

从工程师到架构师，Android程序员的进阶之路

2016-11-03 安卓开发精选

从第一次写出Hello World，到成为一个优秀的工程师的距离有多远？

从工程师到架构师，又需要多少技术与非技术方面的积累？

在工程师职业发展的过程中，不仅会遇到各种技术问题，也会遇到各种技术以外的项目问题。如何解决这些问题，是每一个工程师进阶之路必不可少的经历。

这篇分享来自于网易资深Android开发工程师郑文（他目前主要负责网易严选、易信公众平台、gacha二次元等产品的开发工作），在文章中，他详细阐释了不同阶段技术岗如何在项目发展的路上“升级打怪”，或许能对你有启发。

做一个产品，不可能一个人完成所有的东西，一个产品的开发到发布都是各个角色合作的。产品经理出交互，视觉来切图，开发者进行开发工作，测试做开发的测试，项目经理控制我们的整体进度和流程。

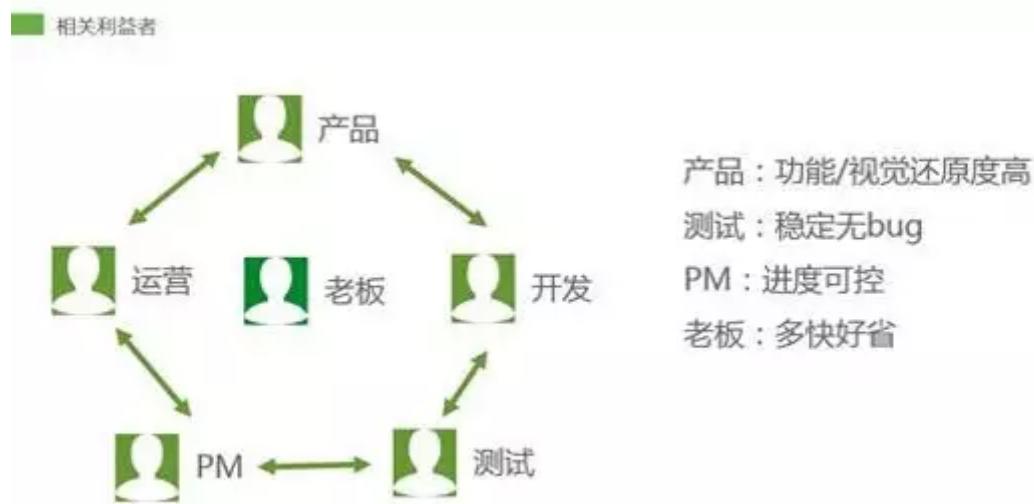
作为一个工程师，你首先需要了解各个角色关心什么。

产品和交互关心他们理想中的功能能否被正确的实现；

测试关心的是一个开发周期结束以后，提供的测试版本稳定没有bug

项目经理关心开发计划确定以后，产品迭代能否按着流程走

而我们的老板，关心的其实是投入和产出的比例。



这些当然是完全理想状态，但其实他们的需求和我们的角色是冲突的，甚至说相互之间都可能是冲突的，那作为APP的开发负责人怎么办？

1 产品从0到1阶段

如果你入职的是一个全新的团队，服务一个全新的产品。一般来说这个时候，产品首要目标是将功能做出来。这个阶段的团队可能是非常精简的，Android移动端团队最常见的规模是1-2个人。你可能需要经常加班，的确比较累。但这个阶段的收益是：你经历了一个产品从0到1的过程，你需要做技术上的选型，去做初始的代码设计，你会熟知整个测试、上线的流程是怎么样的。

在这个阶段，你的目标是尽可能输出产品的原型，让你的老板或者用户尽快看到你做的东西。所以这时做技术选型的原则主要是你自己的技术背景——你需要选择你熟悉的框架或者最主流的框架，不要轻易尝试一些新的框架，万一踩到坑很难跳出来。

虽然此时我们的关注点主要在业务上，但是必须还是有一些工作必须做的：

1. 规范代码：团队内部的代码规范必须一致的，比如说命名，同时也需要把视觉的风格给确定下来，如：通用边距是多少，一级标题的字号是多少；

2. 版本管理：即使是初始版本，也需要做好版本的管理，否则后期的版本管理会不得不做很多兼容的方案。我认为必须的版本管理有以下几个：

应用的版本

应用启动的时候，需要在后台有一个版本的检查机制。之所以在第一个版本就必须有，是因为在非常特殊的情况下，比如说线上app有重大bug，或者说有重大的业务调整，我们可以强制下线有些非常老的版本。

协议的通信版本

你和服务端之间的协议版本需要在协议参数中体现，这样可以避免每次的版本升级都产生一个新的url；

3. 做好代码的管理工作：我们需要抽象出主分支，开发分支，还有各种版本的tag分支。保持你的代码主干是处于随时可以发布的状态。同时，为了保持任何一个发布的版本是可追踪的，你需要每发一个版本就有一个对应的代码分支，这样发生了线上的崩溃以后才能映射会我们的代码。

