

1.0版本

# Programmers' Guide 程序员生存手册





---

**Programmers' Guide**  
**程序员生存手册** 1.0  
版本

---

# 前言

各位程序员朋友们,欢迎大家来到阿道的世界。

我叫阿道,出生于黄海之滨——青岛,今年已经10岁啦。据易企天创的大家说,之所以要为我起“阿道”这个名字,是因为希望我能秉承“禅道”的精神,做一名朝气蓬勃的禅道人。所以为了不辜负大家的期待,阿道也一直在朝着这个目标前行。今年我想要为大家准备一本可读性高的手册。



在IT行业中活跃了这么久,阿道也接触过很多IT从业者,他们来自各个行业,有做游戏研发的,有做Web应用开发的,还有做系统嵌入式研发的……这些IT从业者在各自的领域中熠熠生辉,但无一例外地,都或多或少地为职业规划所困扰。

瞄准这一现实问题,阿道决定将手册的内容主题定为“生存”,生存就意味着不仅要实现生活质量上的丰富,更要实现工作质量的深化。顺理成章地,手册的名字就定下来了——“程序员生存手册”。

在手册中,阿道分别从技能篇、方法篇、发展规划篇、资源篇、管理篇五个部分来介绍相关的知识,涵盖了每位程序员从初入行业到准备转型,直至最终实现自己人生价值的全过程。希望能为阅读到这本手册的每一位程序员提供一些提高能力的新途径,以及进行职业规划的新思路。

好了,阿道不再啰嗦了,我们在正文中见哦~

# 目录

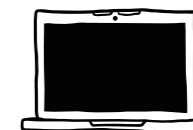
## 第一部分:技能篇 03

1.1 编程语言	04
1.2 英语	05
1.3 Linux	06



## 第二部分:方法篇 07

2.1 敏捷开发	08
2.2 CMMI	15



## 第三部分:发展规划篇 17

3.1 研发人员	18
3.2 产品经理	18
3.3 测试人员	19
3.4 项目经理	20

## 第四部分:资源篇 21

4.1 书籍推荐	22
4.2 网站推荐	27
4.3 工具推荐	33

## 第五部分:管理篇 34

5.1 健康体魄	35
5.2 自我认知	37
5.3 主动学习	38
5.4 沟通能力	39
5.5 领导能力	40
5.6 决策能力	41
5.7 时间管理	44



## 第一部分：技能篇

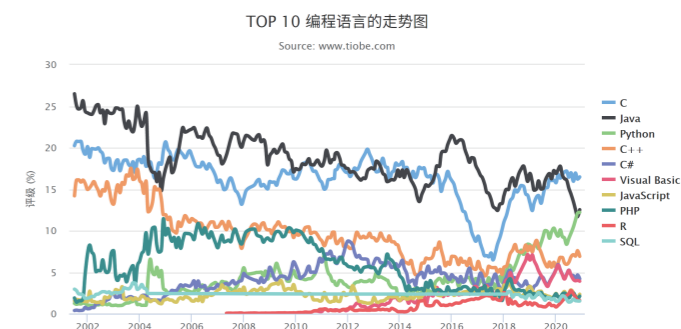
程序员的成长就像我们玩游戏打怪升级一样，想要开更高级别的副本，执行更高级的任务，就需要不断地历练，给自己增加技能点，提升等级。那么，快来点亮你的技能点吧。

## 1.1 编程语言

编程语言是程序员赖以生存的基础，因此程序员有必要拓展自己舒适圈边界，让自己“胖一圈”。尤其在当下 IT 技术更新迭代迅猛的大环境下，程序员只有不断地投资知识、管控风险、提升价值，才能确保在竞争越发激烈的职场中长期立足。

### 1.1.1 每年学习一门新语言

学习不同的编程语言，有助于让我们拓宽思维，用更多的不同方案来解决同一个问题。学哪门语言呢？右图或许能给你一些启发。



### 1.1.2 每月读一本书

很多编程语言都在不断地更新完善，例如 PHP 已经更新到 PHP8 版本。所以，我们需要不断地学习。没时间读书？建议大家可以充分利用每天的“碎片时间”来阅读，比如上下班路上、高铁上等。

时间就像海绵里的水，只要愿挤，  
总还是有的。

——鲁迅

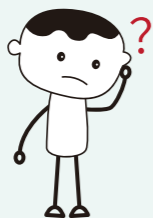


### 1.1.3 不断尝试、与时俱进

如果你只在 Windows 下工作，可以尝试折腾下 Linux，玩玩 Vim。关注周边最新的技术动态，这样在学习工作中总会不断有新的发现和惊喜。

## 1.2 英语

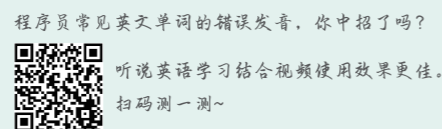
对于所有程序员而言,无论你主攻哪种编程语言,有一种语言是每个人技术生涯发展道路上必不可少,且更重要的——英语。



官方一手英文资料看不懂?  
这个变量怎么命名更合适?  
要是能和国外大牛一起交流学习该多好!

### 1.2.1 积跬步以至千里

平时只要遇到不认识的英文单词或语法,马上查词典。遇一个学一个,长此以往,我们每个人都可能成为团队里“行走的小词典”哦。



### 1.2.2 坚持

每天坚持阅读一小节技术相关的英语文档、书籍。刚开始时可能阅读速度会较慢,但不必着急,只要坚持,速度会很快提升。



小秘密: 阅读一本英文技术书籍, 比阅读同样篇幅的新闻、小说等简单得多。

## 1.3 Linux

无论主攻哪种编程语言,也无论采用哪种技术架构,学习使用 Linux 系统不仅可以扩宽我们的见识,还有助于提升我们的工作效率与质量。

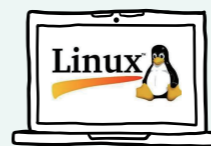
### 1.3.1 拥抱开源

Linux 系统是自由、开放、开源软件的典型代表。学习 Linux 系统,来更多地了解、接触开源社区,理解开源精神,与众多开源爱好者们一起为开源社区做贡献吧。



### 1.3.2 能用键盘就不用鼠标

使用 Linux 系统最大的特点莫过于“离鼠标越来越远”。建议逐步学习掌握 Linux 系统的常用操作命令,同时,学习使用 Vim 神器,长此以往,双手就可以从鼠标上解放,并提升效率。



### 1.3.3 自动化、效率提升

建议学习、掌握 Shell 语言,它可以帮助我们将平时复杂、重复性的任务,通过写 Shell 脚本工具,实现自动化,以提高效率和产出。



## 第二部分:方法篇



恭喜你已经完成技能篇的通关，但在程序员的修炼之路上，只靠个人技能buff的加持是不能最终登顶的，毕竟孤军奋战不是我们的传统，团队合作才是王道。

## 2.1 敏捷开发

目前，以敏捷为基础的软件开发方法论已形成一个庞大的体系，主要是以一种更加简洁、可持续、短周期、高效率的方式进行软件开发。其中包括 Scrum、XP (极限编程)、Kanban (看板)、规模化敏捷等。接下来，阿道会向大家依次介绍。

