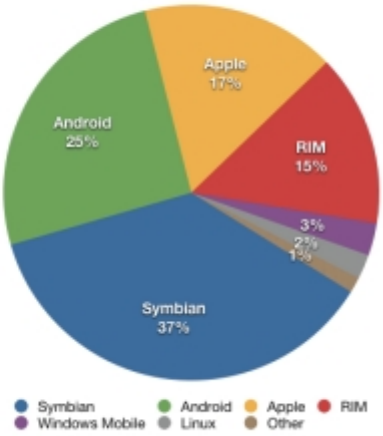
Android 装置的开发挑战：软硬件如何巧妙整合

- 随着科技的快速演进，现代人对行动通讯、无线上网与多媒体娱乐的需求更甚以往，所谓的智能型手机（Smart Phone）便成了炙手可热的个人消费电子产品之一，从Apple不断推出iPhone企图颠覆消费者对手机的想象、RIM推出主打商务功能的黑莓机、Google的Android系统让众家手机厂商争食大饼，到微软屡败屡战的从WinMo一路开发到WP7，智能型手机的这块战场可说是打的如火如荼。然而在这些众家竞争者中，Android可说是目前行情看俏的一套操作系统，以国际市调研究机构Gartner最新出炉 2010 年第三季的调查为例，采用Android操作系统的智能型手机在过去一年以来成长幅度最高，光是市占率便是前一年同期的七倍之多，销售量更是达到 14 倍的成长，同时也一举从市占率排名的第六名窜升到第二名。而在今年一月份甫落幕的国际消费性电子展（CES），也处处可见各式各样采用Android操作系统的产品。

* Gartner 2010 Q3 Worldwide Smartphone Sales

Worldwide Smartphone Sales to End Users by Operating System in 3Q10 (Thousands of Units)

Company	3Q10 Units	3Q10 Market Share (%)	3Q09 Units	3Q09 Market Share (%)
Symbian	29,480.1	36.6	18,314.8	44.6
Android	20,500.0	25.5	1,424.5	3.5
iOS	13,484.4	16.7	7,040.4	17.1
Research In Motion	11,908.3	14.8	8,522.7	20.7
Microsoft Windows Mobile	2,247.9	2.8	3,259.9	7.9
Linux	1,697.1	2.1	1,918.5	4.7
Other OS	1,214.8	1.5	612.5	1.5
Total	**80,532.6**	**100.0**	**41,093.3**	**100.0**



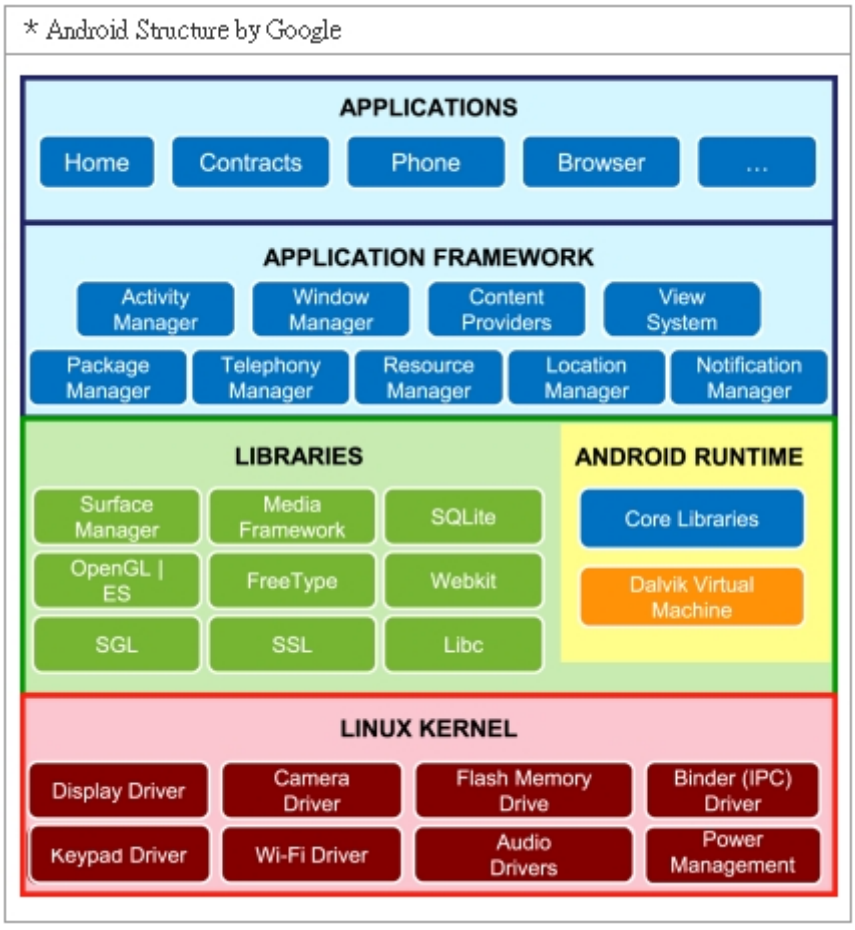
Source: Gartner (November 2010)

