

程序员生存指南：我的成长实践

到去年年底，和工作了三年的公司解除合同，算算我在软件开发这行已经干了超过10年了。离职后，我计划用三年的时间读书，先泛读再精读，会读到什么书，我也不确定。当然读书变成了我的主业，并不等于我不用赚钱养家，我一样需要承担家庭的责任，只是家人会理解和支持我。只要家里还能揭的开锅，家庭的生活质量不严重下降，这三年我可以少赚点，把精力放在我想做的事情上面。

就这样，我变成了一个自由职业者。当然，我不是要鼓励大家都放弃工作，做自由职业或者创业。有时候，你们看似羡慕的结果，往往是当事人不得不做的选择。于我也是一样。



人的每个阶段都有每个阶段要做的事情和要修的“课程”，别人代替不了你去上课，你也无法领会别人给你的关于这门课程的考试答案。但我还是想把自己的课堂笔记分享给大家，希望走到类似我这个阶段，或者对这种体验感兴趣的朋友，可以从我的一些经验、方法和心得中获得启发。

15年前

15年前我从广西的一个贫困县城只身来到成都上大学，我虽算不上家境贫寒，但也非富裕家庭出身，所以当时上大学时也想做兼职为家里减轻一些负担。

还好，我比较理解赚钱的关键是：交换！但那时，我能交换的是什么呢？

在15年前，网络并没有现在发达，并不是所有人都“在线上”。和大多数考上大学的莘莘学子一样，我想到的就是通过做家教赚点钱，交换一下自己的高中学习经验。不过，最终我没有因为家教赚到一分钱，相返还被骗过两次。一次逛街时有个人跑出来就说我才高八斗是个做家教的奇才，我竟然信了，就交了中介费；还有一次在学校看到招家教的广告，我去交了中介费，待我第二天再去时“中介们”已经人去楼空。

只能怪自己“So young, So simple”，并由衷的感叹“城里人套路真深”。

虽然还有很多可以做兼职的活，如发传单，到电脑城打工这类的，但相对于那时还比较木纳的我来说，似乎很难找到可以胜任的，或者我都没有途径找到。

生活中总会有窗子给你开着的，还好在计算机和网络的世界里，交流的规则就比较简单直接。

我的第一份兼职

在计算机的世界里，虽然我的起步比较晚，但它是公平的，你努力总会有些回报。而且，学习计算机编程是成本最低回报最直接的，当然，我那时并不想这些，只是觉得自己喜欢而已。

我大一时就自学去考了计算机等级，然后开始鼓捣一些系统的漏洞和安全方面的问题，估计是控制欲在起作用，总想黑一下学校的服务器，当然我也做到了。然后开始在一些黑客杂志上投稿，在系里面小有名气（我们是数学系，不搞计算机）。

第一个“私活”是接了同寝室的一个同学的，给他的爸爸的厂子建一个宣传网站。那时候会建网站就很了不得了，别人都要以高手称呼你的，后来和同事们聊天，他们说那个时候在上海建这样的网站都是2万起。不过我完全不知道怎么谈价钱，特别是和认得的人谈钱更是羞于启齿，最后一身汗说完发现我只和别人要了域名和空间的费用，这个项目没赚到一分钱。

而我大学里的零花钱主要是靠稿费。其间也接过几个编程的活，不过金额都不大，因为那时我在“Delphi盒子”（一个Delphi编程相关的网站）上发过一份远程控制的源码，所以经常有人会来找我。实践证明，要让别人能找到你，你得先给别人找你的理由和途径。

大学期间，比较正式的兼职是在一家黑客杂志做编辑（在线的），主编每个月给我500元，说其他的在线编辑都没有，我不知道是真是假。

工作后的兼职

工作后我有意往移动端开发靠，所以第一年做了J2EE后，就从手机游戏开始进入到移动开发的领域。直到Android的风起，也是软件外包比较火的时间，那几年我接了很多的Android项目，也给别人介绍了很多项目。有按整个项目算钱的，也有按小时算钱的（和客户开会也计费）。

带我入行的人看中了我“不看重钱”，刚开始做Android的工作时，接私活锻炼技能的价值很大，私活的经济回报对初出茅庐的我并不重要。当我在Android这个领域沉淀了几年后，私活带给我的性价比就不高了。

我遇到了很多机会，每一次兼职每一个客户，都是一次机会。而我，大多时候都没有做好，不懂得“把自己当成一家公司来运营”，不知道要经营好自己的“个人品牌”。

接私活这类兼职方式，本质上就是让你用现有的时间去交换金钱，虽然可能数额还可以，但你单位时间的价格并不会提高多少。而且你一天只有24个小时，你能用多少时间去兼职呢？

