



A PROGRAMER'S
GROWTH ROADMAP
FROM BEGINNING TO EXCELLENCE

程序员成长路线图

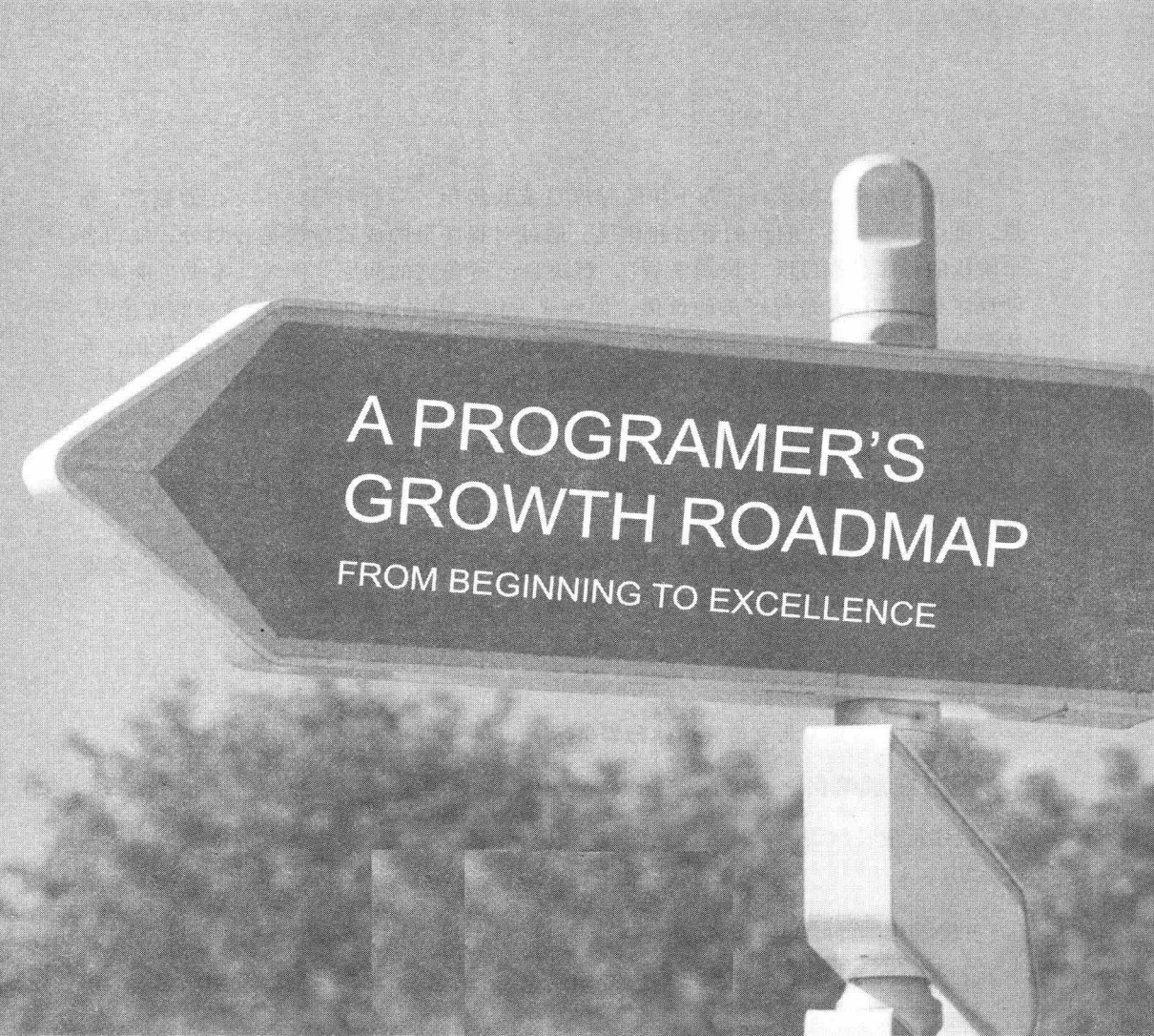
从入门到优秀

揭示程序员成长中的热点、重点、难点问题

N216 张磊 吉阳 著



机械工业出版社
China Machine Press



A PROGRAMER'S
GROWTH ROADMAP
FROM BEGINNING TO EXCELLENCE

程序员成长路线图 从入门到

N216 张磊 吉阳 著



机械工业出版社
China Machine Press

作者回忆和总结了自己几十年的程序员成长经历，对当前程序员关心的热点、重点、难点问题给出了自己的看法和建议。通过对程序员的成长阶段进行划分，使得各个阶段的程序员都可以“按图索骥”，解决自己所遇到的问题。同时，本书也能够帮助程序员了解什么是程序员的价值，如何成为优秀程序员，如何实现自身的价值等，从而给程序员提供向上进步的动力。本书既不是纯技术文章，也不是纯个人传记，而是采用随笔形式，通过问题提出、分析、解答的形式，并辅以个人成长的经历，展示作者对程序员成长的实践与理解。本书以技术成长和心理成长为两条主线，通过这两方面的结合，展示了程序员应如何实现自己的价值。同时本书还涉及了“企业经营模型”，据此针对程序员介绍了一些企业经营模型的知识，使得程序员能够看到更高层次的未来。

本书适合阅读的对象包括程序员、软件设计师、软件项目经理、软件公司（企业内部科技部门）职员、国家政府机关等相关企业信息化部门职员。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

程序员成长路线图：从入门到优秀/N216，张磊，吉阳著. —北京：机械工业出版社，2011.4

ISBN 978-7-111-33913-7

I. 程… II. ①N… ②张… ③吉… III. 程序设计—工程技术人员—自学参考资料 IV. TP311.1

中国版本图书馆CIP数据核字（2011）第051152号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：秦健

北京诚信伟业印刷有限公司印刷

2011年5月第1版第1次印刷

170mm×242mm·16.25印张

