

25.02-你是什么段位的程序员?

作者: 霍丙乾 bennyhuo

来源: 知识星球: bennyhuo

版本: v0.1

引言

朋友们好, 这一期会员节目我想做一件比较有难度的事情, 那就是建立一个参考标准, 给互联网软件开发人员划分等级。

说到等级, 大家很容易就能想到的就是阿里的 P7、P8, 腾讯的 T9、T10 这种等级。企业的技术等级一定意义上可以区分开发人员的技术水平, 不过这些只是企业的管理手段, 其目的是为了激发员工的工作积极性, 为企业产出更大的价值。说直白一点儿, 你是 T 几, 更多是用来告诉你, 你的工资不能超过这么多钱; 如果想要更多回报, 请按照公司的等级设计努力工作, 提升自我。公司很多时候还怕员工不能理解其中的奥义, 对晋级提出明确的要求: 要武功, 更要战功。也就是说, 要做出成绩, 拿业务成果晋级, 而不是拿技术实力晋级。

这是符合企业利益的, 你不能要求企业在这个上面纯做慈善。当然, 企业的技术等级是有其积极意义的。企业的等级要求可以为个人的成长提供很好的引导和依据, 按照这些要求提升自我也能让从业人员更好的适应职场, 软实力和硬实力一起抓, 全方面的掌握做事的技能。不过, 作为日常脚踏实地编码干活的开发者, 我们难免发自内心地更认可技术硬实力过人的技术同行, 而对只会向上管理、汇总周报的团队润滑剂嗤之以鼻。事实上, 得益于众所周知的原因, 后者在企业内往往如鱼得水, 这使得不少一心只想钻研技术的开发者陷入自我怀疑。

基于这一点, 我准备以纯技术成长为尺度、以个人技术能力为依据, 整理一个程序员等级标准, 希望能够为大家的技术成长提供一些指引和帮助。相比之下, 我们这个标准更强调纯粹的个人技术能力, 不关注业务能力和管理能力。换句话说, 套用这个标准, 你的 leader 很有可能级别比你低; 粉丝几万甚至上百万的博客专家可能也不能得到比你更高的评价。此外, 我们也不关注年龄和工龄, 这是这个行业最具魅力的地方, 因为个人的努力完全可以超越时间限制, 17 岁的高中生可以比工作十年的老油条更懂如何开发和维护一个开源项目。当然, 这套标准受限于我个人的见识和水平, 一定会出现一些片面的情况, 因此我也希望多与大家讨论, 一起持续更新和修订这个标准。

正文

我们将程序员分为 11 个级别, 即 0-10 级。为了有别于企业的技术等级, 我们在命名上不叫 **P 几**, 也不叫 **T 几**, 直接称为**几级**。当然, 如果你乐意, 你可以称之为**几 J**。比如, 我是 **4J** 程序员。

0 级和 10 级这两个级别用于涵盖什么都不会和什么都会的情况, 所以关键级别是 1-9 这 9 个级别。其中, 1-3 级重点强调能做出什么, 4-6 级则强调能想到什么, 7-9 级则强调能影响什么。级别越高, 能力和想象力也越大, 描述也必然更加抽象和宽泛。

如果评估你是否满足 **N 级** 的要求呢?

1. 对于 1-3 级, 基本符合 80% 左右描述即认为满足该等级的要求, 4-6 级为 50% 左右, 7-9 级为 30% 左右。
2. 满足 $\leq N$ 的所有级别的技术能力要求。

例如:

1. 你已经做过一些简单的性能优化, 但你仍然不能熟练掌握自己最常用的编程语言, 不符合 **5 级: 性能观察员** 的要求。
2. 你表达能力出众, 甚至已经有畅销书在售, 符合 **7 级: 技术探索者** 的部分描述, 但技术能力仍然不满足 **6 级: 质量监督员** 的要求, 因此评估级别只能在 6 级以下。当然, 这说明你有很大的提升潜力, 但我们主张要脚踏实地, 不要眼高手低。