本质上我并没有改变：我一直是一个出卖时间的人。

培训和咨询

最近这两年，我渐渐意识到一个程序员，除了出卖劳动时间，还有很多方式可以赚钱，但那些行动往往需要一定时间的积累才会有成果。所以我定了3到5年的计划，但在积累没到之前，我总不能坐吃山空吧？

既然我现在只能出卖时间，那么怎样才能让时间更值钱呢？或者说要如何提高我的单位时间的收入水平呢？

只要你准备好，命运（或者说运气）随时可能会来光顾你。在一些技术群里讨论一些Android技术问题时，我获得了一个做讲师的机会，得意于我多年在公司内部做分享，和这几年口才和演讲能力的锻炼（平时吹牛的水平那是练得杠杠的），我顺利完成了第一次培训课程，而且获得了好评和不错的收益。

重要的是，我喜欢这个过程，我享受分享，虽然我做讲师的第一天讲课后嗓子差点哑掉。除了喜欢和享受外，做培训还有一个关键：做培训的准备和课件是可复制的，第二次再讲类似的课程时我便轻车熟路了。

你好好对你的客户（同事、认识的人，付钱给你的客户），他们也会好好对你的。后来我还到一些公司去做顾问（技术咨询），主要帮助他们解决一些性能和他们遇到的不好解决的问题，也都是认识的客户介绍的。

都是做开发，别人不好解决，你就能解决吗？这就是做程序员和咨询师的区别：咨询师不一定技术要很厉害，但思维方式要有不一样的地方。这样你才能想到客户的开发工程师不能或者不敢想到的点上，用内行一点的话来说咨询师要学会系统思维。

斜杠青年

当然，不管我满不满足于现状，我都想要改变只靠出卖时间赚钱的方式，或者说是探寻财务自由之路，俗称“睡后收入”，我即使不干活了我的输出或者产品还能给我持续创

造收益。

而培训和咨询与在公司上班的区别不过是：一个是将时间只卖给一个客户（在公司上班），一个可以选择卖给不同的客户（独立咨询师）。并不是我最终的追求的工作方式。

经过我的研究和一些尝试，我发现了一个词：“斜杠青年”。

“斜杠青年”最近2年才在国内流行，源头则是在10年前的纽约。《纽约时报》专栏作家 Marci Alboher 注意到这种现象，并写了一本书——《多重职业》，书中称为“The Slash Effect”（斜杠效用），后来在国内被翻译成了“斜杠青年”。

根据 Marci 的本意，斜杠的标准应该是“拥有2个以上能被他人辨识的技能优势”，即这个技能至少能达到最低的职业交付水平，也就是能收到钱，最好能达到专业水平，与专业选手有一拼。

Marci 将斜杠青年模式进一步总结为以下五种：

1. 稳定收入+兴趣爱好的组合。
2. 左脑+右脑组合。
3. 大脑+身体组合。
4. 写作、教学、演讲、顾问组合。
5. 一项工作多项职能型。

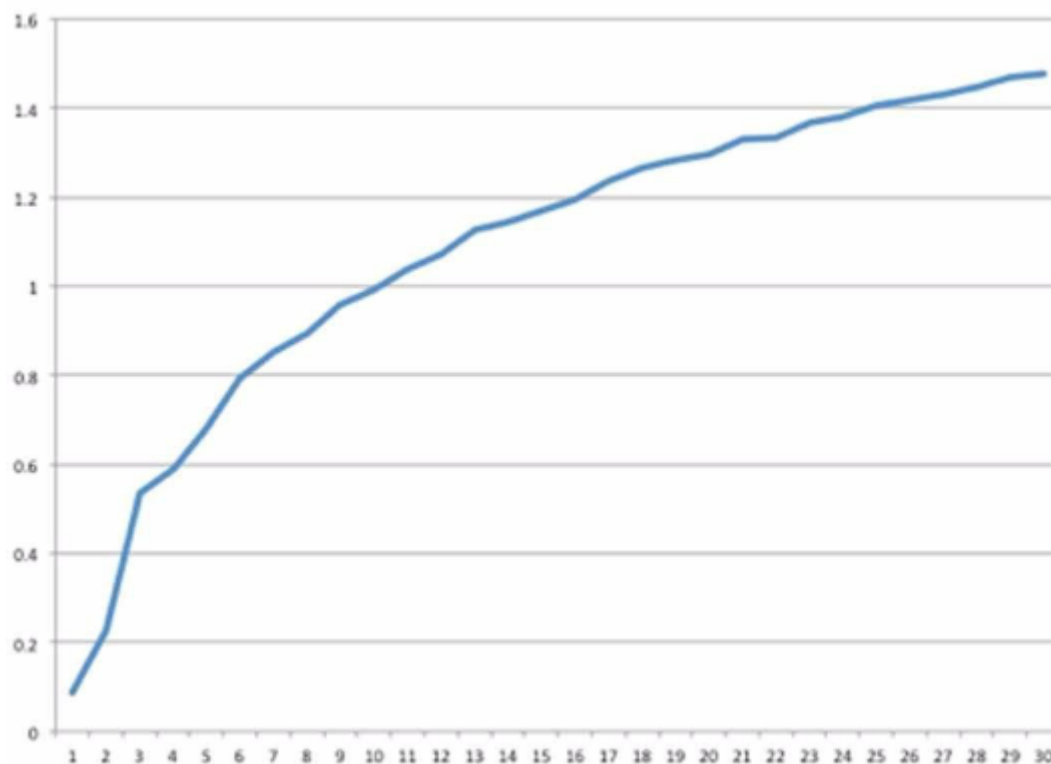
每个程序员都有各自的优势和局限，你可以自己去思考和尝试这些模式。第四种模式可谓是一个黄金组合，也是 Marci 自己的斜杠模式。这四种身份之间可以形成完美的循环推动，因为写作可以让你成为某个领域的意见领袖，演讲的邀约也会随之出现，等到经验足够的时候又可以开展教学和顾问工作。

