

## 35 | 关系：学徒与导师

2018-10-22 胡峰



现在很多公司都有一种带新人的导师（**Mentor**）制度，导师制的初衷是为了帮助新员工快速熟悉公司环境，并提供工作技能和个人成长的帮助，正所谓“传帮带”。

这是用制度建立并约束了一种在新、老员工之间的关系，这本是一个很好的出发点。但想要类似这样的制度关系发挥期望的作用，恐怕就需要“导师”和“学徒”都有一个更高层次的清晰认知，毕竟制度只能在其中起到催化的作用。

### 起源

导师制诞生于十四世纪，随之带来的是一场翻天覆地的变化。

突然之间，那时的年轻男女们可以用自己最富余的资产——时间，去交换当时最稀缺的资源——培训。在那个时代，经验丰富的手艺人，比如，铁匠、鞋匠、木匠等，他们指导这些年轻人，并承诺将来某天年轻人能学会他们的技能然后去开创属于自己的事业。作为交换，年轻人会提供低成本且廉价的劳动力。

作为学徒，年轻人可能赚不到什么钱，但却能学到关于这门手艺的各种经验和技巧。比如：一个铁匠学徒，能学会或掌握如何去建造高温火炉，组合不同的金属以产生不同熔点的合金混合物，以及制作耙、刀或犁等工作技能。这些经验和技巧，在当时的学校里是都教不了的，只能进入这行业去获得第一手的经验。

那么程序员这行的导师制，像是中世纪时期那样吗？似乎有点像，但也不完全一样。我们都知道

编程这门手艺，你读的书再多、再好也不如真正动手去做。可是你一旦开始做了，也会很快掉入迷宫，因为路径千万，到底怎样才是对的？怎样才是好的呢？所以，现在好多公司都会说，“我们会为新员工或学生配备有经验的‘导师’来领路....”但，很多有经验的程序员并不能很好地理解（这也包括曾经的我），作为“导师”到底该做什么？要怎么做？以及做或不做于自己有什么关系？

比如，一个有经验的程序员，走到一名新员工面前，问：“你会 **Java** 吗？会这个框架吗？”

“学过 **Java**，但框架不太懂。”

“来，这里是框架文档地址，你先看看，搭个 **demo** 先跑起来。”

“恩，...”

这样的场景，也许大量存在于新手程序“导师”和“学徒”之间。

## 导师

有经验的程序员、老员工，站在“导师”的视角，会如何看待这样的关系呢？

从某种意义上来讲，经验丰富的程序员，就和中世纪的老师傅一样，他们经历了大量的时间犯过大量的错误，积累了很多难以言说的经验价值。他们已经经历过你所犯的错误，已然能够轻松应对如今让你痛苦和头疼的问题，所以他们具有能够引导你迈向正确方向的潜能。

但反过来想，他们为什么要指导你？只是因为公司有个导师制，并安排了他成为你的导师？那么这样的指导通常也就变成了上面那种场景。为什么他们要牺牲自己的工作时间，甚至私人时间来无私地指导你？也许作为新同学的你，甚至包括制度的制定者本身，可能也没从这个角度来看待该问题。

但如果不从这个角度来思考一种制度，那么很可能制度期望的是一回事，行动起来却是另一回事。若只是通过单纯的职业道德约束或价值观教育是解决不了这个问题的。毕竟中世纪的老师傅还可以靠利益交换与绑定来稳固这个机制和关系的。

大学里读研读博，也会有个导师。这样的导师，相对比职场的导师更进一步，因为你们之间有经济交换，你交了学费，所以学校导师就对你的毕业负有一定的指导责任。但你能获得多少质和量的指导与帮助，其实取决于你的态度和反馈。

所以你看，学生参加导师接的一些项目，本质上和中世纪的学徒提供廉价劳动力换取经验和指导是一样的。还有些学生，会为导师收集材料，用于发论文或写书，有些甚至干脆就是写好了论文或书，最后导师只是署个名。人品好点的导师可能还会给你留个第二作者的位置，差点的也许你连露脸的机会都没有。

而职场导师制，如果公司没有相应足够的考核、评价和激励制度支撑，那么这种师徒关系实际上

没有任何约束，完全靠运气、投缘之类的。站在导师的角度，对于凑巧碰到的一个职场新人，他有什么样的利益或情感驱动要去更积极地做这件事呢？其实最直接的，还是由对方的态度和行动来驱动的。

而在没有这些更实质的驱动因素时，有人如果愿意去积极地做这件事，那一定是在更高的维度看这件事。借用一句话来说明：

取得领先的方法，就是提携你身边的人。你对待别人的态度始终会伴随你，人们会忘记你所说和所做的一切，但永远不会忘记他们对你的感觉。帮助别人就是影响别人，如果你能帮很多人，你本身就是高手，你的影响力就很大，你就能做更大的事。