### 2.1.1 用户故事与 Invest 原则

在介绍诸多敏捷方法之前，我们需要先了解一个重要概念——用户故事。用户故事是敏捷方法的重要基石，早在1996年，极限编程就提出了“故事” (Story) 的概念，这是用户故事的起源。无论是 Scrum 还是看板，甚至是规模化敏捷 SAFe 等，在需求领域，最终都会落在用户故事上。

这里和大家分享比尔韦克 (Bill Wake) 的 Invest 原则，它可以帮助我们提高用户故事的质量。

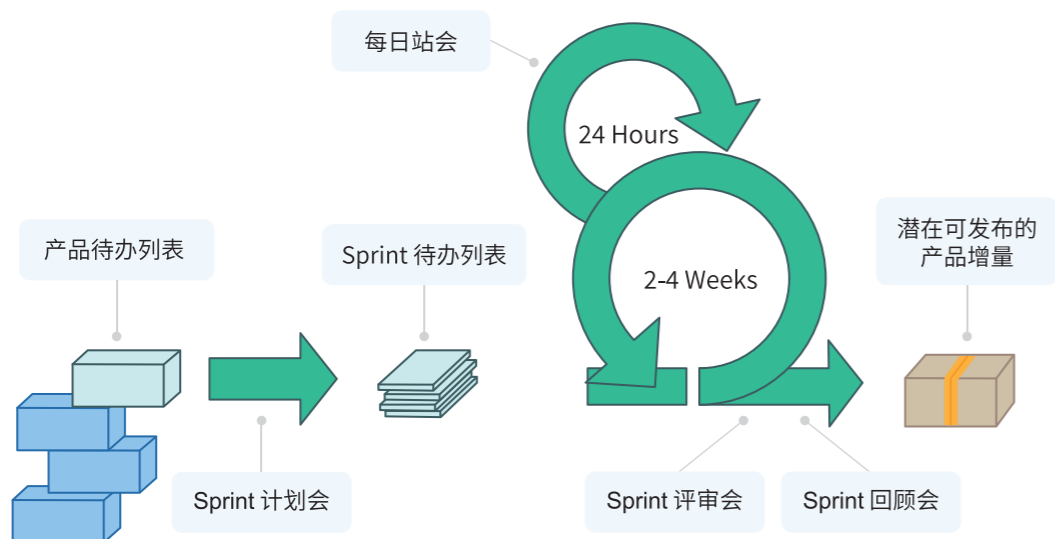
实施 Invest 原则，用户故事应该如下表



字母	原则	释义
I	独立的 (Independent)	一个用户故事所描述的特性不依赖其他用户故事。
N	可协商的 (Negotiable)	无需太过详尽，为细节留出讨论和补充的余地。
V	有价值的 (Valuable)	用户故事是为了给客户展现产品价值；它描述的是特性，而不是任务；它使用客户语言，且解释简单明了；使用产品的人群都能理解这个用户故事。
E	可估算的 (Estimable)	用户故事是描述性的、准确的以及简练的，以便于开发者能大体估算创建用户故事中的功能所必须的工作量。
S	小型的 (Small)	计划和精确估算较小的用户故事会更加轻松、高效。
T	可测试的 (Testable)	你可以轻松验证用户故事，并且能得到明确的结果。

### 2.1.2 Scrum

在敏捷开发的方法论体系中，Scrum 高居敏捷方法论的首位，也是团队转型敏捷的首选。但 Scrum 究竟是如何解决软件开发过程中的沟通、协作问题的？



敏捷作为一种软件开发运动，其发起人试图以一种更为敏捷的新方式来思考软件开发方法论以及组织架构，从而帮助行业中的人们更好地进行软件开发。而其中稳居 C 位的 Scrum 方法论，关注每一 Sprint 结束后的成果交付，促使每一团队在现有框架的基础上，能够根据自己团队的实际制定合适的 Sprint 任务，提高工作效率、扩大反馈。

在以 Scrum 为核心的软件开发过程中，以下五个会议能够帮助团队高效沟通、及时改进，也是需要团队格外注意的。让我们一起来看看会议流程：

名称	产品待办列表梳理会议 Product Backlog Grooming	计划会议 Sprint Planning	站立会议 Daily Scrum	评审会议 Sprint Review	回顾会议 Sprint Retrospective
目的	梳理产品待办事项列表，整理出1-2个Sprint满足“就绪定义”的待办项。	为接下来的Sprint对齐目标，承诺需要完成的故事，并做出计划。	更新Sprint目标达成状态，并根据当前进展重新计划剩余的工作。	检验所交付的产品增量并按需调整产品待办列表。	回顾团队的迭代工作，找出改进事项，为将来的改进制定计划。
阶段	贯穿在整个Sprint中间的活动，为下一个Sprint做好准备。	Sprint开始的第一个事件。	Sprint中的每天。	Sprint结束前的倒数第二个事件。	Sprint中的最后一个事件。
时长	不超过团队一个Sprint时长的10%	每星期的Sprint不超过2小时	小于15分钟	每周的Sprint不超过1小时	每周的Sprint不超过45分钟
人员	开发成员、产品负责人、Scrum Master	Scrum Master、开发成员、产品负责人	开发成员、产品负责人、Scrum Master	开发团队、产品负责人、其他利益相关者	Scrum团队成员
议程	<ul style="list-style-type: none"> <li>讨论需求，拆分、澄清用户故事；</li> <li>完善验收标准；</li> <li>对故事的优先级进行排序；</li> <li>估算故事点。</li> </ul>	<ul style="list-style-type: none"> <li>明确迭代目标；</li> <li>划分任务优先级；</li> <li>拆分细化任务；</li> <li>确认任务完成的定义；</li> <li>预估工时；</li> <li>确认Sprint Backlog。</li> </ul>	团队各成员依次进行陈述，包括且不限于： <ul style="list-style-type: none"> <li>我昨天完成了什么；</li> <li>今天的计划；</li> <li>遇到的困难或障碍。</li> </ul>	<ul style="list-style-type: none"> <li>团队介绍迭代成果；</li> <li>演示新功能；</li> <li>依据Sprint目标对项目进行评估；</li> <li>调整产品Backlog。</li> </ul>	<ul style="list-style-type: none"> <li>回顾上次会议改进结果；</li> <li>概括性总结迭代；</li> <li>总结量化结果；</li> <li>总结迭代过程；</li> <li>做会议纪要及迭代总结。</li> </ul>
产出	梳理好的产品待办列表。	Sprint目标；Sprint Backlog。	对团队成员工作及工作项状态形成共识，并对后面的计划进行调整。	收集用户或干系人对产品的反馈，以便后期进行调整。	一个排列好优先级的改进代办列表，最高优先级的改进项应该放入下一个Sprint Backlog中。

欢迎扫码观看我们自制的 Scrum 小视频！



### 2.1.3 极限编程

Scrum 和 XP 没有高下之分,若将两者结合,那一定能发挥出更大的力量。前人对 XP 方法论总结出了什么样的实践呢,跟着阿道一起来看一看吧。



### 2.1.4 看板

在团队转型敏捷的道路上,有很多阻碍转型的因素。而在这一过程中,看板能够通过其简单、高效、灵活、改善工作流程等优势,帮助团队排除重重阻碍。



我们在设计看板的时候,要注意几个小贴士:

