

33 | 定义：阶梯与级别

2018-10-17 胡峰



从今天开始，咱们专栏会开启一个大家可能都比较感兴趣的**主题：程序员的职场阶梯，以及攀登阶梯的晋升博弈。**

任何种类的职场上升通道都是一个阶梯，但程序员的阶梯有何不同呢？

在程序员职业生涯的发展过程中，都会经历一个修炼成长、打怪升级的过程，而每个公司可能都会定义自己的升级阶梯。以 **AT** 为首的两大巨头，其对技术人员的级别定义在互联网业界比较公开。例如，阿里的程序员级别从 **P4** 到 **P14**，而腾讯则定义了五个大级别：从 **T1** 到 **T5**，并且 **T4** 之前的级别内部还会细分为若干小级别。

相对来说，腾讯的 **5** 个大级别与我自己一路走来经历的几个阶段感觉会比较匹配一些，而大级别之间的分界线也会更明显一些。我对升级阶梯的定义也是 **5** 个：初级、中级、高级、资深和专家。

至于对不同级别的定义，我选择了三个相对容易判断的维度：

- 具备什么能力？
- 解决什么问题？
- 产生多大影响？

初级

初级，多属于刚入职场的新人。

一般刚从学校毕业的同学，具备基本的专业技能和素养，能快速学习公司要求的常用开发技术、工具和框架，能理解所在的业务和产品领域，并按照设计要求来实现功能。他们通常都工作在系统中局部某个区域内，能独立或在有限指导下实现功能并解决该模块碰到的具体问题。

这个级别基本完成的都是螺丝钉级别的工作，影响很有限。但如果从这个阶段你就开始定期归纳总结这些局部的工作经验，不断优化工作内容，并能在团队小组内部做出分享，甚至帮助其他同学解决问题，那就说明你已经走上了一条快速成长的通道。

刚入职场的同学，有本科，有硕士，还有博士，这有区别嘛？我个人感觉本科和硕士进入职场相差不大。当年我是硕士毕业，进入第一家公司算初级，本科算助理工程师，有一个小级别的差异，而薪酬待遇则相差无几。

那时腾讯也来学校宣讲，本科年薪 6 万，硕士 8 万，而博士 10 万。仅仅从年收入差距来看，读硕、读博似乎不是个划算的选择，可恰恰很多人选择读硕就是为了能有一个更好的工作起点，而选择的标准也可能恰恰就是薪酬占据主导方面，这貌似是一个误区。

以前看过一期《奇葩说》，一个清华男从本科读到博士，跑去节目上说了半天就是为找什么工作而苦恼，惹得同为清华毕业的高晓松当场发飙，而同为点评嘉宾的蔡康永也说了句很中肯的“实在话”：

一直花时间求学，也许是为了拖延人生做决定的时间。

中级

中级，相对初级最大的质变在于：独立性。

初级同学经过两三年工作历练，对实现各种业务功能、开发规范流程都很熟练了，摆脱了对基本指导的依赖性，这时就进入了中级阶段。中级工程师已经能够独立承担开发任务，设计实现他们负责的系统模块，以及通过搜集有效信息、资料和汲取过往经验来解决自己工作范围内遇到的问题。

中级这个层面的基本要求就是：**完成动作、达成品质和优化效率**，属于公司“动作执行”层面的中坚力量。观察下来，这个级别的工程师多数都能做到完成，但品质可能有瑕疵，效率上甚至也有很多无效耗散。不过，效率和品质总是在不断的迭代中去完善，自身也会在这个过程中不断成长并向着下一个阶梯迈进。

不少同学卡在这一阶段，就是因为虽然不断在完成工作，但却没有去反思、沉淀、迭代并改进，从而导致自己一直停留在了不断的重复中。所以，在工作中要保持迭代与改进，并把你的经验分享给新来的初级同学，这样在未来之路你不仅会走得更快，而且也可能走得更轻松。

