

# 程序员业余时间修炼指南

本文主要与大家分享我业余时间如何学习多门技术以及开发业余项目的一些心得，分为 初衷、硬技能、软技能、第一生产力、环境、时间几个部分，阅读大概需要 10 分钟。

关于我之前做的一些业余项目，我在知乎上的一个回答大概可以总结：[小芋头君：作为程序员，你有哪些正在做的个人项目？](#)

## 我的学习方法

在开始介绍我开发业余项目的经验前，先和大家分享一下我的学习方法吧，不过我认为学习这件事情因人而异，每个人都有适合自己的学习方式，所以这是作为一个参考，希望大家也都能先找到适合自己的学习方法。

我在学习一门技术的时候，通常都是直接切入实践，曾经我也尝试先看各种基础的语法或者稍微深入的一些设计模式之类的书籍，但是发现这种方式对我来说很低效，例如 Objective-c 的学习，当初我买了几本书，啃了半年还是一知半解，当然另外还有一个关键点就是要时间连续，如果每天看几页的话，也是低效的一个重要原因，因为程序的语法是偏生涩的，死记硬背很难记住所有知识点。

所以后来，在有一次 职之后，我直接接了一个 iOS 的外包，那时候我甚至还从来没有在 xcode 上写过 iOS 代码，我逼自己花一个月时间，掌握并且做出一个 app 来，然后在这一个月时间里，我将大量的时间投入了 iOS 开发的学习，并且尽量采取高效的学习路径，我来不及看明白所有的基础 OC 语法，就直接去看讲述 iOS 编程的文章（苹果官方有很多实践入门的文档），然后直接实践，在 xcode 上敲代码做布局，然后运行看效果，对于不懂如何实现的功能，就直接去 google 搜索文章，快速查看别人的方法，然后实

单，于是就想把自己的博客搞成一个社区，大家可以来这里发表博客，或者是问答，还有论坛发帖等功能，整个网站的结构变得复杂了，对程序的结构设计，数据库设计的要求也就凸显了，通过这个网站的不断迭代，我对 nodejs 的了解也越来越深入。

我的主要技能仍然是前端开发，这个是我的工作。然后其他开发技能基本都是在业余时间学习与实践的，包括其他一些软技能，例如设计，产品，运营之类。方法无非就是我上面讲的，所以接下来讲的业余开发项目的感受，这些项目，其实最终的初衷除了创造之外，还有很重要的一个点，就是我用来自学习技术的实践，不管你的产品做成功还是失败了，至少我的技术长进了。所以，不要在业余时间做一些没有技术含量的事情，例如接一些切图的外包或者你已经很熟悉的事情，多给自己一些挑战自己的事情。

## 业余开发项目的初衷

总体来说，在我的感觉里，国内的环境对于业余开发者来说其实是非常不友好的，不过作为一个开发者，一味抱怨并不会有大的改观，能做的就是不忘初心，多做尝试多做积累。包括我之前也和一些朋友深入聊过类似的话题，例如“在行”上约我聊相关话题的同学，我其实是鼓励大家勇敢的去做自由职业者或者业余开发者，虽然环境的确不是很友好，但是年轻时多尝试总是没错的，毕竟我们还有资本去寻找更多可能性，而且我相信每个开发者其实内心都是向往利用自己的力量改变人们的生活方式的，而我们通常做的事情却总是一个大工程中的螺丝钉，时间长了失却了创新的初衷意味，这时候离一个创造者越来越远，自会觉得内心恐慌，我最初的爱去哪了？虽然这些大工程的确在不断改变我们的生活，但是我的初衷却是能够主导一场改变，即使它可能改变的只是一小群人的某个小小的生活方式。