Android在过去一直扮演后起之秀的角色，切入智能型手机的速度似乎慢了苹果的iOS一步，但与Apple相同的是，它也成功的将其应用从手机移植到了平板计算机（Tablet PC）上。Android开放原始码（Open Source）的特性，能轻易地提高厂商对自家产品的接受度，更不用提背后Google的强力撑腰能带来多大的经济效益。目前可见包括手机厂商HTC、Motorola、SAMSUNG，以及计算机大厂HP与Dell等皆投向Android的怀抱，Android被广泛应用可说是势在必行。

尽管Android系统的普及看似指日可待，但在实际的产品应用上，也有其可能产生的问题风险。Android作为一个开放式的操作系统，是Google提供厂商的操作系统参考架构（reference design），厂商能有充足的发挥空间，以Android为基础向上开发设计自家产品，但也因为这样的开放性与自由性，让厂商在软硬件结合的这个环节必须下更大的功夫，像是如何挑选合适的硬件包括基频处理器、通讯芯片、触控感应芯片、天线与内存模块等，以及如何调整出最适当的软件设定等，更重要的是如何将软硬件整合，开发出差异化的产品。这中间所有的细节都会对产品最终样貌产生莫大的影响，像是其功能的完整度、使用接口的设计、效能表现（例如触控滑动画面、开启程序所需时间）、品质可靠度、甚至是后续的韧体升级动作等等。在此百佳泰便试图以专业中立的测试实验室角度，来点出厂商应用Android于手机、平板计算机或其它装置时应注意的开发重点，以希冀作为一个有效的参考信息。

解构 Android 基本技术架构

首先我们先来看到 Android 的基本技术架构，Android 是以 Linux 为核心，并采用软件堆栈（software stack）的架构延伸发展的一套软件平台与操作系统。根据下图可以看出，其基本架构分为五层：



· Linux核心（Linux Kernel）：以Linux开发提供最底层的核心系统服务，包括安全性（Security）、内存管理（Memory Management）、进程管理（Process Management）、网络堆栈（Network Stack）与驱动程序模型（Driver Model）。

· Android 执行环境（Android Runtime）：透过 Core Libraries（核心函式库）以及缓存器型态的 Dalvik Virtual Machine（Dalvik 虚拟机器）来执行程序。

· 系统函式库（Library）：使用 C/C++函式库的系统组件以供呼叫使用，开发者可透过上层的应用程序框架来运用这些功能，这也是主要 Android 装置的效能关键。

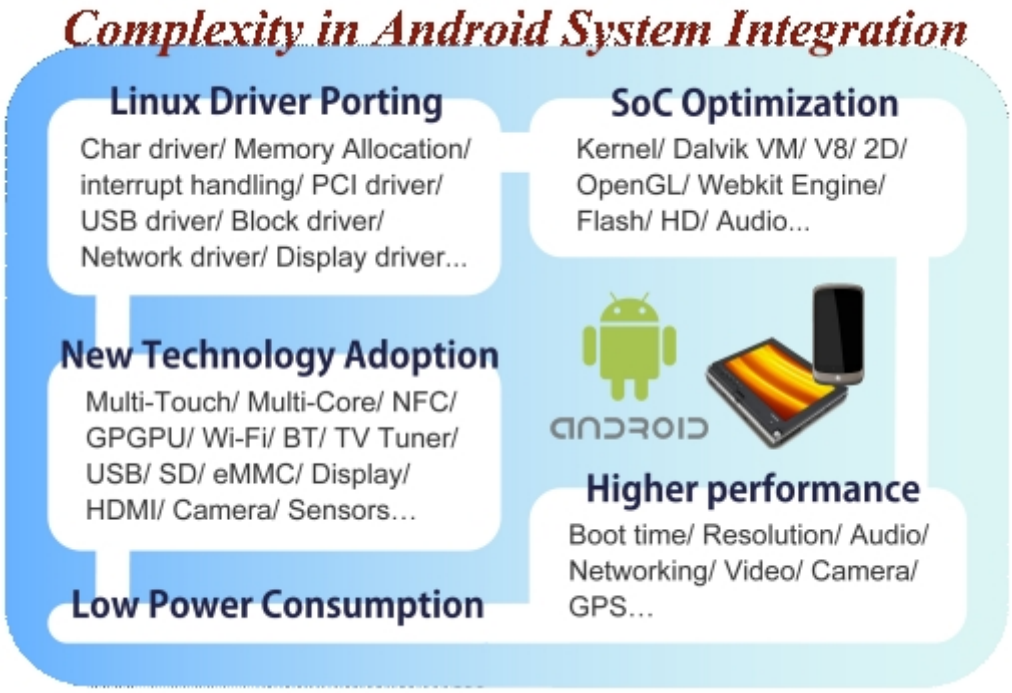
· 应用程序框架（Application Framework）：被设计来简化组件的再运用，开发者能完整存取使用与核心应用程序（Core Application）相同的API，应用程序可以发布功能并为其它应用程序所使用（需受限于其安全性限制），开发者也可运用同样的机制来新增与置换组件。