编码之外的能力

作为程序员，并不是你在 Coding 的时间越久你的编程技能就越强。而且，现实也会给你教训，你很容易遇到编程的能力不如你，但收入却比你高很多的人。即使编程的能力再好，你也会遇到一个职位和收入的天花板。

在技能水平的成长上有两种曲线，表示花费的时间和成果之间的关系。

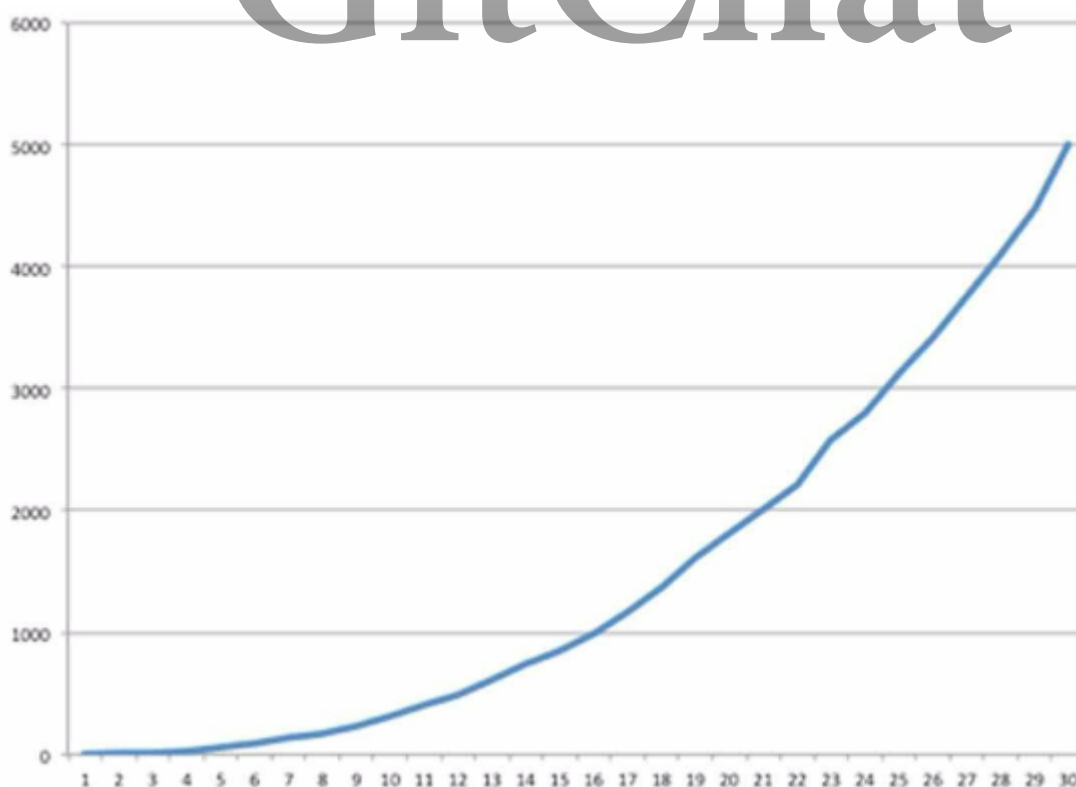
对数增长



如学习一门新的编程语言，在初期时进步非常快，到后面则越来越慢，最后几乎都不怎么增长了。在这个阶段，哪怕你付出极大的努力，也只能获得一点小小的突破。

指数增长

GitChat



这种增长情况和对数曲线相反，在一段很长的时间里，虽然你一直勤加练习，但技能增长并不明显。然而突然到达某个阈值后，好像突破了障碍一样，水平一下子就显现出来，而且后边越增越快。