#### 可视化原则

工作状态可视化:以等待、处理中、已完成三个元素呈现工作状态;  
任务项可视化:成员任务处于团队的迭代周期中,任务、时间、工作量透明且直观;  
工作流程可视化:通过任务状态的变换完成工作交接,拉动价值从左向右进行流动。

#### 流动原则

看板系统强调端到端的价值流动,我们以一个产品开发过程为例制作看板,看板的任务项要包括:开发、测试、待发布、发布、已发布,其中开发又可以划分为实现、评审、完成。  
这样从用户需求开始,最终以价值交付为结束的开发过程,实现了价值的流动,并且价值最终又流回用户手中。

#### 拉动原则

看板通过看板传递指令,由后面的流程拉动前面流程的生产,实现清晰、有序的生产管理,最终拉动价值流动,追求团队、用户双方满意的产品质量。

#### 协作原则

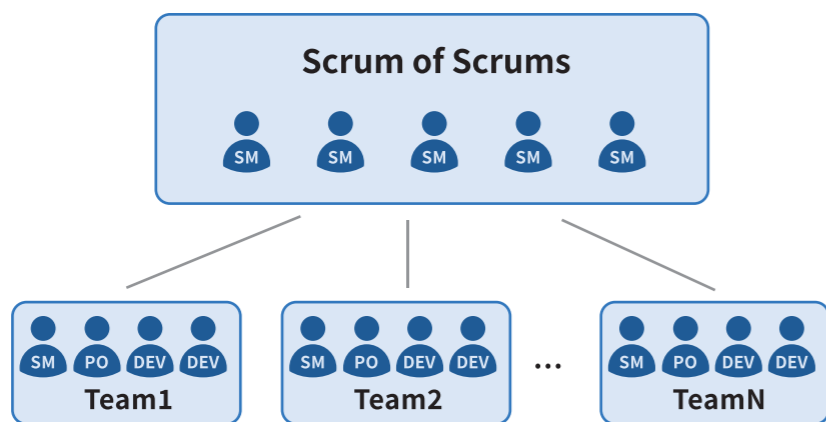
看板要求限制 WIP (在制品),在团队协作的过程中,如果同时需要拉动多个价值的流动,会导致每个人的精力过于分散。在对任务划分优先级的时候,团队需要有这个意识——集中力量解决优先级较高的任务,保证需要快速解决的事情被优先处理。

### 2.1.5 规模化敏捷

在软件开发行业中,一般情况下,敏捷团队通常为5—12人,但在大型企业中,开发团队往往会有几十乃至上百人,这时候要怎样敏捷起来呢?

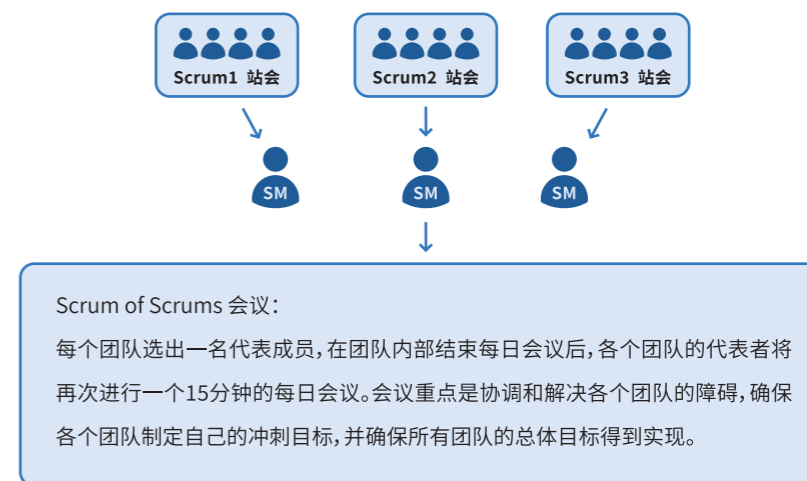
#### Scrum of Scrums

Jeff Sutherland 在《规模化敏捷:五家公司的Scrum创新实践》一文中对此作出了回答,他首次提出了 Scrum of Scrums 这一规模化敏捷方法。



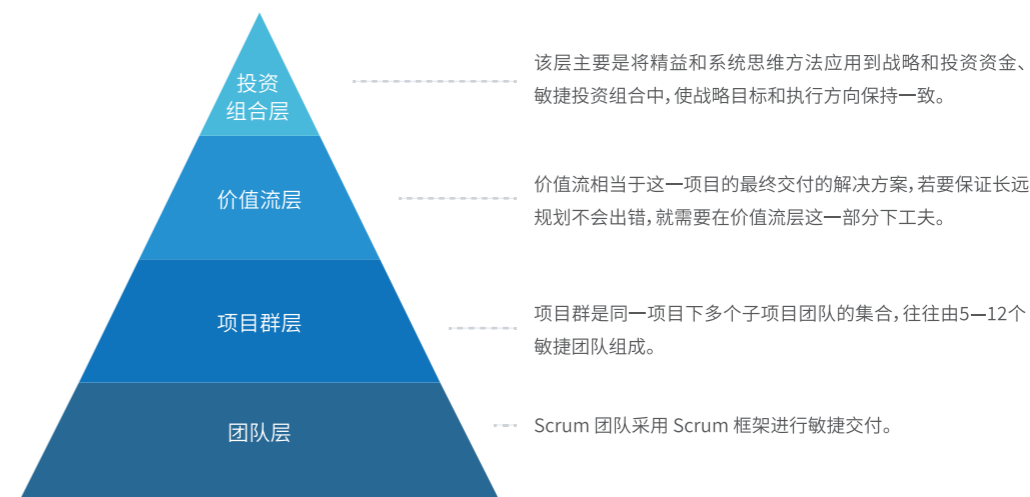
Scrum of Scrums 主要集成同一项目中多个 Scrum 团队的工作,通过召开 Scrum of Scrums (以下简称“SoS”)会议消除障碍,确保提高各个团队输出的协调性。

SoS会议流程如下:



#### SAFe

说到百人团队,就要涉及管理层面的概念,如何在保持敏捷的状态下去管理这上百人的团队,成了管理者亟需解决的问题。而 SAFe 就是以敏捷为基础,糅合管理知识而成的规模化敏捷框架。



## 2.2 CMMI

CMMI 是软件能力成熟度模型集成 (Capability Maturity Model Integration), 是由美国卡耐基梅隆大学软件工程研究所组织全世界的软件过程改进和软件开发管理方面的专家历时四年而开发出来, 并在全世界推广实施的一种软件能力成熟度模型, 主要用于指导软件开发过程的改进和进行软件开发能力的评估。



CMMI 实际上是一种管理流程的标准化, 其主要目标是创建“产品超前、服务高效、部门高产的可靠环境”。具体来说, CMMI 的目标包括: 提供高质量的服务或产品、提高客户满意度、增加利益相关者价值、实现全行业的认可、建立更大的市场份额。

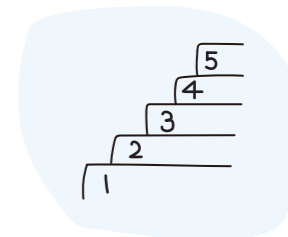
卡耐基梅隆软件工程研究所指出, CMMI 的目的是帮助“整合传统的相对独立功能、设定过程改进目标和优先级, 来为质量过程提供指导, 并为当前过程提供参考”。