· 应用程序（Application）：所有 Android 应用程序皆是以 Java 程序语言编写，原始就会包含像是 Email、简讯、日历、地图、浏览器、联络人等其它应用程序，让使用者一开始就拥有这些基本功能，开发者也可在此客制其使用接口。

厂商越想要设计出与原始设定不同且增强效能的产品，便越需要对这五层架构进行修改。譬如像是多任务处理能力（multi-tasking），便可能需要修改包括 Linux 核心与应用程序框架的设计；而应用程序的开发者更可能需要针对应用程序与框架进行调整。由此可见，对 Android 装置而言，任何一个功能的置入或是对硬件设定的细微更动，都需要对 Android 系统进行从下到上的调整以达到最优化的效能，而这正是最为困难与需要验证的一环。

Android 装置软硬件整合的五大技术环节

如前所述，对众家开发厂商而言最大的挑战其实在于，如何将自己理想的产品诉求，与 Android 系统巧妙结合成一个功能完整并使用流畅顺手的产品，这其中牵涉了不同技术间的整合与运用。在此我们便根据其多年的测试与研究经验，归纳出五大 Android 相关装置在技术整合上的重要环节：



- **一、Linux 驱动程序的导入**

由于Android是根源于Linux所延伸出来的操作系统，因此各种关键功能的驱动程序也必须要能顺利的写入其中，举凡像是字符装置、内存的空间配置、中断处理、网络通讯、屏幕显示或是连接接口像是USB与PCI的驱动程序，这些可能是自行撰写、或是来自不同组件厂商的驱动程序，都必须要能被导入到Android系统，并维持良好稳定的效能表现。

二、系统单芯片的优化处理

对厂商而言，开发一款Android装置，不仅仅只是将所有零组件组合成为一个产品那么容易，最大的学问便在于将系统单芯片（System-on-a-chip，SoC）、各种新技术和Android系统进行整合，SoC涉及像是Dalvik Virtual Machine、OpenGL、V8、Webkit Engine等上层的演算，与Android间的结合便必须透过不断的尝试与验证，才能研发出既符合成本效益、又有良好效能的优化产品。目前市面上有些SoC厂商已针对Android系统的特性，提供整合过的SoC平台，将蓝牙、相机或上网等常用功能模块预先写入，减少终端成品厂商费力整合开发的时间，但对厂商而言，这样的预先整合是否适合自身产品，以及是否需要再作更细致的修改，则又是更困难的课题。

三、新技术的移植

随着技术的快速发展，更多新兴的技术规格也逐渐应用在手机等手持装置上，以手机为例，已经从过去以拨打电话为主要功能，转变为拥有各种多样化用途的产品。像是触控技术

让消费者可以透过手指的滑动传送指令甚至是具备多点触控的支持、Wi-Fi 模块提供随时无线上网的可能、通用图形处理器（General-purpose computing on graphics processing units，GPGPU）则能以并行方式透过图形处理器来执行通用计算任务、Android 2.3 版所支持的 NFC 近场通讯技术，以及更高阶的相机模块等等，背后都有各自的驱动程序与软件技术，也必须要与 Android 系统相结合使用。