标准书号：ISBN 978-7-111-33913-7

定价：39.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

前 言

只要在编程序就应该称为程序员。若以这个标准来衡量，我可能是国内从事编程工作时间最长的程序员之一。几十年前我是一名程序员，几十年后的今天我依然是一名程序员，未来几十年我还会是一名程序员。

2008年我提出了EOM（Enterprise Operating Model，企业经营模型）理论，并把EOM系列文章发表到博客园网站上。也许是EOM对普通的程序员来说显得过于抽象，若没有丰富的工作经历和一定的编程技术，一般人会很难理解，因此网上反应平平。2009年年末，为了让普通程序员能够了解EOM，于是我从程序员关心的热门话题开始，由浅入深地发表了数十篇博文，最终让大家看到了EOM是程序员成长中的一个重要结果，它与程序员的成长密切相关。没想到这些文章一发布到网上就引起网友热议，无论是点击率、回帖数，还是作者与网友之间的辩论，激烈程度都很高，而且转载甚多。我的朋友、同事、家人甚至出版社见此状况都极力建议我将这些写成书，让更多的程序员能够分享这种成长。

于是我在2010年5月正式和出版社签约，开始进行本书的撰写工作。在写作的过程中，我和我的合作者对网上发表过的文章进行了整理，并根据本书的编写大纲增加了很多新文章，把这些文章按照程序员的成长阶段划分为入门篇、成长篇、成熟篇和优秀篇，从而形成了程序员成长的一个完整的“路线图”。在这个过程中我重新回忆了我成长的各个阶段，每个阶段的故事依然让我感到鲜活、感动和难忘。我努力去发现程序员最关注的问题，努力去分析这个问题后面的原因，努力去给出解决问题的建议。我想我的努力会给读者带来不一般的体会。

本书适用于各层次的程序员：有刚出校门寻找工作的新手，有在工作岗位年限不长的程序员，有工作多年的项目经理、技术高手，有在编程事业中表现突出的优秀人物。各层次的程序员都可以从中找到自己想要的看点，例如：新手想知道如何找工作、如何面试、工资待遇、掌握什么语

言、编程的前途如何等一些经验和观点；成长期中的程序员关注的是如何面对加班、如何提高自己的编程能力、如何确定自己的编程水平等；优秀的程序员可能关注优秀程序员应该具备哪些素质、哪些技术水平，如何证实其优秀，软件发展方向是什么，自己事业的未来发展方向是什么，如何实现自己的价值等。