### CMMI 能带来什么价值



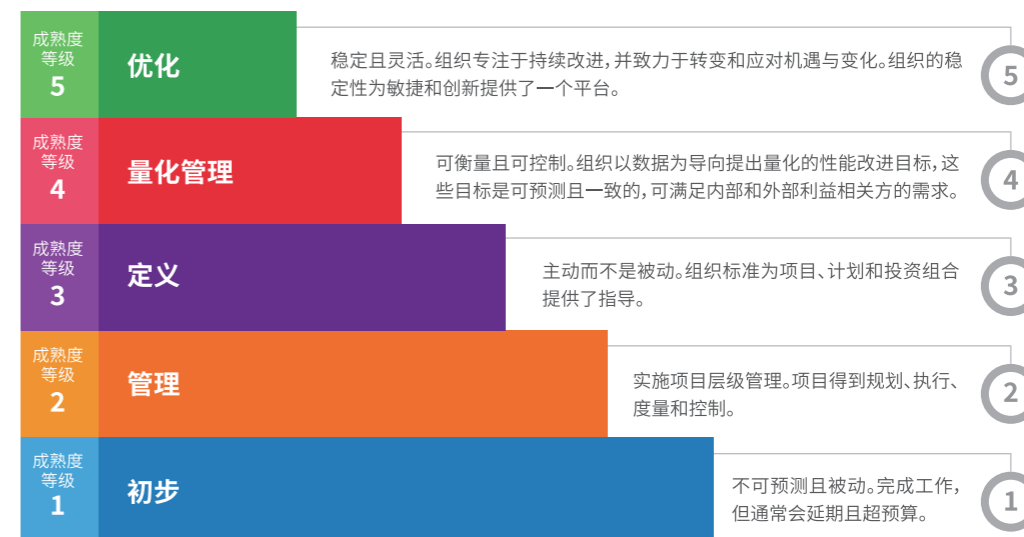
1. 对项目开发过程进行规范, 保证软件开发的质量与进度。同时, 规范化的流程也有助于提高软件开发者的职业素养, 养成好的开发习惯, 员工做事方法将变得标准化、规范化。
2. 降低产品开发成本, 提高项目控制能力。流程规范后, 产品质量随之提高, 产品缺陷率也大大降低, 客户满意度提升, 后期修补和维护的成本大大降低。

3. 提升企业的竞争力。CMMI 是世界范围内认可的软件能力评估标准, 越来越多的大型企业要求其工程承包商具有一定的 CMMI 级别。级别高的企业在项目的竞标中具有一定的优势。



### CMMI 的级别

CMMI 成熟度级别提供了强大的标杆评级方法, 能把被评估公司的能力与竞争对手、业界和自己的历史进行对比。CMMI 共有5个成熟度级别, 代表软件团队能力成熟度的5个等级, 数字越大, 成熟度越高, 高成熟度等级表示有比较强的软件综合开发能力。5个成熟度级别提供了一条清晰的改进路径, 任何想提高自己能力的公司, 都可以通过识别改进点、解决这些改进点的问题、并把解决方案融入到整个组织的方式来达到更高的成熟度级别。