四、效能表现的稳定

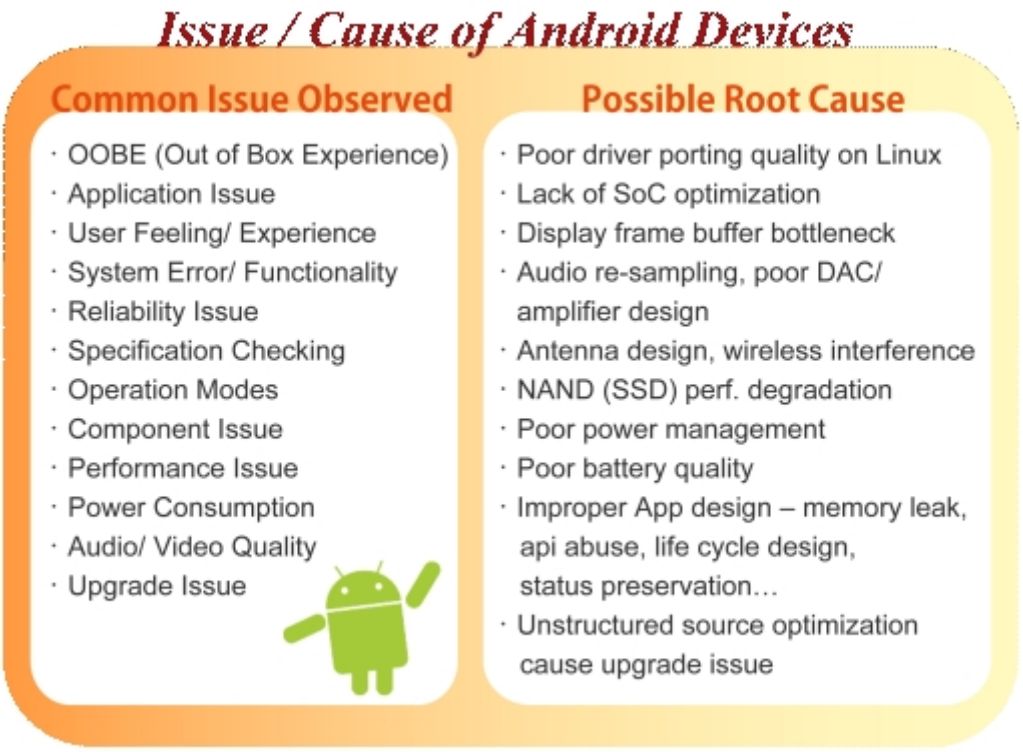
尽管上述这些技术不断推陈出新，但也都不能因此而牺牲装置原本的效能表现，让处理速度因此变慢或造成使用上不顺畅的状况。除了采用更好的硬设备外（例如现今处理器的时脉已迈向 1GHz），更需要操作系统的支持，像是如何在多任务运作的状况下维持程序执行速度以及系统满载的处理等等，都必须要透过软件面的奥援。也就是说，一台 Android 装置除了要能将各种功能与技术收纳起来、将软硬件整合外，更必须同时注重它在效能上能否维持应有的水准，以提供使用者在操作上流畅易上手的感受。

五、低电耗设计

Android 的设计概念主要是应用于可携式装置上，目前市面上可见的像是平板计算机与智能型手机等。对这类产品而言，电池续航力的好坏可说是影响消费者使用感受的关键之一，试想，若是一台智能型手机的待机时间过短，而使用者在外时又无法随时充电使用，不能实时的连网查询资料或执行其它手机功能，这样的产品便失去了它作为可携式行动装置应有的便利性。追根究底，良好的待机时间除了需仰赖高容量的电池以提供充足电力之外，另一个重点就是装置本身在被使用执行时能否作到低电耗设计。Android 装置让使用者能透过各种多样化的应用程序，来达到各种不同的使用目的，举凡像是单纯上网、观看新闻、邮件推播或是游戏等等，各种不同功能的程序都能透过自由下载使用，也由于其多任务处理与让程序背景执行的能力，更让降低耗电量成为开发者不可轻忽的一项课题。

持续验证修正，找出最佳 Android 整合方案

正如前面我们不断提到的，对 Android 装置而言，最困难的开发挑战便在于如何完美地”整合”软件与硬件，以开发出一项功能完整又同时注重使用者感受的产品。从对 Android 本身程序代码的修改、相关硬件的选择，到驱动程序的结合运用以及能否维持稳定的效能表现等，在在都必须要透过仔细的研究与不断的尝试，才能找出问题的根源并解决、更进而找出最合适的整合方案。



附图我们归纳出一些在Android装置上经常出现的问题与其可能肇因，而这些也都是开发厂商必须重视却可能忽略的一环。像是Android原始码中对音源的重新取样（Re-sampling）设计，就会导致装置在读取 48K音源时重新取样成 44K，而造成谐波失真的现象影响音质，这便是厂商不会注意到而未去修改的问题；另外像是天线位置的设计，也可能直接的影响到收讯能力的好坏；而不良的电源管理设计，也极有可能影响到装置在持续使用状态中的耗电情形。百佳泰在此仅以专业测试验证实验室的角度，希冀以宏观的方式，针对Android装置的开发设计提供可用的参考，近期内我们也将会提供实际的相关测试数据报告，并进一步指陈这些可能的问题风险，以期让更多厂商与消费者都能注意到品质验证的重要性，是从产品设计的根源就要开始层层把关。