

Android 软件过程分析

Godplayer 小组

汽 03 陈吉阳 匡旭 何坦璿 黄强强

一. 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。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 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及其他领域上，如电视、数码相机、游戏机等。

2011 年第一季度，Android 在全球的市场份额首次超过塞班系统，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。

二. 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。

三. 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、Gooole 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。

四. A brief summary of New Methodology

1. Methodology was introduced to make software predictable and efficient.
2. Agile methodology: just enough process. Code oriented. Adaptive and people oriented.
3. 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.
4. 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.

5. 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.
6. Austin's conclusion is that you have to choose between measurement-based 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.
7. 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.
8. XP (Extreme Programming)

Extending their ideas of a software development approach that was

both adaptive and people-oriented.