Android Commercialization – Making Android Ready for Your Device

Originally thought of as a mobile phone software stack, Android is now finding its way into all types of devices. Today you'll find Android being used in the development of devices ranging from smartphones, to mobile internet devices (MIDs) and tablets, to digital media devices like set top boxes and video on demand (VOD) systems. Let's take a look at why the use of Android is evolving, and what the challenges are in commercializing Android for use in these new use devices.

What is Android?

Android is not just a mobile device operating system or another Linux distribution. It's much more. It's a complete open source software stack that includes:

- A Linux kernel and devices drivers
- The hardware abstraction layer (commonly called the HAL)
- The Android runtime, the Dalvik Virtual Machine, and a set of libraries
- An application framework

We won't try to go into a technical explanation of each layer of the stack in this article. There are lots of other good references out there. Try http://developer.android.com/index.html as a starting point.

Why use Android?

As mentioned above, Android is a complete open source software stack. That means you are starting with a lot of the components you need pre-integrated together, rather than starting with a blank slate. This ultimately means a faster development cycle allowing you to focus most of your efforts on the value you add, the user experience and applications. Plus it's open source, so you have complete access to the source code to port and customize Android to any platform you need to. The bottom line is, developing on Android gives you a time-to-market advantage. In most cases, you can get products to alpha and beta stages faster using Android than starting from scratch.

Challenges

There is more to developing an Android device however than simply downloading the code, creating a few applications, and shipping your new device. Like most open source projects, Android is not commercial quality 'out of the box'. There are currently over 6000 issues reported with Android (see http://code.google.com/p/android/issues/list) of varying levels of seriousness. Will any of those affect your project?

Like any open source project, the rate of development and change with Android is very rapid. Overall this is a good thing as it drives innovation, but how do you bring a commercial product to market when things are constantly changing? Which Android version do you start with, Donut (1.6), Éclair (2.1), Froyo

(2.2), or the upcoming Gingerbread release? When you have a problem, where do you turn for support when you have a project schedule to adhere to and critical milestones to meet?

What architecture are you designing your device around? Is it currently supported, or will you be required to do a custom port of Android to your selected hardware? Are there peripherals on your new device that aren't supported in the Android framework today? If so, what's involved in enabling them throughout the Android stack?

When development is complete, how do you effectively test the entire software stack, from the kernel to the applications? How can you ensure you are delivering a quality user experience to your customers?

Android Commercialization

Android commercialization is the process of addressing all the challenges mentioned above and more. It involves:

- Putting together the right team with the right skills
- Prototyping, developing, and porting to your custom design
- Integrating 3rd party and open source software into the Android framework
- Fixing bugs and optimizing the final design

Let's address each one of these points in more detail.

As you begin a new project on Android, one of the first questions to ask is do you perform all the development yourself and do you have the expertise at all layers of the Android software stack? This can require developers that are familiar with platform development (Linux), framework development (Android) and even special requirements (telephony, digital media, etc.) If you don't have the skills, how do you proceed? Do you engage a commercialization partner to help or try to develop the skills in-house?

If you are enabling new hardware and peripherals that are not currently supported by Android you will need to work at the lowest levels of the software stack. This means working within the Linux Kernel and developing new device drivers. You will also have to add support for your new devices to the Android hardware abstraction layer (HAL) so the rest of the Android stack can recognize and interact with the hardware.

Depending on your device design and use case, you may need to develop new applications, or port existing applications to your custom design. You may need to integrate a combination of open source and proprietary software and debug the final image.

When you've completed your development you have to put together a test and QA plan and develop comprehensive test suites, or turn to commercial test suites. Google provides the Compatibility Test Suite (http://source.android.com/compatibility/cts-intro.html) to ensure application compatibility, but CTS doesn't test the complete software stack from the kernel all the way through the applications layer.

Finally, you'll need to optimize the software stack for memory usage and performance to get the most from your design and present the best user experience to your customers.

Summary

If current trends continue, the use of Android as a development platform is only going to increase as time goes on. As new versions of Android are introduced, new features and capabilities are introduced. It's been shown Android can cut the time required to deliver a new product to market, but it's not without some work on your part. Developing on Android takes careful planning, a solid strategy, and a complete understanding of what's really involved in developing a commercial product on Android.

About the author:

Dean Misenhimer is the Director of Marketing at MontaVista Software. He has over 15 years of experience in software development, technical sales, and Marketing. MontaVista Software is a wholly owned subsidiary of Cavium Networks (CAVM: NASDAQ). For over 10 years, MontaVista has been helping embedded developers get the most out of open source and Linux by adding commercial quality, integration, hardware enablement, expert support, and the resources of the MontaVista development community. To find out more about MontaVista's Android offerings, visit http://www.mvista.com/sol_detail_android.php