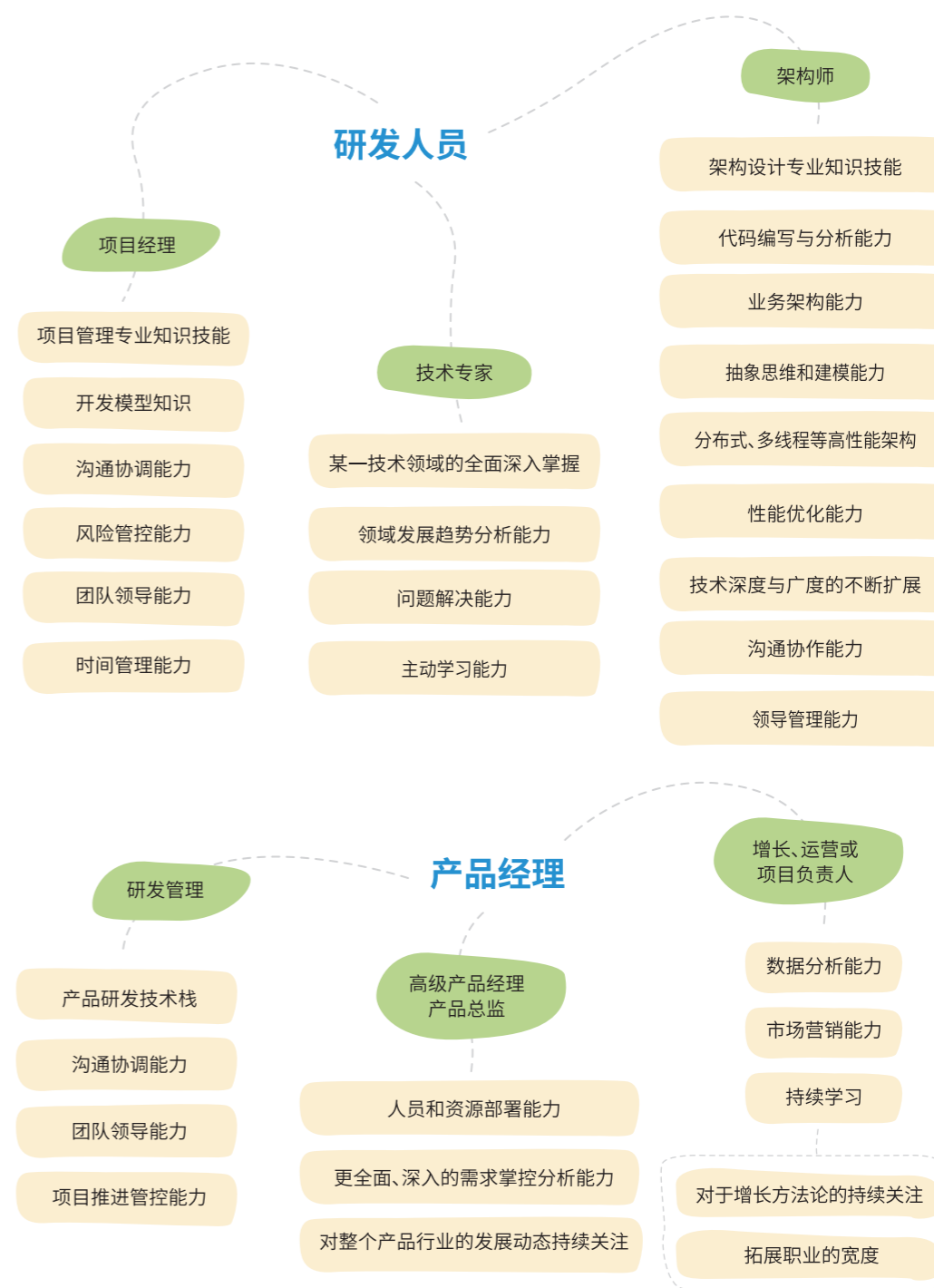


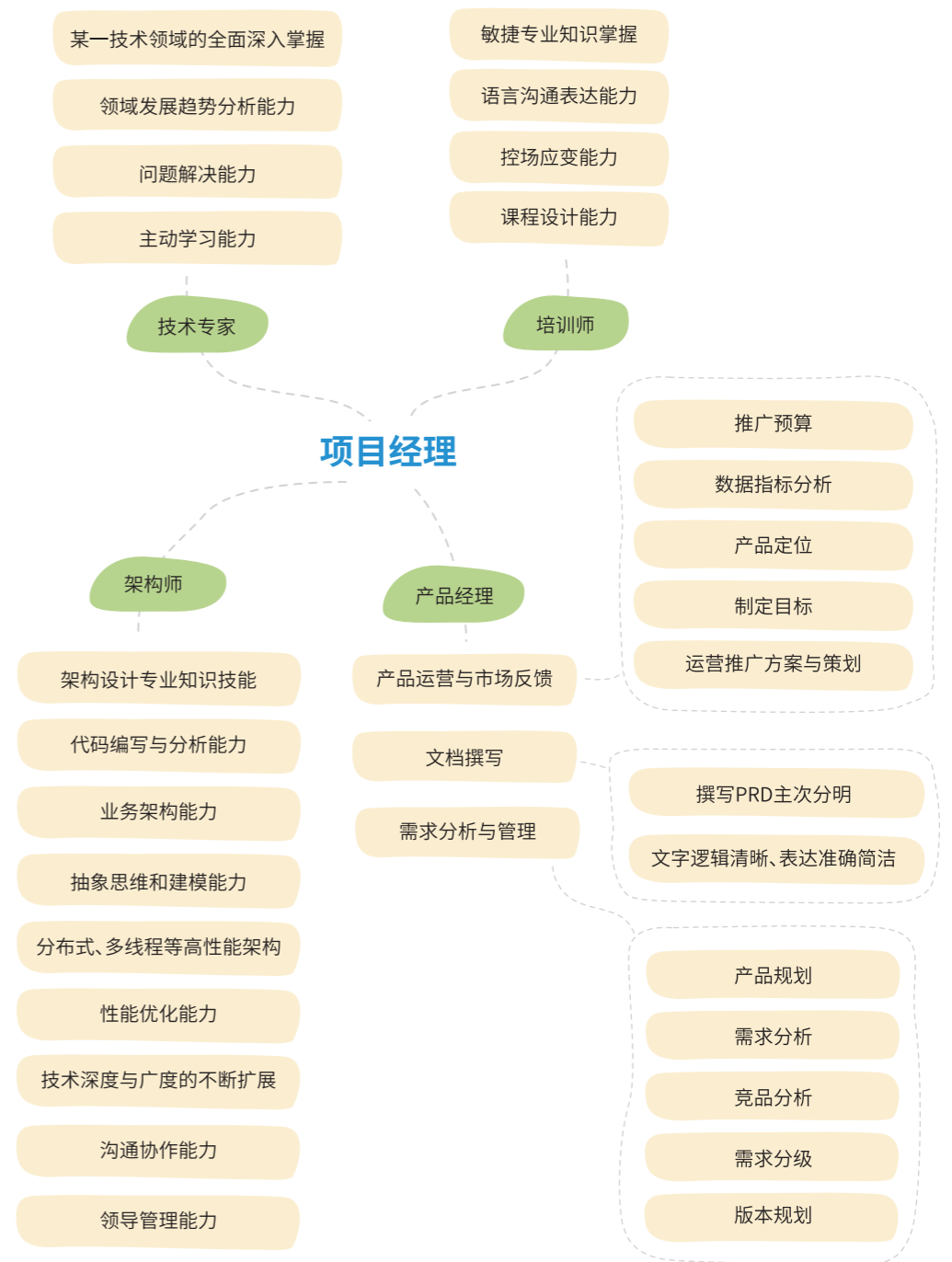
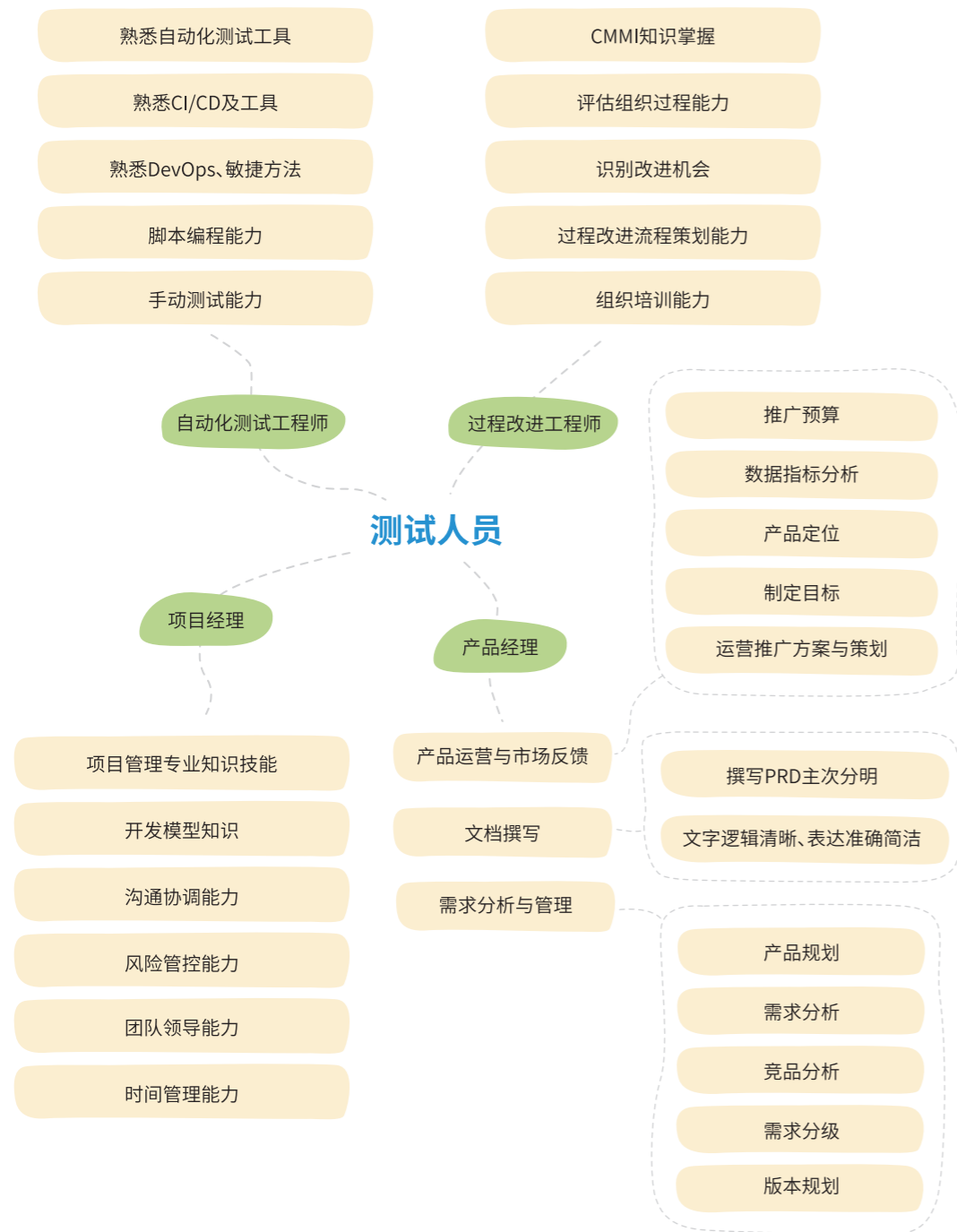
## 第三部分：发展规划篇