高级

高级，不仅要能独立完成工作，还要能独立负责。他们能独立负责一个大系统中的子系统或服务，并成为团队骨干或最重要的个人贡献者。

相比于中级，高级工程师在“动作执行”层面，不仅能独立完成高级难度的开发任务，而且在用户体验（品质提升）和性能优化（优化效率）方面还都能做出更全面的考量。也就是说，他们不仅仅可以把开发任务完成得又快又好，而且还能清晰地定义出多快、多好。比如，一个服务的响应时间 **99.9%** 是在 **20 毫秒**内，内存消耗最大不超过 **1G**，并发吞吐量 **10000+/s**，类似能用清晰的数据来定义服务品质和效率。

另外，高级别需要面对的问题就不再是单一维度的技术问题了，他们需要结合业务特性去考虑设计合理的解决方案。熟悉业务领域内的应用系统架构以及各个部分使用的技术，能根据业务特性，合理进行分层设计，实现高效率、低成本的运维或运营。

初、中级别的能力提升与影响输出是通过经验的归纳总结与分享，那么高级则需要在经验这种偏个体特性的基础上，再进行抽象提炼，沉淀方法论。换言之，通过个人的经验，研究行业的优秀实践，再结合自身实践和逻辑推导，沉淀出切合现实的方法论，并在团队内部推广应用。

资深

资深，有深度和资历（即广度）两个层面，对应到职业生涯路线上，也有两个方向。

- 资深工程师
- 架构师

在偏基础研发、算法和特定技术复杂领域，会向“资深工程师”方向发展，属于深度优先。而在面向业务开发的领域，业务复杂度高于技术复杂度，则会向“架构师”方向发展，属于广度优先。

但无论深度还是广度，进入这个级别即说明你在特定领域都已经具备了相当的积累。这时你是作为相关领域的专家，深度参与和支持团队项目，在领域内进行关键的技术判断和决策，进而帮助团队项目或产品加速成功。在这个层次上，你面临的都是一些更复杂的、具备一些灰度（不是非此即彼，而是需要折中权衡）特性的问题，这时就需要你能够全方位、多层次、多角度地深入理解问题，评估每种方案的收益、成本和潜在未来的长短期影响等。

这个层次的影响方面，除了经验分享和方法论沉淀，还有**产品**和**团队**两个考虑维度：即使是做纯技术的东西，最终的影响也是通过技术产品来完成的；而另一方面则是团队的梯队建设、结构调整与协作优化，决定了团队外在表现。这两个维度，前者可能资深方向侧重多一些，后者则是架构师方向需要侧重思考实践的。

专家

专家，表明了某种领域的明确建立。

也许架构师和资深工程师也具备在特定细分技术领域的深厚积累，说明他们和专家一样也有属于自己的领域，但这个领域还不算明确建立，它还需要有公认的影响力。公认影响力实际指一个范围，如果是公司的技术专家，那么范围就是公司或行业。

虽然以“家”冠名会让人感觉太高不可攀，遥不可及，但实际“家”也分大小：一般的“大家”可能属于稀世珍宝，举国稀有的，确实是遥不可及；但也有“小家”，相对来说就没那么遥远了。“大家”和“小家”的区别，就在于影响建立的范围大小。

影响力听起来可能很虚，那我换个相对实的角度来说说。作为一个 **Java** 程序员，在学习使用 **Java** 的过程中总有那么几个人，你不仅要去读他们的书还要去看并且使用他们写的代码，反正在 **Java** 这个领域你总是绕不过去。那么，这就是他们在这个领域实实在在的影响力，自然也是这个领域的专家。所以，专家可能就是“这个领域内你绕不过去的人”吧。

积累多年，建立体系，形成领域，他们需要解决的最重要的问题是：面向未来不确定的战略问题。这就像机器学习用过去长期积累的数据，建立起一个模型，用来预测和判断未来。未来不可测，但建立好了一个领域体系后，当未来到来时，就可以很快地将新出现的信息加入到现有的领域体系中去，从而修正模型，做出快速地调整与决策。

最后，我借用鲁迅在《故乡》里说的一句名言：

其实地上本没有路，走的人多了，也就成了路。

前面定义出来的阶梯就是那很多人已经走过的路。不管现在走到了哪个阶段，我们都走在同样的路上，但会遇见自己不同的风景。

在你攀登职场阶梯的路上，你走到了哪一级？对每级阶梯有怎样的理解呢？

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

