

TURING

# 程序员进阶之路

[韩] 郑昞勋 著  
张翔 译



从码农到  
中层管理者

一线  
程序员

+

领导力

=

团队  
负责人!



中国工信出版集团



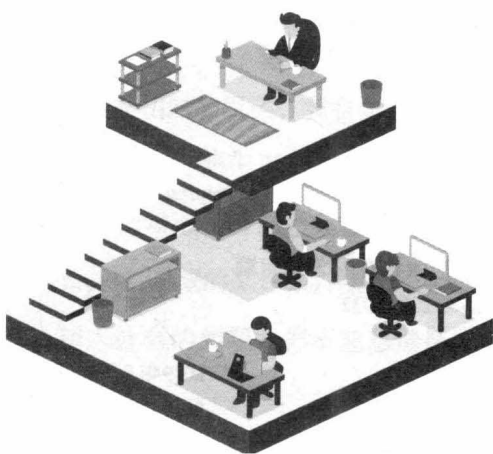
人民邮电出版社  
POSTS & TELECOM PRESS

TURING

# 程序员 进阶之路

从码农到中层管理者

[韩]郑暎勋 著 张翔 译



人民邮电出版社  
北京

## 图书在版编目(CIP)数据

程序员进阶之路:从码农到中层管理者/(韩)郑  
映勋著;张翔译.--北京:人民邮电出版社,2019.5  
ISBN 978-7-115-50405-0

I. ①程… II. ①郑… ②张… III. ①软件开发—项  
目管理 IV. ①TP311.52

中国版本图书馆CIP数据核字(2018)第287315号

## 内 容 提 要

程序员未来将成为中层管理者,而此时当下他们所欠缺的就是“领导力”。本书讲解了高级程序员、项目管理人员必须具备的领导力,以及如何具备这种领导力。书中介绍的不是枯燥的理论,而是作者在完成项目过程中积累的实际经验,通过多种角度讲述了其自身感受。读者通过本书可以了解程序员的特性,详细学习能够获得程序员进阶所需的领导力。

本书适合IT公司中层管理者、关注自身职业规划的程序员、希望成长为高级程序员的初级IT从业者阅读。

---

◆ 著 [韩]郑映勋  
译 张翔  
责任编辑 陈曦  
责任印制 周昇亮

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市君旺印务有限公司印刷

◆ 开本:880×1230 1/32

印张:6.5

字数:139千字

2019年5月第1版

印数:1-3000册

2019年5月河北第1次印刷

著作权合同登记号 图字:01-2016-1580号

---

定价:49.00元

读者服务热线:(010) 51095186转600 印装质量热线:(010) 81055316

反盗版热线:(010) 81055315

广告经营许可证:京东工商广登字20170147号

## 前言 永远保持一颗一线程序员的心

我工作的地方有几台日本产的设备，偶尔经过的时候，可以看到白发苍苍的日本工程师开发程序的身影。那个场面让人觉得他们不是在做软件开发，而是在完成一项真正的工程，甚至可以感受到他们身上那种日本人特有的“匠人精神”。看着这些二三十年一直从事软件开发的外国工程师，感受着公司这样的氛围，我为什么会感到些许羡慕呢？

如果韩国业界也是相同的氛围，我自然会产生同样的敬佩之情，但事实上，我在韩国很少看到年迈的工程师，而且就算看到也会产生诸多猜测，因为谁都知道，韩国公司的情况并不是那样的。程序员的职业寿命不长，IT领域的工作条件更是艰苦，这些都被人们所熟知。现在，初入社会的年轻人都不愿意选择工科职位，从某种角度上看，这也无可厚非。

程序员这个职业充满魅力，能够充分展现创造性。可以说，程序员是各种文化内容的搬运工，对激活内容起着核心作用。他们有着可以左右IT机器性能的超强能力。如果各位是在恶劣的IT工作环境中从事软件开发的程序员，一定会被这个职业的魅力所折服，