“出路在哪、何去何从、转行经验、失业、跳槽……”打开搜索引擎，映入眼帘的全都是这样的焦虑。无须讳言的是，35岁单纯 coding 能力可能确实不如刚出校园朝气蓬勃的自己。但是这不重要，因为编程在你整个武器库中已经不那么重要了，你的经验，你的视野，你的架构能力，你的管理能力，你分析和解决问题的能力已经远远不局限于技术这个领域。

下面的发展路线图与你共享。





## 第四部分:资源篇



前面说完了技能方法和路线，想必你一定跃跃欲试了。这不，从理论到工具，书籍、网站、课程一个都不能少，方法和落地一并给你！一起学习吧！

## 4.1 书籍推荐

### 4.1.1 读书的正确姿势

阿道该如何激励你读书，是要说“书籍是人类进步的阶梯”，还是说读书是一种快速进阶的方式呢？

读书的好处我们从小到大听过不少了，但道理我都懂，可臣妾做不到哇！尤其处于现在的信息时代，太多浮云遮望眼，找资料容易读书难。那么，阿道就从授人以渔开始，教你如何读书吧。



**推荐：**《如何阅读一本书》

**作者：**Mortimer J. Adler, Charles Van Doren

**译者：**郝明义，朱衣

**解析：**每本书的封面之下都有一套自己的骨架，这本书就是教你找出书的骨架。从书籍的层次、主旨到阅读不同体裁的书籍，这本书都从落地方面给到了切实可行的行动建议。不懂阅读或初探阅读的人，读这本书可以少走冤枉路，避免很多无效的阅读，节省大把的时间。



**推荐：**《如何有效阅读一本书》

**作者：**奥野宣之

**译者：**张晶晶

**解析：**对于一些强功能性的书，记笔记的方法决定着记得如何。这本书会教你如何用书一本笔记，将书中的精华用最高效的方式整理记录下来，更有“葱鲷(wěi)火锅式”摘抄法等很多有趣的笔记方法。

### 4.1.2 硬技能

硬技能的书里,除了讲编程的基础,还有很多提高代码可读性的小技巧、黑客的工作方法介绍,也有通过例子和有趣的类比,来讲软件开发不同方面的陷阱。



**推荐:**《编写可读代码的艺术》

**作者:** Dustin Boswell, Trevor Foucher

**译者:** 尹哲, 郑秀雯

**解析:** 插图幽默合理,问题直指要害,方法简单易行。这本书关注编码的细节,总结了很多提高代码可读性的小技巧,看似都微不足道,但是对于整个软件系统的开发而言,它们与宏观的架构决策、设计思想、指导原则同样重要。

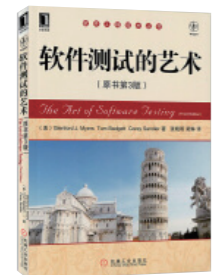


**推荐:**《代码整洁之道》

**作者:** Robert C. Martin

**译者:** 韩磊

**解析:** 本书适合有志于改善代码质量的程序员。书中介绍的规则均来自作者多年的实践经验,涵盖从命名到重构的编程全流程,遵循这些规则才能编写出干净的代码,从而有效提升代码质量。

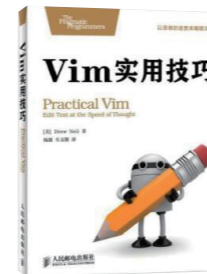


**推荐:**《软件测试的艺术》

**作者:** Glenford J. Myers, Tom Badgett

**译者:** 张晓明, 黄琳

**解析:** 这本书适合软件开发人员、IT项目经理等相关读者阅读,全面而细致地展示了久经考验的软件测试方法和智慧。如果你正在参与重要的软件项目开发,本书绝对值得仔细研读。



**推荐:**《Vim实用技巧》

**作者:** Drew Neil

**译者:** 杨源, 车文隆

**解析:** 阅读本书是熟练掌握高超的Vim技巧的必由之路。书内示例丰富,讲解清晰,可帮助读者快速掌握和精通Vim。

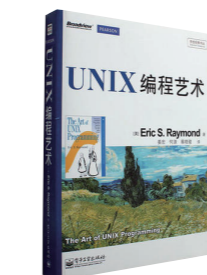


**推荐:**《Linux命令行与shell脚本编程大全》

**作者:** Richard Blum, Christine Bresnahan

**译者:** 门佳, 武海峰

**解析:** 看了这本书,你可以轻松写出自己的shell脚本。本书针对Linux系统的最新特性进行了全面更新,不仅涵盖了详尽的动手教程和现实世界中的实用信息,还提供了与所学内容相关的参考信息和背景资料。

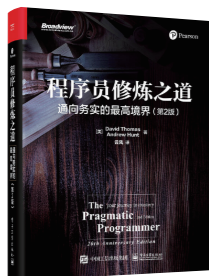


**推荐:**《UNIX编程艺术》

**作者:** Eric S. Raymond

**译者:** 姜宏, 何源, 蔡晓骏

**解析:** 这是一本关于黑客文化的书,书里没有数学公式,甚至连代码也没有。本书主要介绍了Unix系统领域中的设计和开发哲学、思想文化体系、原则与经验,覆盖面广、内容深邃,展现了作者极其深厚的经验积累和领域智慧。



**推荐:**《程序员修炼之道》

**作者:** David Thomas , Andrew Hunt

**译者:** 云风

**解析:** 技术是基础,还要辅以其他方面的素质才算从优秀到卓越。这本书可以作为程序员培养个人责任、职业发展、使代码保持灵活并易于改编等基本素质的参考。

另外,禅道正在拍摄《程序员修炼之道》系列的小视频,看倦了书籍,视频也是个很好的交替学习方式。扫码来看下!



### 4.1.3 软技能

在程序员的成长道路上,技能学习需要软硬兼备。程序员在职场上往往会出现很多困惑:为什么专业技能差不多的同事,几年下来,有的人职位晋升,有的人还在原地踏步?是什么造成了这样的差距?又如何解决呢?阿道着重推荐下面这本。



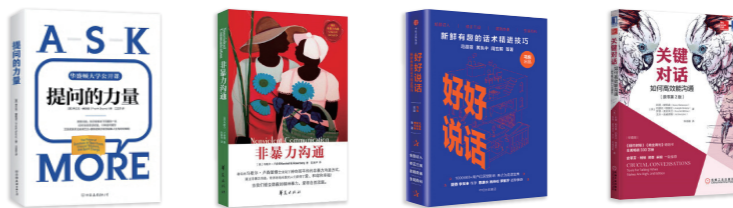
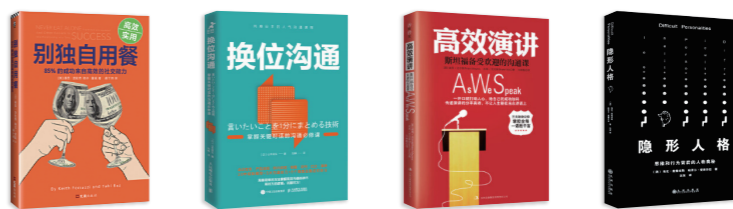
《软技能——代码之外的生存指南》书中,作者认为每个人要想在这个职场上不吃亏,最好让自己具备不可替代性。而这本书最精彩的地方,并不是一味地在理论上强调软技能有多重要,而是明确给出了培养软技能的方法,程序员们可以随时参照训练。

