**Android软件过程分析**

Godplayer小组

汽03 陈吉阳 匡旭 何坦瑨 黄强强

1. **Android发展历程**

Android之父Andy Rubin，曾在苹果、微软工作，后来想开发智能手机，2000建立danger公司，其sidekick是最早的智能手机之一。

2002年初，Andy Rubin在斯坦福大学给硅谷工程师讲课，其间谈到了Sidekick的研发过程。他的听众中有两个不平凡的人物——Google创始人拉里·佩奇和谢尔盖·布林。这是两人第一次与Andy Rubin结缘。下课后，佩奇走到Andy Rubin身边查看Sidekick，发现Google已经被列为默认的搜索引擎。“真酷。”佩奇由衷地说。也许受到Sidekick的启发，佩奇很快就有了开发一款Google手机和一个移动操作系统平台的想法。正是这样的想法促成他与Andy Rubin再次结缘。此为后话。

离开Danger后，andy创办android公司，着手设计一款对所有软件开发者开放的移动平台。至于为什么叫android，可能是因为andy非常喜欢机器人，而一位作家在其作品中将人形机器人称为android，然后这个称呼就被andy借用了吧。在andy拉风投的时候，突然想到了larry page，结果google几周后就完全收购了android。

2005年8月17日，[Google](http://baike.baidu.com/view/105.htm" \t "_blank)低调收购了成立仅22个月的高科技企业Android及其团队。安迪鲁宾成为Google公司工程部副总裁，继续负责Android项目。

2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良Android系统。随后Google以 Apache开源许可证的授权方式，发布了Android的源代码。第一部Android智能手机发布于2008年10月。Android逐渐扩展到[平板电脑](http://baike.baidu.com/view/74538.htm" \t "_blank)及其他领域上，如[电视](http://baike.baidu.com/view/8623.htm)、[数码相机](http://baike.baidu.com/view/13650.htm)、[游戏机](http://baike.baidu.com/view/15830.htm)等。

2011年第一季度，Android在全球的市场份额首次超过[塞班系统](http://baike.baidu.com/view/2059103.htm" \t "_blank)，跃居全球第一。 2012年11月数据显示，Android占据全球智能[手机操作系统](http://baike.baidu.com/view/148382.htm" \t "_blank)市场76%的份额，中国市场占有率为90%。2013年09月24日Google开发的操作系统Android在迎来了5岁生日，全世界采用这款系统的设备数量已经达到10亿台。

1. **Android软件过程分析**

Google发布Android时，建立了由34家手机制造商、软件开发商、电信运营商以及芯片制造商共同组成的“开放手持设备联盟”（OHA），这样就将用户和软件开发人员紧紧结合了起来。Android具有2个预发布的内部版本“阿童木”和“发条机器人”，这些内测版本相当于原型模型，采用了一些未经充分论证的技术（比如自己开发的虚拟机Dalvik、opencore代替FFmpeg等等），作为早期可运行版本表现出了新一代互联网手机的特性（开放性、可移植性等），从而在发布之前就获得了用户关于系统可用性的反馈，系统具有便于进化的特性。

而从2008年9月发布第一版以来，每过几个月就会发布更新的版本，短短5年来Android就已经推出了9个重要的新版本，而一些变化较小的版本更是数不胜数。这体现了短周期迭代的特性，在很短的时间间隔内就交付软件增量。Android不断更新拍照、网络、GPS、短信、邮件、多媒体等各个方面的功能。这些功能都有针对性地面向了科技发展趋势和生活习惯变化，比如触屏手机、便携拍照、智能电视、3G网络等，反映了客户的需求。与此同时，Android的架构、需求和设计都掌握在google自我组织的OHA手中，开发人员主要来自于google并相对稳定，google内部保持了良好的研究氛围，这些都非常符合“敏捷开发”的原则。

所以，Android系统早期的过程模型是原型模型，而进入商业化以后则采用了敏捷开发。

1. **Android对于Google的商业价值**

Google为何要收购Android？费时费力又不讨好的事情没有人回去做，那么Google究竟是如何在这样一个开源的平台上实现盈利？

首先我们要弄清楚Google的商业模式，Forrester研究公司的分析师Charles Golvin言简意赅地点评道：“把搜索、地图、Ggoole Talk等即时通信程序都和广告进行挂钩，那就是他们的商业模式。”所以，Google的盈利原理就是用他们的服务捆绑住尽可能多的顾客，Google那不引人注目的文字广告创造的利润就能滚滚而来。

通过Android这样一个开放而强大的操作平台，他们驱动了成千上万的手机品牌，google的服务不再受到终端的限制，不用担心某一天apple的ios对Google关闭大门。Google在手机业领导了一场新的革命——广告主的广告能到达更多的目标客户、运营商能获得比以往的服务费更多的广告成分、手机厂商能以低廉的成本推出自己的手机、消费者会发现手机更便宜，交给运营商的服务费用比以往更低甚至可以不花分文——这是一种人人受益的商业模式。

事实也证明了Google的成功，在Android手机市场份额高达79.3%的今天，在开源光环和Google本身过硬的品牌形象下，Google的Play Store、Gmail、Google Map、Voice Search等已经拥有了一个惊人的用户群体；Android的手机默认搜索引擎是Google Search；Google Mobile Service成为了Android系统的核心，成千上万的Android应用基于它进行再次开发；数不清的免费Android软件中嵌入了Google提供的广告；Google甚至拥有了自己手机和平板。

不得不说，Google是一个可怕的公司，在不知不觉中，你会发现，你已经离不开它，而你也正是“Google应用——广告——广告主（商业公司）——商品——客户”这条利益链中的最后买单人。

1. **A brief summary of New Methodology**
2. Methodology was introduced to make software predictable and efficient.
3. Agile methodology: just enough process. Code oriented. Adaptive and people oriented.
4. In building business software requirements changes are the norm. After all software is supposed to be soft. If you cannot get stable requirements you cannot get a predictable plan. Instead you need a process that can give you control over an unpredictability. That's what adaptivity is all about.
5. If you expect all your developers to be plug-compatible programming units, you don't try to treat them as individuals. But for the highly creative and professional work, which I believe software development to be, this does not hold.
6. When measuring performance you have to get all the important factors under measurement. Anything that's missing has the inevitable result that the doers will alter what they do to produce the best measures, even if that clearly reduces the true effectiveness of what they do. This measurement dysfunction is the Achilles heel of measurement-based management.
7. Austin's conclusion is that you have to choose between measurement-base management and delegatory management (where the doers decide how to do the work). Measurement-based management is best suited to repetitive simple work, with low knowledge requirements and easily measured outputs - exactly the opposite of software development.
8. Technical people need guidance on the business needs. This leads to another important aspect of adaptive processes: they need very close contact with business expertise. So it's important to ensure that there is good quality business expertise that is both available to the developer and is of sufficient quality that the developer can trust them.
9. XP (Extreme Programming)

Extending their ideas of a software development approach that was both adaptive and people-oriented.