

54讲侠客行：一技压身，天下行走



从今天开始，我们进入了专栏的第 5 部分 —— **寻路：路在何方**？这是一条关于方向、角色和自我定位的探索，那就让我们开始一起走走这条程序江湖路吧。

大约三年前吧，读到一篇文章《为何我工作十年，内心仍无比恐慌》，来自一位腾讯产品总监的演讲分享。文中分析了一个让其感到恐慌与焦虑的深层次原因：好像不会什么技能，技能门槛低。

这种恐慌和焦虑感在这个行业中普遍存在，不止于产品经理，程序员也一样。一些传统行业的生命已经远超过一个人的寿命，而 IT 互联网行业还不满三十岁，也许正是因其还很年轻，生命力旺盛，远超传统行业的发展速度和新陈代谢规律，让其中的从业者深感疲惫，同时对未来又充满了不确定性，而未来的不确定性通常正是让我们感到焦虑的一个主要原因。

门槛

技能的门槛高低，决定了让我们产生恐慌和焦虑的水位线。

在前面提到的《恐慌》一文中说，产品的从业门槛足够低，作者十年的从业经历中见过从事产品的人来自各种专业，还有各种改行做产品的。而从业门槛主要来自于技能门槛，特别是硬技能，硬技能属于行业的专有技能，需要足够的时间积累，通常这个积累时间就是你可能熟悉的理论值：一万小时。

产品看起来是一个缺乏硬技能门槛的职业，因而感觉门槛低。而程序员职业其实是有一定硬技能门槛的，但这种门槛随着技术和工具的进步正在变得越来越低。如今 IT 互联网行业当然是繁荣的，繁荣的行业带来利差，自会吸引大量其他行业的从业者进入，而这些进入者自然会选择门槛低的职业工种来跨越边界。

在其他行业干了些年头的人，有些可以在这个“互联网+”的时代通过垂直行业专家来进入互联网行业，但要进入程序员这个职业就得赶早了，毕竟硬技能需要的积累时间是很难省却得了的。大部分人都是在大学期间或刚毕业不久就完成了转行到程序员职业的切换，如我的一个高中同学，她本是文科专业中文系的，大二就毅然开始辅修计算机的第二学位了。

还有个行业一直繁荣，需求永续存在而且供不应求，但却从没见过任何其他行业的人进入。我说的就是“医生”这个职业，它的

硬技能门槛之高不免让人联想起《冰与火之歌》里的绝境长城，让人完全兴不起翻越的欲望。我听说过小说写得好的前妇产科医生，却没听说过手术做得好的前小说家。

医学院的学生本科都要比其他专业多读一年，但本科毕业可能都找不到什么好工作，至少要读到硕士，想有点发展还得读博，十年一晃而过。而本科毕业的程序员，一进入 IT 互联网行业可能拿的工资比医学博士生刚进入医院还高，这就是行业繁荣的好处。但坏处是，这个行业变化太快，有时你没什么错，只是因为老了。很多互联网公司喜欢年轻人，标榜年轻，员工平均年龄二十多，所以才能最懂年轻人。

而医生呢？这么说吧，你是喜欢年轻有激情的医生，还是经验老道的中年“老”医生？

程序员看似是很有技术含量的硬技能门槛，实际远不如医生这个千年来的“古老”职业，行业的最低技能门槛要求挡不住很多人热情地涌入，而技能成长的天花板也感觉并不高，如何能不恐慌与焦虑？

模型

有时可能我们会有一个职业理想，叫“一技压身，天下行走”，就像一名侠客一样，学好了功夫，从此闯荡江湖，好不逍遥自在。

之前看过一本武侠玄幻小说，里面有一些角色就叫“天下行走”，他们都有自己厉害的独门绝技，不厉害怎能天下行走。其中，剑客的剑快，野人的身体坚硬如铁，和尚从不说话修的闭口禅，一开口就人人色变，这些就是他们独特的技能模型。

技能模型才是区分不同专业人才特点和价值的核心关键点。

而技能模型的形成是一系列选择的结果。以前玩过一个游戏叫《暗黑破坏神》，正常不作弊地玩，一个角色是很难点亮所有技能的，游戏是故意这样设计的。所以你可以反复玩来尝试点亮不同的技能组合方式，这样游戏才具备反复的可玩性。而与游戏不同的是，人生只有一次，你无法点亮所有技能，只有唯一的一种点亮路径选择塑造独一无二的你。

而这种选择，可能一开始是无意的，比如我成为一名 Java 程序员是偶然的，而你成为一名 C++ 程序员也可能是偶然的，早期的技能点亮策略有很多的偶然性。但到了后期，我们逐渐成长，有了更多的经验和选择权，这时就需要主动选择去建立自己的技能模型。

记得有一篇关于工程师思维的文章是这么说的：

工程师思维的大道，就是先创造一个好模型，然后想办法实现这个模型，工程师关心的是能不能用这个模型创造出东西来。

而技能模型其实正是工程师创造的第一个元模型，这个模型决定了后续作为工程师的你还能基于此创造怎样的模型，从而完成产品的实现。

当只拥有一些零散的技能点，而且这些技能点还会随着时间流逝而过时，我们当然会感到恐慌与焦虑；但如果能够将这些技能点组合成我们独有的技能模型，提供独特的价值，从此独步江湖，甚至开宗立派，想必也就没那么恐慌与焦虑了。

以前文章写过关于“知识体系”的内容，那它和技能模型有什么区别？知识体系本质也是一种知识模型，但技能模型更深一个层次，因为技能是对知识的应用。知识模型构筑了理论边界，技能模型是实践的路径。