从而积极投身于这项事业。

我也曾是其中一员，写代码的时候完全感受不到时间的流逝，熬夜修改 bug 也是家常便饭。当我经历了 10 年的时光、终于明白了程序员的真谛时，公司又交给了我一项新的任务：一个人跟键盘和显示器过日子的时代已经结束，是时候帮助后辈奋勇前进了。更准确地说，是让我去管理一个团队。由于和代码的前缘未尽，对编程也还恋恋不舍，所以再像以前一样写代码时，我都会被上司批评：

“不要再自己一行一行地写代码了，放手让团队成员去做吧！”

遗憾的是，韩国 IT 界对程序员的职业规划并不是“初级程序员→中级程序员→高级程序员”，而是“程序员→中层管理者→主管”。由此可见，程序员并不是“只写好代码”就万事大吉了，还必须是“管理团队”的一把好手。所以，如果一个程序员不能成为中层管理者，那么就只能被淘汰。

既然不能一直做一线程序员，那么就应该以最快的速度适应。其实不只是“适应”，而是应当下定决心做好管理。虽然我满腔热血想做的是一线程序员，但还是要暂时放下代码，走上管理之路。从一线程序员转型为管理者并非一蹴而就，而是需要不懈地努力，并总结和学习各种经验教训。我在这一过程中并没有得到过来人的指导，也是自己经历了一切后，触发了将这些故事写下来与他人分享的想法。在本书中，我希望能与读者分享自己在成为一名出色的管理者这条道路上积累的知识、付出的努力以及收获的经验。

本书会告诉你，高效管理程序员需要树立的战略和成功所必备的要素。程序员出身的我们也许对管理方面的基础知识了解不多，但对程序员的心理和业务了如指掌。熟知团队成员的业务和特征为我们成为中层管理者铺平了道路，也是我们最有利的出发点。因此，我认为，以现有的经验加上管理方面的知识，各位一定能成为优秀的程序员领导。



## 致谢

弹指一挥间，我被编程的魅力所感染并以此为业已经十多个年头了。在这期间，无论是和我一起并肩战斗的前辈、同事，还是一起度过美好时光的后辈，都让我无法忘怀。我想将我的经验和情感写出来和大家一同分享，虽然不敢说我走过的道路一定是正确的，但希望能给那些跟我有相同困扰的同道中人提供一些帮助。

在此，要特别感谢一直关注本书进展的 Hanbit Media 宋诚根组长。也要借此机会衷心感谢在资料编写方面给过我许多帮助的郑智妍女士、为我提供很多想法的朋友圭达，以及技术人员赵诚文和李向韩。

同样要感谢一直在我身旁支持我、鼓励我的爱妻秀美，还有两个孩子雅仁和时宇。

郑暎勋

## 本书结构

本书讲的并不是领导力的专业知识，而是我作为一名程序员和项目经理在项目进行过程中亲身经历的一些事情。因此，本书不会像教科书那样包含与领导力有关的全部理论，而是以“蒙太奇”的手法，对我主观上认为重要的内容加以描述。下面介绍一下程序员的特性，以及具备什么样的领导力才能获得程序员的支持。一个程序员若想发挥领导力，需要具备以下 4 项要素：

- 对程序员和领导力的理解
- 项目管理
- 领导的沟通
- 领导的自我开发

第 1 章主要讲述领导力的必要性及意义、影响力的源泉，还会对程序员的心态和特殊文化进行探讨。通过这个行业的特殊性导致的问题及其应对方法，学习如何正确发挥领导力。

第 2 章讲解为获得项目成功而应当实施的资源管理方法，以及一些有用的小技巧。比起大家都耳熟能详的软件工程理论知识，这



一章更注重那些在项目进行过程中获得的经验以及欲与他人分享的内容。

第3章讨论高效的组织管理方法和沟通方案。例如,作为领导所需要的心理建设、与成员进行灵活沟通的方式方法、要避免的问题等。这一章有助于理解年轻一代程序员的所思所想。技术培训既可以提高团队能力,也可以强化领导的影响力,这是非常必要的。新一代程序员都个性十足,可以说这是新时代带来的结果,所以必须理解他们。只有理解了发生变化的程序员群体并与之产生共鸣,才有可能得到他们的信任。