0 级: 预备程序员

- 你可能刚看完张雪峰老师的直播, 准备报考计算机专业, 立志做一个优秀的程序员。
 - 你也可能刚学完大学计算机基础, 对计算机有一定的认识, 可以熟练打字, 甚至能熟练掌握 Office 三件套的使用。
 - 你也可能踌躇满志, 准备照着教程安装 Turbo C 或者 Visual C++, 开启新世界的大门。
 - 你在思考如何开始学习编程, 或许你会遇到一些小问题, 不知道如何在网上找到答案。
 - 你已经有了一些学习的热情, 可能会参与编程兴趣小组, 也可能成为各类基础视频教程的常客。
 - 你可能还需要一些时间来找出自己最感兴趣的技术方向和编程语言, 又或者, 你还没有意识到这是个问题。
 - 在你看来, 通过几行代码就能让屏幕上出现跳跃的字符, 实在是太神奇了!
-

1 级: 环境配置工

- 你已经至少掌握了一门编程语言的 Hello World 的编写和运行, 掌握了开启新世界大门的钥匙。
- 你可以熟练地配置开发工具和开发环境, 尽管你可能并不清楚为什么要这样配置。
- 你能完成简单的程序开发, 能使用循环来帮助家里的小孩子解决数学难题。
- 你对编程语言的基础知识有了初步的掌握, 但仍然缺乏对编程范式的理解。
- 你可能会在网上查找如何解决一些具体的开发环境问题, 但并不一定能完全理解背后的原理。
- 你还没有真正接触到编程中的常见问题, 如调试技巧、错误处理等。
- 你可能对操作系统的基本概念(如文件管理、内存管理)知之甚少。
- 你可能不清楚什么是代码版本管理, 会用 QQ 或者微信传输代码, 也会把代码上传到云盘。
- 对你而言, AI 是很好的帮手, 它能帮你解答数不清的问题。

2 级：代码搬运工

- 你已经能熟练使用编程语言的基本语法，编写较为复杂的程序。
- 你已经知道代码版本管理的好处，并且学会 git 的一些基本用法。
- 你能够在团队协作中发挥作用，处理一些简单的任务，并且能够理解团队的工作流程。
- 你通常会参考现有的代码，按照需求修改，而非从头开始编写。
- 你对技术栈的掌握还不够全面，常常需要查找大量文档来解决问题。
- 你经常在技术交流群里说你遇到了一个问题，但总是描述不清楚。
- 你觉得算法没用，平常也用不到什么算法，面试考算法完全就是难为人。
- 你会疑惑设计模式是什么，听上去很高级的感觉。
- 你的工作内容通常较为简单，你觉得 AI 代码补全真的很有用。
- 你的梦想是成为架构师，可什么是架构师呢？

3 级：接口装配工

- 你已经能独立开发并实现较为复杂的系统功能，能够自己构思代码实现。
- 你已经熟练掌握 git workflow，并且在团队里积极推广。
- 你能够在已有的系统中添加新功能或调整旧功能，熟悉接口和模块的设计与调用。
- 你开始理解并运用设计模式，单例模式有几种写法是你最喜欢讨论的话题之一。
- 你了解并使用过一些常见的开发框架和库，能够提高自己的开发效率。
- 你能理解并运用一些基本的软件设计原则，但大多数时候依然是依赖别人提供设计。
- 你能够按照需求完成较为规范的代码，但可能在代码质量、可维护性上有所欠缺。
- 你对编程语言的语法了解不深，以至于编码时总是出现技术性 bug。
- 你对不同编程语言之间的差异了解不多，也没有深入了解过计算机的底层知识。
- 遇到问题你总是能通过搜索引擎或者 StackOverflow 找到答案。
- 你开始意识到数据结构和算法有用，但仍然对于面试中如此重视算法而感到不能接受。
- 对你而言，AI 依然是很好的帮手，但你也经常抱怨 AI 有时候不怎么靠谱。
- 你对自己的技术能力充满信心，也经常在技术交流群里侃侃而谈。

4 级：问题研究员

- 你已经可以熟练运用你常用的编程语言和编码工具。
- 你开始有意识地通过优化工具和编写脚本来提升自己的工作效率。
- 你完全无法理解为什么有人会不使用快捷键。
- 你已经开始注意代码规范和代码质量，并有意识地为程序增加更多的可扩展性。
- 你开始不满足于表面的现象，并尝试深挖问题的本质。
- 你开始意识到基础知识对探索问题的本源所带来影响。
- 你开始撰写技术文档，总结解决问题的思路和方法。
- 你在团队中能够独立承担功能模块的开发任务，偶尔能为同事提供技术支持。

- 你开始参与团队内的技术决策，能够提供建设性的意见，推动技术方案的改进。
- 你可能还没意识到，你已经是一个有追求的程序员了，言谈举止都透露着优秀程序员的气质。