根据上图所示的两条曲线，你应该明白对于大多数人和行业来说，花点精力在业务领域之外是很有必要的。说的直白一些，大家都是程序员，都会编程，你的编程水平又能高我多少呢？如果我们只比编程技能，那么这个技能在学到90分以后，大家的区别就很少了，而后边你每提高一个点的技能值都需要付出很艰辛的努力。正如体育运动一样，世界冠军就那么几个付出了超凡努力的人有能力争取。

编程技能的提升是一个对数曲线的增长模型。

而对于大多数一般程序员来说，在编程这个维度外发展几个其他维度的能力就尤为重要。可能我的编程技能只有75分，但已经够用了，你的编程技能有90分，但再提高也很困难了。然而我可以加上两项都是70分的其他技能，我的综合实力就变成了： $75 \times 70 \times 70 = 367500$ ，这和单一编程的90分已经不是一个数量级了。

所以花点精力在编码之外极其重要。提升自己在编码之外的软技能，让自己具备多维度的竞争能力。

OK，现在问题来了，程序员需要提升哪方面的能力呢？

沟通能力？

学习能力？

领导能力？

运营能力？

.....

也许你还可以总结出很多能力，但有两种能力非常关键：演讲和写作。用通俗一点的话来说就是：“未来属于会说和会写的人”，其实和Marcy提到斜杆青年的第四种模式是一个意思。

显然并不是人人都这样认为，或者适合这两样技能。《哈佛商业评论》说这是网络个人经济的开始：“新经济的单位不是企业而是个体”。未来是一个“个体崛起”的时代，个体的能力也越来越有发挥的空间，个体崛起的时代，并不是“每一个”个体崛起的时代，而是少数个体崛起的时代。

不妨思考和开始尝试一下演讲和写作，这是运营和提升自己的一种很好且有效的方式。而且演讲和写作是指数曲线的增长模型，需要你及早做准备，而且开始你可能看不到希望，但一定要坚持到指数开始突破。

如何用自己的能力赚钱

想要利用自己的能力赚钱，你首先要将自己的能力变成可输出的产品！

我们常常会鄙视某某人的网络课程、或者出的书多么的没有技术含量，太初级，完全不屑一顾，总想着自己能做出更好的，只是没有时间做而已。就和我们在工作中也会鄙视别人的工作一样，总觉得别人做出来的东西一般般。

然而，不要忽视了，不管这些人的产品质量如何，他们的产出都是可以立即进行交换的商品，是一个完整的成果。而你心目中认为自己很好很有技术含量的东西，它不是！别人无法获取和使用你所谓的很棒的东西，即使你实现了它一些功能，那也只是一个半成品而已。

用做IT培训师例子说明一下：作为一个程序员，我有Android的编程能力，那如何把自己的编码能力转化成可变现的产品？

首先并不是你说你能你就能！如果客户没有和你合作过，不是你把你的简历一摆别人就能认可你，好比找工作一样，还需要经过面试和考核。即使你口才再好，也需要有输出的东西让客户先看到，这就需要你编程和写作的能力（写PPT并不比写文章简单）。而通过考核最好的方式就是：现成的成功案例。

那么，第一次做没有案例怎么办？那就拿出现成的课件和培训的练习设计，将你编码的能力转化成演示和设计的练习Demo，直接演示给别人看，没有什么比让客户直接感受成品更有说服力了。而你的演讲能力也是你能否成功获得这个机会的关键，程序员里有太多编码很厉害的家伙，但不会说也不知道怎么写。

说了这么多，其实总结起来就一句话：把自己的能力产品化。而且最好把这个转化的过程放在平时的工作和学习中。

如果你在平时工作时就想到这些，在遇到问题或者BUG时就有意的记录下来，做成可分享的PPT，那么当你真正要把你的编码能力转化成产品时，你会发现过程一点都不痛苦。

当然每个人的情况不一样，很难有一个大家都能成功的方法，但却可以分享几个问题给大家，持续地问这三个问题，让你做事情更有价值，而且容易学会以终为始的成果导向思维。

三个问题如下：

1. 我做这件事，结果是什么？
2. 这个结果对谁有价值？做到什么程度对方才觉得这个结果是可交付的？
3. 如何让这个结果易于复制，可持续下去？

最后

从一个程序员到现在独立的咨询师，目前的状态对于我仍然是一个过渡状态，终极的目标是要不断的成长。而在这个过程中，我们无法忽视生存问题，所以我们需要学会如何利用自己的优势或者能力，并转换成可进行交换的产品。

在追求“诗和远方”的前提是你先跨过“眼前的苟且”。

留一些不便书面说的话题到线上再和大加一起交流，当然到时大家也可以提自己感兴趣的问题。

如：接私活怎么定价？

做讲师和咨询师怎么谈价钱？

怎么找活？

做个自由职业者要面对的问题等。

GitChat