第4章强调领导也要不断努力开发自我,要有勇气并能适应飞速发展变化的IT界。此外,还介绍了作为领导如何才能当好模范、起到表率作用,并强调了一些经常被程序员忽略的部分事项。

本书罗列了程序员发挥领导力时需要具备的4项要素,以及我的一些经验和想法。

# 目录

## 第1章 程序员领导力 1

- 01 程序员者，专家也 2
- 02 为什么要成为程序员的领导 9
- 03 程序员领导必备的领导力 13
- 04 提升领导力的有效方法 16
- 05 传授技巧，打造左膀右臂 20

## 第2章 项目管理 23

- 06 不畏惧新项目 24
- 07 做好项目的前期准备 26
- 08 高效工作的日程管理方法 29
- 09 需求事项必须落在笔头 33
- 10 开发现场最重要 36
- 11 结对编程 39
- 12 项目遇到危机时如何化解 42
- 13 当程序员诉说技术难关时如何应对 45
- 14 走过低迷期 49
- 15 不要吝惜对程序员有帮助的硬件 52

第3章 领导的沟通 55

- 16 组织内的交流 56
- 17 解决方案不是绝对需求 60
- 18 领导影响力的来源与解决问题的类型 64
- 19 程序员的性格与应对方式 69
- 20 称赞的技巧 75
- 21 称赞上司的方法 79
- 22 人不知而不愠，不亦君子乎 83
- 23 不要对成员讲述烦恼与不满 88
- 24 必须适应初级程序员的变化 92
- 25 让聚餐变成沟通的场所 95
- 26 升职的方法 98

第4章 领导的自我开发 103

- 27 我就是唯一 104
- 28 要有规划 112
- 29 摔碎自己的饭碗 117
- 30 自我投资 120
- 31 编程的乐趣 126
- 32 趣味编程 132
- 33 学习如何学习 136
- 34 程序员的写作 144
- 35 如何撰写文档 149
- 36 宣传团队成果的必要性 154
- 37 陈述发言的方法 157
- 38 紧跟时代变化 162

- 39 “饭碗”的大小 169
  - 40 站在巨人的肩上 173
  - 41 软件工程的必要性 176
  - 42 10年计划 181
  - 43 面对加班的心态 184
  - 44 乙方的命运 187
- 
- 后记 程序员的前景 191

# 第 1 章

## 程序员 领导力

# 01 程序员者，专家也

程序员熟知编程语言、算法、数据结构、计算机系统专业知识，对欲创建的对象进行分析、设计、调试、实现 UI 并正式运行。光看这些名词就知道，这并不是一个容易上手的领域。但是在韩国，相较于医生或律师等职业，人们并不认为“程序员”也需要具有专业技术。其实，与其他职业相同，程序员也需要不断扩充自己的知识储备，对技术有更深入的了解，与时俱进。

我认为，人们对该职业认识不足的原因在于，在其他国家，成为程序员的门槛很高，但韩国并非如此。就韩国来说，随着 2000 年 KOSDAQ<sup>①</sup> 时代的开启（互联网泡沫），IT 业迎来了自己的春天，国家从政策上提出大量培养程序员，也就是从那时开始，程序员的行业门槛变低了。

在美国，计算机工程师、程序员及系统分析师的平均起薪为 59 000 美元，而出纳员的起薪仅为 21 000 美元。这种起薪差距充分体现了人们对 IT 领域专家具有极高社会价值的认可。虽然

---

<sup>①</sup> 韩国证券交易商协会自动报价系统，是韩国的创业板市场，隶属于韩国交易所。——编者注

人们普遍认为 IT 专家可以创造巨大的附加价值，当然可以拿到高额年薪，但是从“供求法则”上看，则可解释为供小于求。也就是说，现有的 IT 人员储备并不能满足社会的需求。正因如此，美国、新加坡和欧洲的一些国家都在为获得充足的 IT 人员储备而不懈努力。

与之形成鲜明对比的是，韩国程序员的工作环境往往与发达国家相差甚远。如前所述，我认为是在 IT 业春天到来的时候，支持“大量培养程序员”的国家政策造成了这一现象。供给在不断增加，却不能提供数量与之匹配的高端职位，工作环境只会越来越差。究其原因，可以归结为“供求关系”的破裂。为了适应这种艰苦的环境，程序员必须努力成为专家并得到认可。因为只有成为专家，才能在艰苦的环境中获得较好的待遇。