我一直觉得程序员是一个很神奇的职业，不是说这个职业本身神奇，而是代码赋予我们的能力神奇，我们可以凭空创建起一个虚拟的世界，可以制定这个虚拟世界的规则，可以让这个世界按照我们的想法运作，但是这个世界也会反过来要求你改变你，因为一个正常运作的虚拟世界需要严谨的规则，这就需要有设计产品，设计合理规则，设计运作闭环的能力。这也是我们重点要强调的业余开发者的能力模型，你不能只是拥有 coding 的能力，不能只是 create，而是要合理的创造，探索人群的新的需求，创造真正有用的产品。

外，还远远不够，平台差异，平台特性，版本兼容，自适应，还有最麻烦的一环，你的 app 开发成功之后如何发布到对应的平台？如何设置内购等？特别是 iOS，发布一个 app 的成本远比想象的高。另外与端相关的，还有基础的数据统计，消息推送，分享集成等，在初版的时候最好就能将这些工作做好，而这就需要你了解这些功能的集成开发。

2. 前端开发，故事上传有一个后台，当然这个很简单，然后故事从 app 里分享出去之后有一个 H5 页面，这个页面里可以直接播放音频，然后有一些下载 app 的引导。所以前端开发的技能也是必备的。还有就是一些平常要用的宣传页之类的。
3. 服务端，所有故事都是在后台存储的，需要数据库，还有服务端应用提供数据分发，以及一些数据统计之类。
4. 数据记录与分析，我需要在用户的使用过程中，记录用户的动作和其他有用信息到服务端，然后以这些数据分析用户行为，以便决定后续哪些故事是最受欢迎的，以及受谁的欢迎，我可以做定向的推送或者收费服务。
5. 设计能力，说起来这也是非常核心的需要，如果没有基础的设计能力，很难做出一个让大家眼前一亮至少看着舒服的 app，特别是现在市面上的主流 app 都拥有比较高的体验门槛，粗制滥造的 app 很难吸引用户，在这点上，也需要做一些刻意的培养，虽然我们可能没有受过专业的训练，但是可以多看看多想，从别人的设计风格中借鉴，慢慢培养审美，当然还有最重要的就是设计工具当然需要熟练掌握。

除了技术之外，还需要其他一些技术之外的能力，这些能力可能是区分你与其他程序员的重要分水岭，它决定了你可以创造产品，而不是掌握了哪些具体的技术。

## 软技能

我们简单谈谈：

1. 需求分析，产品规划。要做一个什么产品，不是因为我掌握了什么技术，能够写出什么样的代码来，技术只是一个工具，要发挥其价值，必然是需要有原始的需求和

方面的能力，我推荐大家首先多去尝试挖掘，另外就是在平常公司的工作中多多留意大家的惯用手段，多向相应职位的同事学习，另外在职场中遇到类似问题的时候也多从自身思考，如果让你来推广一个 app，你有什么办法？具体如何推广 app，这里也可以分享一下。首先，你要保证你的 app 是优秀的，否则你把 app 推给用户，他不愿意下载，那无论如何推广都是无效的。然后推广渠道，可以找一些免费渠道，一些 app 推荐的网站、app、公众号，想办法联系到他们的编辑，然后请他们推荐你的应用，如果你的应用足够优秀，他们会很乐意帮忙，这种渠道还是挺多的，例如最美应用、少数派、appso 之类。另外一个渠道就是一些垂直的论坛和社交网站，甚至是知乎之类。然后所谓的运营，最基础的，在 app 内要有一些与用户的互动，能够维持用户的使用激情，甚至帮你扩散 app 到朋友圈，具体可以是一些活动的形式（运营管用手段）。最后再说一点，适当的时候，要舍得花钱，但是要花在点子上，可以搞一些送礼物的活动，也可以做一些付费的推广配合冲一下榜。

## 创新是第一生产力

当我们知道构建一个产品需要些什么基础技能之后，那我们如下真正的下手做一个产品出来呢？我应该做什么类型的产品？我能想到的产品别人都做过了怎么办？我人手不足无法开展大型工程如何与别人竞争？我的推广和运营能力有限没人用怎么办？