每个程序员的成长之路各不相同，但都会经历大大小小的成功和坎坷，很多人在这个过程中会迷茫、会不知所措。希望本书能成为程序员成长的“路标”，打破那种程序员只看技术类书籍就能提高、就能成长的狭隘想法。希望程序员能多了解其他程序员成长的经历，这些经历不仅指技术上的，而且是指心理上的、职场上的、事业上的和梦想上的。程序员可以从中学到很多成功的经验，避免常见的问题，使得他们能够更快、更全面地成长起来，更好地在这个职业中有所发展。

在此，我要感谢合作者张磊、吉阳两位同事，感谢我的朋友、同事、编辑、家人，谢谢他们给我的每一次鼓励。他们中有很多人都是我的文章的首批读者，在我写作的过程中给予了许多有益的建议。没有他们的鼓励和支持我可能无法完成这项工作。我还要特别感谢远在英国的女儿倪好，她的鼓励如同我给予她的鼓励一样，本书将是我送给她的一件礼物，希望这份礼物伴随她快乐成长。

除了本书之外，我还打算在近期编写有关软件设计师、项目经理、EOM、NSS等方面的系列书籍。我想通过这些系列书籍与那些有志促进我国软件业发展的各位同仁分享自己的经验和观点。

由于篇幅有限，话题广度和深度也不可能拓展太多，请有兴趣的读者访问我的博客，参与讨论程序员以及软件业相关话题。

N216 (倪燕农)

博客：<http://n216.cnblogs.com/>

邮箱：eom_n216@hotmail.com

目 录

前言

引言	1
我的程序员成长之路	1
第一部分 入门篇	6
1.1 程序员的梦想——中国的比尔·盖茨	6
1.2 谈谈程序员的基础知识	8
1.3 语言选择与就业方向	11
1.4 新手如何学习一门新的语言	14
1.5 理性看待考证热	16
1.6 选择大公司还是小公司	19
1.7 新手面试常见问题和对策	21
1.8 薪水的苦恼	27
1.9 求书、求网还是求人	29
1.10 新手看高手	31
1.11 新手应该具备的基本素质	35
第二部分 成长篇	41
2.1 加班，加班，加班	41
2.2 大量编程带来的快乐和烦恼	44
2.3 需求总是变化，程序总在修改	49
2.4 为什么程序员不愿写文档	53
2.5 为什么编程者总是高估自己低估别人	57

2.6	我? 还是我们	59
2.7	为什么程序员的社会地位在下降	62
2.8	加薪的问题	63
2.9	门门通还是精通一门	67
2.10	程序是给自己看的还是给别人看的	70
2.11	程序越长水平越高吗	72
2.12	动手能力强与技术水平低	75
2.13	调试高手和编程高手	77
2.14	如何快速确定自身水平	80
2.15	程序员应该掌握的实用编程技能	83
第三部分 成熟篇		97
3.1	大项目或小项目都是程序员成熟之道	97
3.2	“顶梁柱”与“螺丝钉”的不同	101
3.3	如何对待新人	105
3.4	有关程序员的性别、年龄、个性、编程方法的话题	108
3.5	程序员的上升空间在哪里	112
3.6	跳槽还是留守	115
3.7	你能当老板吗	118
3.8	动手与动脑的关系	121
3.9	编程语言有高低之分吗	125
3.10	面向过程和面向对象的编程	127
3.11	功能和界面哪个更重要	130
3.12	你考虑过程序的复用问题吗	133
3.13	谈谈程序的参数化	137
3.14	漫谈程序的效率和水平	143
3.15	好的程序像一首诗	149
3.16	如何计算程序员自身的价值	151

3.17	程序员成熟的标志	154
第四部分	优秀篇	161
4.1	成熟到优秀的瓶颈问题	161
4.2	梦想回归	164
4.3	激情! 激情! 激情	168
4.4	摆脱技术束缚, 拓展业务视野	171
4.5	预测趋势, 让你的目光看得更远	174
4.6	有意识才会有行动——谈谈市场意识	178
4.7	制作有价值的软件才是程序员内在的目标	181
4.8	从程序制作到架构制作	184
4.9	从定制软件到通用软件	188
4.10	何为 EOM	192
4.11	用 EOM 的眼光评判“我要做全国最好的标准权限组件和 通用权限管理软件” 1	197
4.12	用 EOM 的眼光评判“我要做全国最好的标准权限组件和 通用权限管理软件” 2	201
4.13	用 EOM 的眼光评判“我要做全国最好的标准权限组件和 通用权限管理软件” 3	207
4.14	用 EOM 的眼光评判“我要做全国最好的标准权限组件和 通用权限管理软件” 4	214
4.15	程序员的春天: EOM 与程序员	220
4.16	优秀程序员应该具备哪些素质	224
第五部分	附录	230
5.1	创新模型简介	230
5.2	项目简介	236
5.3	作者和网友之间的精彩回帖节选	239