4. 要做好线上崩溃的收集机制：友盟2015年的移动崩溃报告，日活一万以下的app崩溃率是6%，日活在10万甚至100万级别的产品，崩溃率基本在3%，但这个对于商业的app，这个app崩溃率是远不合格。起码应该到1%一下，我们内部的app，除了新版本发布的时候，基本是需要稳定到千分之五以下。

2 产品进入版本迭代阶段

产品正式发布以后，如果反响还不错，公司继续在项目上投入。这个阶段，你的业务方向其实已经确定了，所以你需要根据业务情况做一些技术上的调整。

这个阶段，适配性问题开始铺开，不一定来自线上的崩溃，也有可能是来自用户的反馈。**用户反馈的优先级当然是比较高的，但此时如果一味的满足用户的反馈，很可能会疲于应付各种bug，你的正常功能开发收到影响。**所以，这时候，你应该建立bug分级制度，根据之前我们线上做的app崩溃比率，以及用户反馈的权重。确定哪些bug应该立即修复，哪些应该是稍微延后一点。优先级高的bug，我们就可能自己发个小版本先出去，优先级低的bug，我们可以延后到下个版本去解决。

随着版本的迭代，我们可以做的东西其实很多。

第一个，我们之前的编码规范和ui规范必须文档化；

第二个，如果某些项目中的框架是通用的，类似网络库，文件存储库，你把抽离出来，单独成为一个module工程，或者说输出个sdk。这一步其实很重要，但我发现很多团队都没做。

第三个，关注android新技术，合理判定某些新框架或新技术能否解决你的业务痛点。

现在获取新技术渠道很多，但我想强调的是，**新技术关注肯定要做，这是技术热情的一部分，但需要建立你自己的技术观。**而技术观的建立是你对这个东西充分了解之后而建立的。我最近也遇到一个面试候选人，聊到新技术的时候，发现他对大部分社区内比较火的技术都比较了解，做过一些简单的demo，也在原来的团队内部做了一些分享。我们当然很喜欢这一类的候选人。但进一步聊，发现他反馈的都是互联网上一些通用的评论，或者是解决方案，基本上就是人云亦云，没有形成自己独立的判断。所以，我的观点是在项目中盲目去引入rn框架不太合适，当然，这是我自己的观点，欢迎拍砖，欢迎不同的观点，但不应该没有观点。

最后，还可以对建立团队氛围。譬如你可以通过制度化分享的机制，或者输出技术文章等，督促团队去主动去学习新的知识，提高整个技术团队的水平。

3 总结

技术是为产品服务的，并不是说出现需求才考虑技术，在需求出现之前，技术可以同步给产品经理，告诉产品经理我们可以做什么，这样产品策划同时可以根据技术去拓展业务，而不仅仅说技术不拖业务后腿，技术人员也可以达到跟APP共同成长。

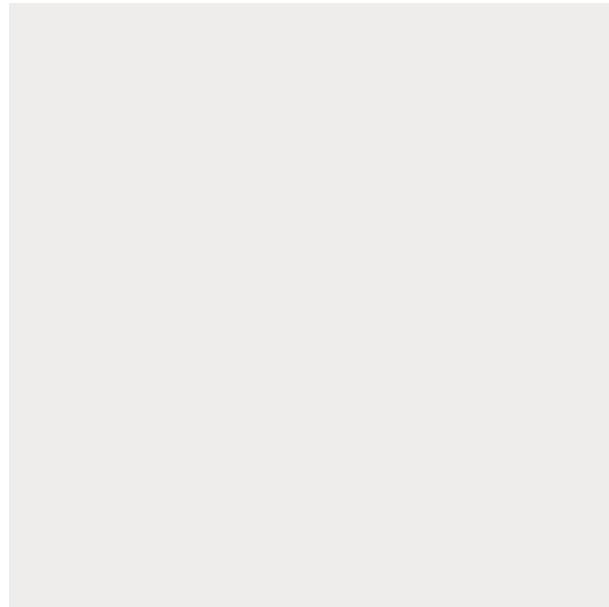
接下来的两周里，我们邀请了业界优秀的Android开发工程师，与大家分享工作和学习经验。

免费直播列表

直播主题：安卓面试重点知识及学习方式
分享人：周江华 网易云信/七鱼Android负责人
分享时间：11月7日 20:00-21:00

直播主题：Android开发入门指导
分享人：梁建 网易云音乐Android主程
分享时间：11月15日 20:00-21:00

如果想要第一时间获得以上免费直播课程的收听地址，欢迎大家扫码加入网易产品移动开发交流QQ群。



此外，网易云课堂的《Android开发工程师》微专业即将开启双11团购活动，**从11月8日开始，支付1元加入学习团，就可以享受学费最低价。**

这两门微专业来自网易邮箱、云音乐、考拉海购等产品的资深开发工程师亲授，结合自身实践开发经验及产品案例，帮助学员完整体验一个项目的开发流程并掌握其中技能要点，真正做到学以致用。