有时候我也会困惑，特别是我做了一个 app 出来，然后千辛万苦推广的时候，要么发现对个人来说推广一个 app 是极其困难的，要么就是发现当我把 app 推到用户面前的时候，用户并不会很兴奋的接受你的产品，这时候我就会开始思考，如何突破？是寻求更多曝光渠道？还是做一些很花哨很吸引人的运营活动？感觉这些事情对我来说都没有那么简单，成本也都很高，越想越头疼，越想越会觉得自己做个事情出来真的不容易。

不过，可能我想着突破的点的方向就是错的。业余开发者的核心竞争力是什么？不是你的资源，不是你花里胡哨哗众取宠的玩法，而是你个人意志主导的创新能力！用户对于创新功能（当然首先要是合理的）的敏感度是很高的，如果你的 app 提供了一种前所未见但是又能够解决问题的使用方式，这才是你的产品的核心竞争力，在极致情况下，可能不需要任何主动地推广、运营，用户会自发帮你传播，并且保持很高的活跃度。当然，这种创新，不是特指一项具体的技术实现，也不是指一个花里胡哨的设计，需要自

难接受的事情。当然，这是大环境，你完全可以跳脱这些束缚，但是大环境对人的影响还是非常大的，你很难摆脱正常人的生活方式与状态，这些都在暗示你，从事不稳定收入的自由职业独立开发不是一个好的选择。还有其他一些问题，例如社保，五险一金，户口，签证等问题，如果你没有正式的工作，这些事情都会变得很麻烦。

其实我非常向往自由职业，所以曾经有一年我没有工作，在家里尝试了一年时间，除了自己的自制力问题之外，我也深知，我没法维持“自由”这个状态太久（除非我一夜成功，但是那时候的我，也并非“自由”了），因为作为一个正常人，我需要承担很多这个社会需要我承担但是我自己可能并不特别乐意的责任。当然还有一点，我需要回到职场观察学习，如何成为一个“正常人”，如何创造“大型产品”，虽然可能这并非我的初衷，但是我们并非天才，做任何事情都需要这个世界的辅助，所以我们需要很好地和这个世界沟通。

在这种大环境下，除了鼓励更多的人参与到自由职业和独立开发的队伍中来之外，最好的选择就是做部分妥协，但是勿忘初心，利用工作之余，拾起你对技术和产品的热爱，慢慢打磨一个业余的产品，通过这个打磨的过程，积累经验（硬技能、软技能），同时保持创造的活跃度。

## 时间

接下来探讨的这个话题可以涵盖好几个问题，例如如何一个人在短时间内掌握多项相关的技能？如何维持自己业余开发的热情？觉得技术一直没有长进怎么办？

除了 0.1% 的悟性、1% 的方法之外，你能做到别人做不到的事情的最大秘诀就是：时间。

如果你真的对一个事情感兴趣，那就投入 100% 的热情和精力，例如每天下班后 7 点到晚上 2 点（当然不鼓励熬夜）的时间，就是 7 个小时，你一天投入到工作学习的时间比别人多了 7 个小时，一个月就是 200 个小时，加上周末的时间，远比你正式工作的时间还要长，这些时间才是人和人之间拉开差距的重点。

例如学习 iOS 开发，一天学一个知识点（加上实践），我觉得最多一个月就可以覆盖大

讲了这么多，不知道对大家会不会有帮助，不论如何，最重要的还是开始执行，脑海里总会冒出一些想法吧，不知道是不是合理？那就先尝试做，慢慢的就会发现什么是合理的，什么是不合理的，每个人都有这样的一个过程。然后多去观察生活中出现的大小问题，总有我力所能及可以用技术去解决的问题，而这可能就是一个很好的 idea。另外希望真正热爱技术和创造的朋友可以一直保有自己的那份初心。

# GitChat