引言

我的程序员成长之路

程序员的成长经历往往很相似，大部分的人走过了最前面相同的一段路，而有的人则走得更远。总结自己这些年来的历程，这也许能让年轻的程序员少走一些弯路，成长得更快；或许更好一些，能让大家从中得到一些启发，早日进入优秀程序员的阶段，实现梦想，释放激情。

第一阶段，最初是在学校里学习计算机基础知识，学习经典的程序设计语言，编写测试用的小程序。这个过程可以说是对计算机和程序设计的入门阶段。这个阶段主要是培养了自己对计算机软件的兴趣，打下了良好的计算机基础知识。

第二阶段，而后参加工作，从事计算机软件开发工作。按照工作要求，一边学习，一边编程，终于可以让自己的程序投入运行了。在这个阶段我突然感觉到了自己的价值，感觉到了软件的神奇，并且自己编写的软件成为了实用产品。这个阶段实现了学习到生产的过渡。

第三阶段，随着工作的增加，开始编写各种程序，开发各种系统，这时候忙于编程知识的积累和应用。应该说在这个阶段自我感觉很充实，好像有做不完的事，程序设计水平还处在语言级阶段。

第四阶段，随着积累了一定编程技巧之后，我开始想这样的问题：我是不是最好的程序员？我能否编写出最好的程序？这个过程是一个反思的阶段。我对自己的要求是：不但要会编程序，而且要编好程序，从关注程序数量开始转向关注程序质量。

第五阶段，开始在提高自己的软件开发水平上做文章。经过各种系统开发，尤其是大型系统的开发，发现了软件中有许多功能是重复的。因此，有一段时间把精力花在编制各种库函数上，通过不同系统调用相同的函数，以便减少重复开发，实现功能共享。当时比较得意的是库函数不是我一个人在调用，而是整个项目小组都在调用，甚至不同的系统也能调用，从而体会到编写库函数特别有价值。这个阶段的标志是库函数，程序员水平上升到库函数那一级。

第六阶段，到了库函数那一级后，很快就发现，单单实现程序函数级的调用是远远不够的。当你做了很多项目，包括大项目和小项目，尤其是做过跨行业的项目之后，你就会把库函数的共享思想用于项目开发。你就会想这样一个问题：为什么不同项目不能有相同的架构？如果有相同的架构，那么开发就有了相对的标准，我们就有可能通过配置的方法实现相同架构的系统。于是我提出了 IASG（交互式软件自动生成器）思想，并在 C 语言和其他一些语言中实现了 IASG 实例。记得最快的一次是编写一个系统（公安部门的自行车信息管理系统，主要用于丢失自行车信息登记）只用了 3 个小时（从需求到安装盘）。这个事情对我影响很大。我在这个阶段上升了一个很大的台阶，从程序上升到软件。核心思想就从库函数共享上升到软件共享。具体过程是建立一个通用的系统架构，架构中有许多共同的功能，例如，参数设置、用户权限管理、库表管理等。另外还提供信息建立查询开发模板，通过配置和特殊功能的编制就能很快完成了一个系统的开发。现在想起来 IASG 距离我已经有 20 年了。

第七阶段，到了 IASG 阶段后，我发现无论技术如何提高，都无法改变开发落后于需求的现实。通俗地说就是：程序员水平再高，仅仅是拉车水平高，但是，应该在什么路上拉车程序员并不知道。如果这条路是一条光明的路，则程序员越拉越有劲，有前途；如果这是一条死胡同，则程序员白费工夫；如果这是一条漫长的路，前途不明，则程序员可能要累倒在路上。现实中程序员水平低、收入低；系统需求不明确，系统开发周期一拖再拖；系统重复开发多，信息甚至不能在一个企业内实现共享，更不用说在企业之间、行业之间实现共享了；各种企业级的软件 ERP、CRM、BI