### 4.1.4 沟通表达类

阿道不能保证,看过这些书就能像呼兰、庞博一样从程序员成功转行脱口秀,但至少可以掌握一些沟通技巧,解决一些沟通中存在的问题,也能减少日常工作中的阻碍。

比如《换位沟通》中的“1分钟沟通方式”,讲了如何在1分钟内吸引对方注意力,并用对方能听明白的方式讲清这件事的沟通技巧。

**其他书籍推荐:**



阿道温馨提示: 想要了解以上书籍的详细内容, 可以扫描右方二维码一键访问哦~



## 4.2 网站推荐

学技术、卖代码、向大牛们讨经验、寻找项目机会哪家强?这些程序员必逛的国内外网站,一定要上去转转。

### 4.2.1 国外网站推荐

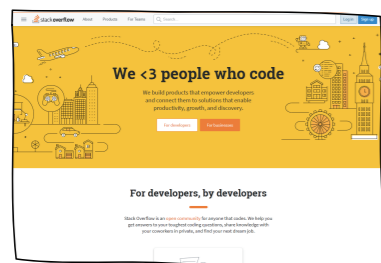


网站: 算法学习——LeetCode

网址: leetcode.com

人群: 需要刷题的程序员初学者

介绍: LeetCode 覆盖的编程领域比较多, 包括算法、数据结构、人工智能、系统设计等众多领域。它收集了最常见、最高频的编程题目, 配有算法学习书籍的配套题库, 每道题都有详细的解答, 可供初学者练习。

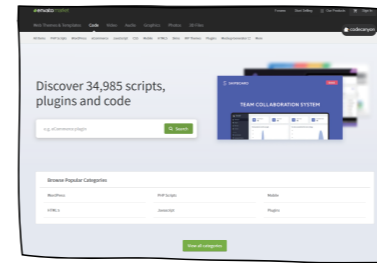


网站: Bug之神——Stack Overflow

网址: stackoverflow.com

人群: 初、高级程序员

介绍: Stack Overflow 可能是拉开初级跟中高级程序员水平的一个网站, 实际上, 它更像是编程的“十万个为什么”, 无论是简单还是复杂的Bug, 在这儿基本都会得到解答。这里还是一个很好的交流平台, 对于有经验的开发者, 还可以来这儿帮助别人解决问题, 增加个人影响力。

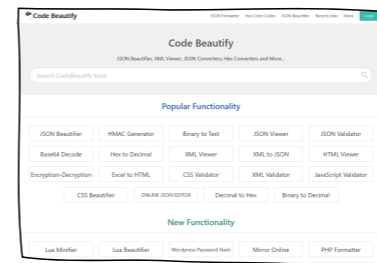


网站: CodeCanyon

网址: codecanyon.net

人群: 想卖代码的程序员

介绍: 代码也可以放在网上卖的! 代码以PHP、WordPress主题、Java、CSS为主, 偏前端。给大家来个鲜活的例子: 这个网站上的一个19刀的代码卖了3万多份。程序员们, 恭喜你们走上了代码致富之路!

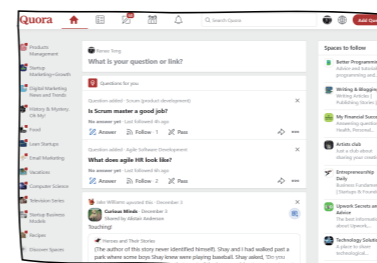


网站: Code Beautify

网址: codebeautify.org

人群: 想让代码变美的程序员

介绍: 你的代码, 曾让别人觉得难以阅读吗? 可以用这款代码界的“PhotoShop”来解救。Code Beautify 有很多可以美化的语言, 可以让代码更容易阅读。另外, 如果代码不想被其他人读取, 这个网站同样也可以做到。



网站: Quora

网址: www.quora.com

人群: 想逛国外知乎的程序员

介绍: Quora 是一个问答社交网站, 可以理解为国外版的知乎。Quora 上的答案排序功能, 会让想学习编程的小伙伴得到自己想要的高质量答案!

### 4.2.2 国内技术社区

网站: 博客园

网址: cnblogs.com

介绍: 博客园可谓国内众多同类社区中, 最为纯净的技术交流社区了。博客园推动并帮助开发者分享技术干货, 从而让更多开发者从中受益。博客园的使命是帮助开发者用代码改变世界。



网站: SegmentFault

网址: segmentfault.com

介绍: SegmentFault 是中国新一代的开发者社区和专业的技术媒体。有问答、博客、活动等专区, 还组织过多个黑客马拉松活动, 为中文开发者提供了纯粹、高质的技术交流平台以及最前沿的技术行业动态。



网站: 测试窝

网址: testwo.com

介绍: 测试窝是一个以软件测试为主题的社区门户。平台提供原创技术写作、海外测试译文、测试招聘发布等内容, 还会发布最新海内外测试资讯、线下沙龙活动等。



网站: 掘金

网址: juejin.cn

介绍: 掘金是一个帮助开发者成长的社区。掘金的技术文章涵盖 Android、iOS、前端、后端、人工智能等方面的内容。用户每天都可以在这里找到技术世界的头条内容。与此同时, 掘金内还有沸点、掘金翻译计划、线下活动、专栏文章等内容。



网站: 敏捷开发社区

网址: minjiekafa.com

介绍: 敏捷开发中文社区提供敏捷开发相关的知识、视频、培训等相关内容。为大家提供敏捷开发方法学习和更好的敏捷开发软件, 以帮助大家更深入和全面地了解敏捷开发。



网站: V2EX

网址: v2ex.com

介绍: V2EX是个汇集各类奇妙好玩的话题和流行动向的网站。共分灰色荒野、混沌海、法印城、机械境、极乐境五个区域, 其中的机械境, 从 Java 到 Rebol, 从 MySQL 到 Sybase, 从 Lucene 到 Geronimo, 在讨论中, 你一定会有所收获。



### 4.2.3 云计算平台

根据中国信通院发布的数据,2016-2019年,中国公共云市场规模逐年上升,增速保持在55%以上。2019年,公共云市场规模达到689.3亿元,同比增长57.6%。

随着云计算的飞速发展,无论是存储、数据库、软件还是网络,越来越多的企业开始使用基于云的企业服务。同时,国内不断地涌现出优秀的云计算服务商,助推企业实现虚拟化、数字化转型。

网站: 青云QingCloud

网址: qingcloud.com

介绍: 青云 QingCloud 是一家技术领先的企业级云服务商与数字化转型解决方案提供商,致力于通过一套自主创新、中立可靠、灵活开放的全维云平台为数字世界的高效运行提供坚实的基础支撑,加速推动百行千业数字化转型。



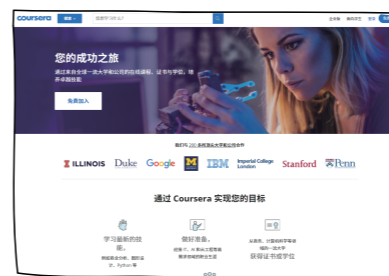
网站: 好雨Rainstore云原生应用市场

网址: goodrain.com

介绍: 好雨科技专注于“连接企业、让云落地”,通过云帮「Rainbond云原生应用管理平台」和云市「Rainstore云原生应用交易市场」为企业IT能力升级、云原生应用交付、搭建行业生态提供服务 and 实施工具。