这是一个气度问题。

## 学徒

反过来，站在“学徒”的视角，该如何看待这样的关系？万维钢有篇文章叫《给前辈铺路的人》说得很有现实意义：

给人当学徒，就给你提供了这个机会。你现在把自己和一个高手连接在了一起，你可以从内部了解第一手的经验。这就是学徒工作的协议：用礼敬和服务，换取机会——而这个机会还不是立功露脸的机会，而是学习实践的机会。

机会，就是得到更快的成长与发展。从导师多年积累的经验中获益，能够缩短获得这些知识经验的时间，并且避免重复错误。但这里面可能还有个障碍，就是自尊心的问题，态度不够谦虚，那么也许是性格还需磨练。如果态度谦虚，双方都投入了适当的时间和精力，那么导师当年花了十数年才学会或领悟到的东西，学徒也许只用短短几年就能学到，绕过了没必要的重复路线。

从学徒方面来说，必要的、简单的、低技术含量或重复性的工作也是必须的，不应该被认为是一种浪费或牺牲。当你在免费获得大量的知识和帮助的同时，却抱怨时间投入太多，或者时间不够，其实是短视的。因为：

当你给人铺路的时候，你实际上也在左右他的前进方向。

这也是一个气度问题。

## 关系

师徒关系有很多种，最让你期待的是哪一种？

对我来说，联想起师徒关系，一下映入我脑中的是金庸小说《笑傲江湖》中的令狐冲和风清扬。在看这部小说时，也曾梦想遇见自己的“风清扬”，学会绝代天下的独孤九剑。但后来随着年龄增长，我开始觉得，现实中也许终究不会存在像“独孤九剑”这样的绝艺，也不会有风清扬这样的师傅，直到我遇到一位美国作者德里克（Derek），他在自己的文章里分享了一个他的成长故

事。

下面，我就从作者的第一人称来简述下这个故事。

## 学徒视角

那年夏天，暑假，我 **17** 岁了，高中刚毕业。开学后，我就将进入伯克利音乐学院学习音乐。那时，我困惑于一些音乐问题，又找不到人解答。所以，我随机打给了一个本地的音乐工作室，工作室的主人基莫（**Kimo**）接起了电话。

我们聊了起来，当他听说我将去伯克利学音乐时，他说：“我就是从伯克利毕业的，之后还留在那里教了好些年的音乐。我打赌，我能在几节课内教会你学校安排了两年的音乐理论与编曲课程。另外，假如你能明白‘不要接受学校速度的限制’这个道理，我猜你也许能在两年内毕业。假如你感兴趣的话，明天上午 **9** 点来我的工作室上课，当然，这是免费的。”

两年内毕业？太棒了，我喜欢这个风格，实在太激动了。第二天一早，**8:40** 我就到了他的工作室门口，但我等到了 **8:59** 才按响了门铃。

## 导师视角

一天早上的**8:59**，我的门铃响了，我当时完全忘了为什么这么早会有人来。一直以来，我偶然遇见过一些孩子，他们都说想成为伟大的音乐人。我告诉他们，我能提供帮助，然后让他们早上 **9** 点来我的工作室，但遗憾的是从来没有人早上 **9** 点来过。这就是我从一堆孩子中识别出那些只是随便说说，还是真正认真严肃地想干点事的人的办法。直到那天，他来了，按响了我的门铃，一切就这么开始了。

后来的故事就是，德里克只用了两年半便从伯克利毕业了，并将这个抬高的标准和速度应用在了之后一生的事业与生活中。而他们也从师徒关系，转化成了朋友关系，维持了几十年，直到今天。

这像不像一个现实版的“令狐冲”与“风清扬”的故事？而这，就是我期待的一种师徒关系。

现实中，对于师徒关系，会有人有这样的疑问：“教会徒弟，会饿死师傅吗？”也许中世纪时期的师徒关系会有这样的担忧，但如今这个信息时代，知识根本不稀缺，也没有所谓的“一招鲜，吃遍天”的绝招。反过来说，**带好了徒弟，接手并取代了你当前正在做的事情，你才有可能解放出来去做更高层次和更大维度的事情。**

而作为学徒，你需要吸取德里克的经验：**学习和成长是自己的事，严肃待之，行动起来，自助者，人亦助之。**

在成长的阶梯上，无论你在阶梯上的哪个位置，都可以努力去寻找和建立这样一种关系，最好的状态，我想应该既是学徒又是导师。你觉得呢？

---

# 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