层出不穷，也没有哪个能满足中国的市场；各种新技术、新概念不断出现，却没有哪种技术或概念能真正发挥其内在价值，最终还是处于被学习、被运用的阶段。

这个过程是程序员脱离技术本身，开始思索、开始求源的阶段。在这个阶段的程序员的思想有了质的飞跃。以前光拉车不看路，现在要抬头看路了。

第八阶段，有了抬头看路的想法，于是我踏上寻路征程。我首先弄明白了我们脚下的路是什么样的，为什么这条路那么不平坦、不宽广。从软件生命周期来看，软件主要由用户需求发起，用户需求是软件生存的根本理由。由于企业、用户的不同而导致不同的需求——大量的无序的需求，这种需求驱动方式必然造成了我前面介绍的各种现象。这个阶段是寻找根源的阶段。只要我们找到了根源，就可以有机会解决问题。这个过程相对来说比较困难，这不仅需要编程技术，还需要很多方面的知识。若要了解这个根源，就迫使你学习和积累更多程序以外的知识。

第九阶段，当我找到软件是需求驱动方式之后，就开始考虑什么是用户需求？用户为什么要提出这些需求？我们可以更深入地分析用户需求产生的根源，我们能否让无序需求变成有序需求呢？当然针对这些问题我们都进行了深入分析，其过程也很难在这里展开说明。我只能说，最后结论是用户的需求来源于企业的经营。很多人思考问题还是就需求而论，并没有站在企业经营角度去考虑问题。千万不要小看这个变化，这个变化最终会产生一个理论。于是我们尽可能地站在企业经营角度看待企业经营方式、企业管理、企业信息化等。但是，我们最终要解决企业经营这个概念问题，如果我们都不能明确企业经营这个概念，或者我们不能科学地定义企业经营这个概念，那一切基于企业经营的各种具体现象就如同无本之源一样无序泛滥。就像ERP、CRM等所谓企业信息化产品一样，由于没有一个企业经营定义的支撑，只能就企业经营的某个方面提出解决方案。这些产品不缺乏需求的支持，缺乏的是最基本的企业经营定义的支持。而这个概念就是EOM。

EOM 是从定义企业经营角度入手，把我们今后要开展的各种研究和开发活动都放在一个理论可支持的基础上。只有定义了企业经营之后，我们才有可能分析我们需要什么软件，我们的软件采用什么技术才能实现企业经营的目标。而程序员则通过 EOM 了解到企业经营需要什么样的软件，这个软件有多大的价值，这个软件采用什么技术才能实现，自己要提高哪方面的技术水平才能获得更大的价值。

这个过程就是 EOM 阶段，通过 EOM 了解软件的根源和有价值的软件所在，进而选择自己未来的方向。

第十阶段，当我建立了 EOM 之后，便开始了 EOM 实现阶段。这个实现阶段分为两部分，通过这两部分的结合，我们就可以逐步看到 EOM 软件产品的实例，看到 EOM 的真正价值。

第一部分是 EOM 的业务实现。当我们明确了 EOM 之后，就可以根据 EOM 来重新规划企业信息化的整体架构，可以细分这个架构中的各种平台产品、通用产品、专业产品，可以细分出这个架构实现的各种技术架构和实现手段，可以细分出这个架构中的各种标准功能和标准信息。通过这样的分析，我们的程序员就可以根据自己的特长和爱好以及价值的判断来选择其中的软件产品和技术。在明确目标和方向的情形下，通过自己的努力，不断提高自己的各种技能水平，让自己的价值和企业经营价值有机地结合在一起，从而实现自己的理想。

第二部分是 EOM 的技术实现。有了 EOM 并根据 EOM 理论构建企业信息化的架构后，我们就必须从技术上实现这个架构，否则这个架构将永远停留在理论阶段，不具有可行性。我们可以采用现有的各种技术来实现这个架构，但是，现有的技术都是基于原有的业务需求而建立和发展的，它适用于原来的应用对象。目前的 EOM 是一个全新的企业经营理念，因此，我们必须建立一种新的软件架构来适应和最好地实现这个理念。幸运的是，我们找到了称作 NSS（New Software Structure）软件新架构的技术，该技术体现了适应企业经营发展方向，将软件合理分层，用最新的软件技术按照架构的方式规范软件开发的模式，可以实现最大范围的功能共享，

