenge of working cross-platform."

Penrillian's Nick Whitehead on cross-platform challenges

There's a dividing line that runs through the recent history of mobile development; it separates the pre-app store world of first generation smartphones from the new app economy. The mobile market is a complicated place, but there's no doubt that the rise of the app stores has transformed it. What Apple started, others would dearly like to finish, and stores have proliferated. The players include network operators, handset vendors, and of course platform rivals - and that's where things become interesting. Every bit as much as they are showcases for apps, the stores are showcases for platforms.

Far from the mobile apps market consolidating around a single platform, or even two, it has instead become more diverse than ever. There are two ways to view that diversity: from an industry perspective, it's fragmentation; from a consumer perspective, it's market choice. Like it or not - and consumers, on the whole, seem to like it - the platform space is fragmented, and is likely to stay that way.

One consequence for developers is that multi-platform and cross-platform expertise has become increasingly important. It's no longer good enough (if it ever was) to be expert in only one platform.



Cross-platform UI engineering

All platforms are *not* the same, and nowhere is that more obvious than in the way that apps look and behave on different platforms. Every mobile platform has its own favoured styles of user interface (UI) and interaction (UX). As a developer, you are not expert in a platform until you are expert in those differences.

For users, it's those differences that give the different mobile platforms their character - and their value. BlackBerry users *like* BlackBerry, every bit as much as iOS users like iOS and Android users like Android. From the user perspective, the different platforms feel different to use. As Penrillian's Nick Whitehead says:

"The way an app works on BlackBerry can feel very different to the way an iPhone app works. It's not just how it looks; it's how the users use it, and get around inside it, how they switch in and out of it, how it sits with the platform and other apps. It's about how the app *feels* to the user. "

So working 'cross-platform' is not just about making apps run on different platforms; it's about making them feel native on each platform, to users of that platform. It may sound simple, but it usually isn't.

Touch is a good example: Touch screens, and touch-based gestures, have certainly changed mobile apps, and transformed the earlier interaction models, based on keypads, scrollers, and joysticks. But maybe not if you're a BlackBerry user. In fact, quite the opposite; touch has nothing to add to the 'classic' BlackBerry style, which is based on a track pad or scroller and the custom 'BlackBerry key' a shortcut key that lets you move around within an app and between apps. For developers, the BlackBerry key is powerful and flexible, because any function at all can be attached to it - say, zooming a map in a navigation app, or switching between emails in an email app. For users, it's an essential part of the classic style that makes a BlackBerry app *feel* like a BlackBerry app.

Another big difference between platforms is device variability. The iPhone, no surprises, is the most controlled and standardised of the mobile platforms, because Apple have complete control of device specs, and complete control of the software. When you write an iOS app, you know exactly what devices it will run on. BlackBerry, too, does a good job of controlling device variability; the different BlackBerry models are organised into device families based on form factor. The developer can target the form factor, and be confident that an app will run, behave, and look as intended on any device in the family.

With Android there isn't this consistency, although the Android UI, with its declarative approach to layouts, is quite good at sorting things out for itself at runtime. But variability on Android can be extreme - from the Sony Ericsson Xperia mini, say, with its 2.55 inch, 240 x 320 pixel display, to the

Samsung Galaxy Tab, with its 7 inch, 1024 x 600 display. You need all your cross-platform skills to design and implement a UI that will look good and feel natural to users of both devices.

At Penrillian we like to play to our strengths; we're software experts, not a UI design house. But design is nothing without implementation, and you might be surprised at how much of a successful app's UI and UX design is actually in the code.

Increasingly our customers are coming back to us for multiple native versions of successful mobile apps. For example, for Orange we implemented the 'Do Some Good' app for BlackBerry, and ['Your Orange'](#) for both BlackBerry App World and Nokia Ovi Store.

Penrillian has unrivalled cross platform expertise. If you need to deliver mobile apps that squeeze the best out of every platform and give all users a native app experience, we can help you.

[Contact us](#); we'd love to hear from you.