5 级：性能观察员

- 你已经可以灵活自如地运用你常用的编程语言的各种特性。
- 如果你最常用的语言是 C++，你应该能够对编译器常见的性能优化如数家珍。
- 如果你最常用的语言不是 C++，你应该对 C/C++ 有一定的认识 and 了解，可以读懂常见的 C/C++ 代码。
- 你编写代码时开始小心翼翼，不仅关注程序正确性，还会关注程序的性能。
- 你很清楚数据结构和算法对性能的影响，会选择合适的算法来提升程序的性能。
- 你对常见的性能问题有较深的理解，如内存泄漏、GC 垃圾回收、CPU 使用率等。
- 你可以熟练运用常见的性能工具分析程序的性能瓶颈，并给出优化方案。
- 你理解系统的各个组成部分，能够在实现新功能时考虑到系统整体的架构和稳定性。
- 你开始看到散落的知识点之间产生美妙的连接，知识体系逐渐构建起来。
- 不过，你也更加清楚的认识到自己存在哪些不足，并开始尝试努力补齐自己的短板。

6 级：质量监督员

- 你会根据项目需求接触并深入研究各个领域的技术。
- 你已经掌握了多个横向领域的技术，能够超越技术栈给出适合项目的技术方案。
- 你对软件质量有深刻的理解，注重产品的稳定性、安全性和可维护性。
- 你能够使用各种工具进行静态分析、动态分析和代码审查，以提高代码的质量和安全性。
- 你总是能在代码评审中发现别人意识不到的细节问题。
- 你会为项目制定合理的量化指标，并使用技术手段持续跟进防止劣化。
- 你可以为项目沉淀技术规范，制定最佳编码实践，提升项目整体的代码质量和编码效率。
- 你在团队中发挥着技术引导作用，推动团队不断提升软件开发的质量和稳定性。
- 你也会利用闲暇时间开发一些开源项目，或者有能力为一些著名的开源项目贡献代码。
- 你也开始在一些场合分享自己的技术实践，尽管有时候你还不太习惯在很多人面前大声表达自己的意见。
- 对了，你可能掌握了多门编程语言，也有编程语言偏好，但对于吹捧特定语言的行为持保留意见。

7 级：技术探索者

- 你能够熟练运用多种技术解决问题，并清楚其背后的原理和细节。
- 你掌握了多种编程范式，熟悉一门或多门的编程语言的编译和运行细节。
- 你可能是操作系统专家，阅读和修改 Linux 内核源码是你的日常工作。
- 你可能是数据库专家，在 SQL 优化时多看一眼 explain 都是对你技术的不信任。

- 你可能曾经参与或者主导开发一些开源项目，也曾向一些著名的开源项目贡献代码，并被接受。
- 你可能是某一种或者几种技术的忠实拥趸，也愿意为之投入时间和热情，不停地挖掘其中的技术细节。
- 你可能曾通过演讲、写作、开源项目等多种形式分享知识，并初步建立了一定的影响力。
- 你具有系统化的技术思维，能够深入探讨并解决行业中具有挑战性的问题。
- 显然，你已经是经验丰富的开发者了，研究技术是你生活不可或缺的一部分。

8 级：领域贡献者

- 你可能解决了行业内的某些技术难题，技术成果得到了广泛应用。
- 你可能发布过影响力较大的技术论文或技术专著，内容有足够的技术含量并得到了同行的认可。
- 你可能是行业标准委员会成员，或参与行业规范的设计和撰写，亦或是为国内外知名项目贡献代码。
- 你通过演讲、写作、学术研究等方式推动技术的普及和应用，影响了成千上万的技术从业者。
- 你积极参与国际性或全国性的技术论坛、学术会议等活动，并担任重要讲者。
- 不得不承认，你的低调，已经无法掩盖你的光芒了。

9 级：行业领导者

- 你是整个行业或技术领域内的顶尖人物之一。
- 你是行业内的重要意见领袖，你的言论和技术实践经常被同行和媒体关注。
- 你领导或推动着一些大型的技术转型和突破，解决了行业的痛点问题。
- 你主导的技术方案和架构设计在行业内成为标杆，被同行争相效仿。
- 对你而言，技术能力的评价显得过于单薄，你想要的是星辰大海。

10 级：计算机科学家

- 您已经是顶级专家了，我们无法对您做出任何描述和定义，一切皆有可能。