但令人感到意外的是，有着“软件架构师”头衔的程序员待遇却与美国相差无几。由此可见，“专家”的名号在韩国是可以得到广泛认可的。虽然如此，但并非所有程序员都要转型做架构师。在自己当前从事的领域内做到最好，积攒经验并能灵活运用，这才是最正确的选择。重要的是，要认为自己就是专家，具备向其他人推销自己的信心，不断提升自我实力。

瑞典斯德哥尔摩大学的安德斯·埃里克森教授提出了“10 年法则”：要想在任何领域成为大师，至少需要 10 年的不懈努力。此处重要的是，10 年间的努力是“用心”还是只“用力”。

韩国不乏在自己的领域中有卓越成绩的佼佼者。体育明星

中的代表人物有金妍儿、朴泰桓、朴智星等，他们都在自己的运动领域中至少坚持了10年。当然，还有很多运动员也如他们一般有着多年运动生涯，但即使超过10年，也并不是所有人都像他们一样出色。由此可见，在某一领域拔尖的人并不是只花费了10年的时间，而是付出了比其他人更多的努力，不断完善自我才实现的。

决定做一件事情时，大部分人都会在初期下定决心、付出全部，但随着时间的流逝，由于未能在短期内达到预期目标，很多人便会失去信心和积极性。如果不能跨过这道坎而只满足于现状，那么就会原地踏步。这样的人数不胜数。至少要付出哪怕比这些人稍微多一点点的努力，才有可能在自己的领域中冒尖。要想成为在10年间不断发展、成长的程序员，必须将自己置于艰苦的环境，不断积累经验教训，只有这样才可能成为专家。

在同一行业或同一家公司工作3年后，基本可以掌握成为技术人员所需了解的98%的知识。从业3年的程序员和从业10年的程序员做出的软件在外观上看不出什么差别，只有实际使用时，才能体会到二者的稳定性和维护便利性存在很大不同。这是决定项目成败的核心要素。从业时间长的程序员具备积攒多年的实力和经验，只懂技术的人无法与之相提并论。技术员与专家的距离只有2%，但这2%却需要用7年以上的经验去填补。为了那2%的空缺，专家需要用至少7年的时间去创造只属于自己的独家秘诀，总结个人经验，提高实力。也许有人会想：“就为了那区区的2%，有必要花费这么长的时间吗？”若想成为被认可的专家，必须付出比这更多



的努力。

我们耳熟能详的那些成功人士，无一不是经过长时间的磨炼才获得今日成就的。他们不只致力于 98% 的技术，而是将更多的时间和精力用在了那欠缺的 2% 上，为了达到甚至超越 100%，他们无惧一天一天近乎残忍的磨炼，最终获得了成功。这种积攒实力的过程正是在自己的领域出人头地的保证。

神经学家丹尼尔·格拉德威尔也提出了相似的理论。他认为，要成为某个领域的专家，需要 10 000 小时。10 000 小时是什么概念呢？我们可以这样计算，如果每天工作 3 小时，每周工作 20 小时，那么成为一个领域的专家至少需要 10 年。这就是“10 000 小时定律”。

这个时代需要的是有实力的专家，如果你是一名具有 100% 技术能力的专家，那么机会自然会找上门来。从我们羡慕的专家身上不难看出，他们都在自己的职业中有着突出的业绩。这些人正是通过工作和职业发掘自身才能的，他们不是为了挣钱才工作，而是为了实现自己的理想和抱负而努力。也就是说，如果只是为了维持基本生计而工作，那么很有可能半途而废。

让我们看看美国计算机科学家、Sun 公司（已被“甲骨文”收购）创立者之一比尔·乔伊的事例。乔伊在 20 世纪 70 年代初开始学习编程。当时的计算机可谓庞然大物，一台计算机可以占满一个房间，而价格也超过 100 万美元，且内存容量和 CPU 运算性能非常不尽如人意。在那个年代，能接触到计算机本来就不是一件容易