实现软件的可扩展性。

这个阶段可以让程序员在软件产品业务设计或软件产品技术实现上等多个方面进行深入钻研，并且成为领域专家。这和我们平时涉及的简单的需求分析和简单的技术实现有着本质区别。

从我的程序员经历可以看出，程序员的成长是无止境的，只要有的放矢地努力，就会一步步登高向上。我认为程序员成长经历主要有三大阶段，即通用技术阶段、市场阶段、专业技术阶段。

1) 通用技术阶段是程序员专注编程水平提高的阶段，也就是说“只拉车不看路”阶段。这个程序员能做的事情那个程序员也能做，程序员的替代性很强，程序员市场价值相对较低，程序员只关注编程技术本身。

2) 市场阶段是程序员跳离技术层面开始考虑为什么要开发这个软件，这个软件有什么价值的阶段，通过求软件之源来重新认知自己的方向。

3) 专用技术阶段是程序员认知了这个软件和技术有很大的市场价值，全身心投入到这个领域中去，并在这个领域成为专家的阶段。程序员不但要懂技术，更要懂得客户业务，不同的程序员的技术和业务变得没有可比性，这种稀缺性造就了程序员极大的价值。

这三个阶段其实就是三个过程，每一个过程都是一次飞跃。程序员知道自己可以飞多高，依靠的是程序员的学习和眼界；而程序员能飞到哪里，那就要靠程序员自身的努力。一个程序员可以没有能力，但是不可以没有眼界。

人 门 篇

1.1 程序员的梦想——中国的比尔·盖茨

作为一个 IT 行业职员，我经历过一个普通程序员成长的过程，同时也接触过许多不同层次的程序员。他们或在我身边匆匆而过，或与我共同工作，或在我可以关注的范围内成长着。他们的喜怒和哀乐、挫折和成功、幻想和现实、希望和失望，无不与我心共振。我知道这个行业从业人员的梦想，也知道这个行业的残酷。无数人怀着希望而来，却抱着无奈离去。我早就有和他们共语的愿望，希望通过这个主题和他们交流程序员所关注的各种问题，希望我的经验有助于他们的成长，同时我也想谈谈 EOM 对程序员的真正价值的影响，以及如何实现“成为比尔·盖茨”这个程序员的最高梦想。

什么是程序员？什么人能称得上是程序员？会编程序的人都是程序员嘛！这个问题看似简单，但仔细想一下，也很难回答。其实在中国，关于程序员的称呼有很多种近似的叫法，例如“开发人员”、“编程人员”、“计算机人员”等。只是现在分工越来越细、专业化程度不断提高的情况下，程序员这个词才逐渐地流行起来。

那么什么人才算是程序员呢？现在看来凡是从计算机专业或相近专业毕业的、以编写程序为职业的人都可算得上是程序员。但是在 20 世纪 80 年代到 90 年代，由于计算机还是新生事物，整个社会对其有种神秘的、高贵的、不可触及的印象。加之当时计算机人才少之又少，除了计算机专

业从事开发工作之外，很多非计算机专业的学生，甚至初、高中生也加入到计算机开发队伍之中。他们充满激情，敢于学习，勇于探索，其中有许多人很快就成为开发队伍中的主力军，成为编程人员中的佼佼者。有的时候，专业的程序员还不如业余的程序员，这种情况比比皆是。那个时候，专业的程序员还不如业余的程序员，这种情况比比皆是。那个时候，开发环境、学习环境要比现在差得多，程序设计语言比较单调，技术书籍更是少之又少。记得当时只能把单位印制的 8086、Z80 等汇编程序设计资料当做教材，用 debug 把操作系统中的代码打印成厚厚书籍来阅读。由于当时我的单位是生产（组装）计算机的，因此，使用计算机还是比较方便的，但是，不像现在，绝不可能在家里使用计算机的。