路径

那么，关于技能模型这条实践路径该如何去选择和构建呢？

程序员作为工程师的一种，必须得有一项核心硬技能，这是需要长时间积累和磨练的技能，要花大力气的，而这个大力气和长时间，也正是这门技能的门槛。关于技能的习得有一个流行的看法是：花 20% 的时间快速获得某个领域 80% 的知识和技能。这看起来像是一种学习的捷径，但一个硬技能领域最核心的竞争力往往都是最后那 20% —— 也就是你用那 80% 的功夫反复磨练出来的最后 20% 的技艺。

古龙小说中有个角色叫荆无命，他腰带右边插着一柄剑，剑柄向左，是个左撇子，江湖中都知道他左手剑快，但其实他右手剑更快。荆无命要是个程序员的话，那可能就同时具备了两个核心硬技能，属于那种 Java 很强，但 C++ 更牛的人。但我从业这些年还没碰到过同时点亮两者的，无论 Java 还是 C++，因为各自都有足够大的生态和体系，已经需要很长的时间来积累和打磨了。

我们大部分普通人，拥有的是有限的时间与才华，面对的是无限的兴趣和技能，同时修炼多个核心硬技能是不明智，甚至是不可行的。记得以前读万维钢有篇文章介绍了一本书叫《达芬奇诅咒》，文艺复兴时期的达芬奇是一位多才多艺的人，但一个人如果像达芬奇一样对什么东西都感兴趣，但又没有和达芬奇匹敌的才华，很可能尝试了很多，最终却一事无成，这就中了“达芬奇诅咒”。

所以，构建核心技能模型其实是关于才华和技能的战略。《达芬奇诅咒》一书作者就选择技能领域推荐了三个标准：

1. 你确实喜欢
2. 你在这个领域有天赋
3. 这个领域能挣到钱

我仔细回味了下这三个标准，真是很接地气，实在可行。你喜欢的领域，至少在启动进入时也容易一些，长时间的坚持时也更有毅力一些；而你有天赋的领域，信心也足一些，并且拥有相对竞争优势；能挣到钱的领域，最好还比别得领域更挣钱，那么外在的经济激励会更强，而同等努力相对收益也更大。无怪乎，一个技术热潮起来后，大家都看到了第三点，急匆匆跳进去，但往往忽视了前两点。

另一方面，多个核心硬技能之间是一种加和关系，若非迫不得已，再下同样的大功夫去修炼另一项核心硬技能显得就不是那么明智了。所以应先深度修炼“一门”核心硬技能，建立门槛，但需要深到何种程度才能天下行走？如果刚开始起步算 0，1 算是行业平均水准，那至少先要专注在核心硬技能上，并修行到 1 以上，能进入前 20% 就更好了。

然后，就可以围绕核心硬技能适度练习和发展一些辅助技能，这些辅助技能大多属于软技能，也有部分硬技能，只是没有核心技能那么硬，通常起到放大和加强核心技能的作用，可以发挥指数效应。这也是为什么核心硬技能要先修行到 1 以上，因为指数关系只有在大于 1 时才有意义。

有些辅助软技能可以通过刻意练习来掌握，而有些则很难，属于埋藏在天生的基因和后天的成长性格中。在漫画《火影》的忍术体系中对这种天生的技能有个术语叫“血继限界”，其中最厉害的当属“写轮眼”。想想在职业发展的技能体系中，有什么是可媲美“写轮眼”的辅助软技能的？如果你幸运拥有这种“血继限界”，可别浪费了天赋。

程序员怕什么？就怕技术潮流的颠覆直接废了全身武功。我读大学时就经历过一次，当时主流的企业应用开发是 C/S 架构的 Delphi 和 VB，如今已是明日黄花。而武功体系由内力加招式组成，技术的演进容易废了招式，却不容易废了内力。

张无忌学会九阳神功，一身内力惊人，招式现学现卖也打的少林龙爪手高僧叫屈，所以在点亮技能模型树的过程中，你得分清九阳神功和龙爪手的区别。类比于技能模型树，内力是根茎，招式如花叶，时间流逝，落花残叶，冬去春来，复又发新。

到这里，关于技能的焦虑和建立技能模型的方法，我们就探讨完了，最后总结提炼下：

- 程序员这行的技术门槛没想的那么高，所以就易引发恐慌和焦虑；
- 建立你自己的技能模型，才能提升门槛和核心竞争力；
- 避开“达芬奇诅咒”，围绕核心硬技能，发展“一主多辅”的技能模型树。

从此，种下技能模型之树，让其茁壮生长，方能一技压身，天下行走。

在程序这个江湖上，你又是靠什么在行走天下呢？欢迎你留言分享。

程序员进阶攻略

每个程序员都应该知道的成长法则

胡峰 京东成都研究院 技术专家



精选留言



璫☆璨星☆空

现在主方向是Python，闲暇时间也会学习go，py前景明了，但我觉得go厚积薄发！所以想以Python为主，go为辅，不知可行吗？

2018-12-05 13:41

作者回复

只看语言不太好判断。要和领域结合，两种语言技能解决同一领域的不同问题

2018-12-05 19:19



石头

核心技能:一主多辅，形成技能模型。注意九阳神功与龙爪手的修炼！

2018-12-08 14:29



Kuer

老师你好，我刚刚踏入职场是做java开发的，想请老师可以指点一下，在以java为基础，会5大框架，后面应该修炼那些辅助技能。

2018-12-07 08:34

作者回复

沟通，表达，展现，多数程序员都不擅长的地方

2018-12-08 10:54



Since

感觉二八定律适应好多行业

2018-12-05 09:12



Dream.

是否存在可复制的技能模型？

又怎样才能找到属于自己的技能模型？

2018-12-05 09:01

作者回复

从最初选择的技术方向就决定了很大程度上这棵技能模型树的成长

2018-12-05 19:27