由于那时程序员可以触及计算机，可以看明白别人不懂的代码，可以让计算机执行自己的指令，这让许多外行甚至内行人很羡慕。

程序员一般只掌握单一的程序设计语言，比如编写汇编程序的程序人员，编写 C 语言程序的程序人员，编写 Unix、XENIX、AIX、SCO、HP-unix 等 Unix 类的 shell 程序的程序人员，编写与数据库打交道的 proc c 程序人员，编写 C++ 程序的程序人员，编写面向对象的 VFP、VB、Delphi、PB 的程序人员，编写 Web 程序的 HTML、CGI、ASP、PHP 的程序人员，编写 C#、Java 的程序人员。这些程序语言有些保留了下来，有的则被时代无情地淘汰了。同样是程序员，如果不能适应语言的发展也避免不了被淘汰的命运。

我自己认为的程序员与其他职业人员之间的区别：

1) 因为比尔·盖茨是编程序的，所以似乎每个程序设计人员都有一个“比尔·盖茨”梦想：比尔·盖茨能做的，我也许能做到，即使做不到，做到一半也是不错的。这个潜在的意识是程序员最大的财富，许多程序员成了有理想、有抱负的人。我想很多人选择 IT 大都与此有关吧。

2) 工作成果完全由自己把握，随时编随时运行随时出结果。这种自我感觉是很多职业都不具备的。因此，程序员有很强的自信心。而且这种自信心往往可以使得程序员产生自己开公司的念头。

3) 有很高的预期价值。软件通过使用创造了价值，程序员通过制作产生了软件。因此，程序员往往把软件的价值看做自己的价值，例如一个软件卖了5万元，程序员就会把自己的劳动价值估算在5万元以上；如果这个软件有100个潜在的市场，那么程序员就会把这个价值升值到 $5 \times 100 = 500$ 万以上。所以在程序员这个群体中，很多人都相信自己未来能够获得更多的收入。

当然，程序员也有其他一些特点，例如：有的程序员喜欢晚上干活，白天睡觉；有的喜欢钻研，连续加班；有的头脑灵活，动手能力强；有的喜欢追逐最新技术，变成别人公司的代言人等。

我注意到了有这样一点，那就是现在的程序员已经失去了神秘感，以往给人以仰目而视的形象正逐步走下神坛。

这是程序员职业发展的必由之路，抑或是再正常不过的社会现象？

1.2 谈谈程序员的基础知识

对于程序员需要具备哪些最基础的知识 and 技能这个问题，不少刚从大专院校毕业出来的新入职员工，甚至是从事过一段时间编程工作的程序员，都是比较模糊的。只有认真掌握一些基础的知识 and 技能，才能走上程序员这条大道。

说句实在话，我在从事编程工作很长一段时期内都没有关注这个问题，基本上是边编边学，边学边编，从学习中积累，从编程中积累。除了和同事进行工作上交流之外，学习材料很少，基本上是一个人在战斗，也不知道自己是不是成为了合格的程序员。也许是受那个年代所限，当时的程序员人数很少，因此同行间没有什么竞争，有了位置就不怕失去。要是放在现在，真的很后怕。

回到正题，我认为程序员在最初阶段要从流程、语法、调用三个层次要求自己。

1. 流程

这是对程序员最基本的要求，这个层次就是要求程序员能够把一个最简单的程序编辑、编译、运行成功，强调的是掌握编程的环境和流程。

在这个层次上又分三个方面的要求：

(1) 计算机基础知识

我见过许多程序员新手，他们都自称学过计算机基础知识。但实际情况是，学而不致用，学的概念太多，自己却理不出头绪，和实际工作对不上号。有的甚至认为自己忘了，什么都不知道。但是我要提醒新手的是，尽管基础知识十分丰富，但是有关编程的一些基础知识和概念是必须掌握的。

1) 操作系统

什么是操作系统？你所要编写的程序在什么操作系统上运行？目前主要有 Windows 类、Unix 类、Linux 类操作系统。每种操作系统对编程的影响是不同的。

2) 计算机、内存、硬盘

这些概念对编程来说也是最基础的，例如计算机分为 PC 机、小型机、大型机。在 PC 机上编程和小型机上编程是有差别的。程序设计语言安装时也要注意内存大小和硬盘大小。

3) 目录、文件

这些是最基础的概念了！一定要掌握和理解。因为你编写的程序就是一种文件，而且要放置在指定目录下。

4) 程序设计语言、程序、编辑、源程序、编译、可执行程序、运行

这些概念也是最基础的。不同的程序设计语言对编程具有很大的影响。