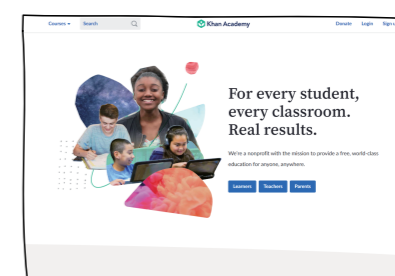
### 4.2.4 公开课网站推荐



网站: Coursera

网址: coursera.org

介绍: Coursera 与多所世界顶尖大学合作,在线提供网络公开课程。Coursera 的首批合作院校包括斯坦福大学、密歇根大学、普林斯顿大学、宾夕法尼亚大学等美国名校。



网站: 可汗学院

网址: khanacademy.org

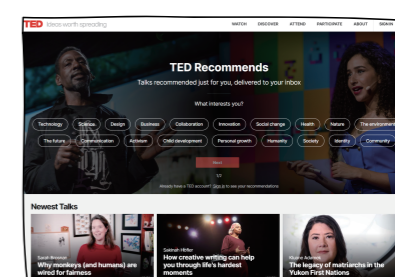
介绍: 可汗学院是非营利性组织,使命是为任何地方的任何人提供免费的世界一流的教育。网站汇聚了世界上很多名师,课程范围囊括数学、科学、计算机编程、艺术史、经济学等。



网站: 中国大学MOOC网

网址: icourse163.org

介绍: 中国版 Coursera ,基本都是985、211高校课程,全部免费。



网站: TED

网址: ted.com

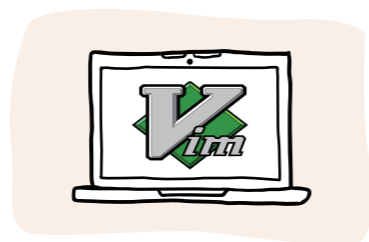
介绍: TED 资源多多,涵盖了100多种语言,从科学到商业再到全球的很多内容。满足程序员的多语言学习,适合准备走向世界的程序员。

## 4.3 工具推荐

前面推荐了诸多可帮助开发人员成长的技术学习交流网站,接下来我们继续推荐程序员工作、学习中提升效率的必备工具。

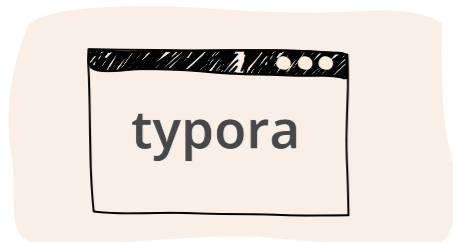
### Vim

Vim 简洁而强大,可谓编辑神器。这里仅是推荐,我们不必纠结哪个 IDE 是最好的。选择工具,适合自己的才是最好的,不为酷炫,而是为提升我们的效率。



### Typora

对于习惯使用 Markdown 写笔记、文档的技术人员来说,开源、免费的 Typora 可谓是一个靠谱、用户体验很舒服的编辑器。



### Everything

Everything 是一款免费、小巧、速度极快的文件搜索工具,它能够基于文件名快速定文件和文件夹位置。Everything 可以帮你大幅提升工作效率。



## 第五部分:管理篇



除了以上那些能量化的内容,还有许多虽然看不见、摸不着,但在潜移默化中影响着工作乃至人生轨迹的“技能”。这些“技能”在学校里老师没教过,可都是工作和生活里不得不面对的。活用“学校已教过的事”,并随时用心学习“学校没有教的事”,已是现代人想要突破挑战、学习致胜的不二法门。

## 5.1 健康体魄

### 5.1.1 996也要睡得好!

睡眠周期:完整的浅睡眠+深睡眠周期为1.5小时,睡够完整周期倍数才能得到深入休息,醒来也不会疲乏、困倦,一般来说可以是7.5小时,自由安排睡眠时间,每天根据睡眠时间定闹钟。

入睡时间		22:00	22:30	23:00	23:30	00:00
起床时间	中规中矩 睡7.5小时	05:30	06:00	06:30	07:00	07:30
	天赋异禀 睡6小时	04:00	04:30	05:00	05:30	06:00

日间小睡:短暂午睡不超过30分钟,时间充足的话可以睡一个周期(1.5小时)。

具体时间:睡眠时长、入睡时间、起床时间因人而异,只要睡够完整睡眠周期,不必过于纠结早睡早起。

其他 tips:胎儿型睡姿、暖光灯、热牛奶、泡脚等都有助于睡眠,可根据自己情况搭配使用。

### 5.1.2 吃好喝好烦恼少!

饮食建议方面,阿道的建议远没有官方来得科学,就直接把膳食宝塔给到你。



一定会有人开始焦虑,但悄悄告诉你,阿道也不能完全按照这个比例来吃东西!放松心情~只要你不是每天五顿饭,顿顿炸鸡啤酒小烧烤,在三餐规律、有奶蛋鱼肉菜搭配的情况下,倒也不必过于担心健康。来张BMI指数(身体质量指数)来算一下自己的体重是否符合健康程度吧。

$$BMI = \frac{\text{体重 (KG)}}{\text{身高}^2 (m)}$$

适用	BMI 指数	身体状况
18岁及以上成年人	BMI < 18.5	体重过低
	18.5 ≤ BMI < 24	体重正常
	24 ≤ BMI < 28	超重
	BMI ≥ 28	肥胖



同时也记得定期体检,除了拖延症、脱发、选择困难症等可自测的“病症”,还得注意一下当代年轻人常见的腰椎病、颈椎病,甚至日渐年轻化的糖尿病、结节等问题哦。

### 5.1.3 体育锻炼

运动的重要性不言而喻,但种种主客观原因导致没有充足的锻炼时间,那么就总结为一句话,适用于工作场合:勿久坐,多走动,有条件的可以站立办公。再来学点轻松易做的锻炼动作:



小美人鱼伸展

双手交叉翻转举过头顶,左侧身体向右侧弯曲延伸,停留10秒,再换另一侧。



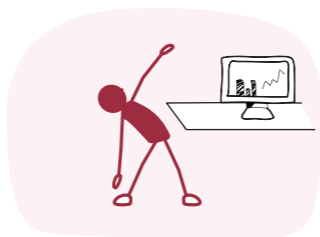
双肘伸展

双手呈90度反握在桌边,臀部如同坐下的姿势,来回上下蹲起。



椅上深蹲

臀部坐于椅子的前端，双脚分开与肩同宽，双手伸直并抬至于肩部平行。呼气，臀腿发力，起身站直。此时臀部夹紧，腹部收紧。重复动作。

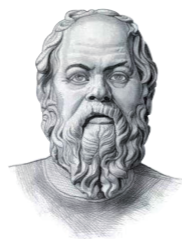


侧腹拉伸扭转

双腿分开站立，挺直腰背，提下巴，双臂完全打开。吸气，上肢前屈的同时，右手触碰右脚尖，左手和脸部朝上。左臂往上拉伸，左侧腰有拉伸的感觉。反向重复进行。

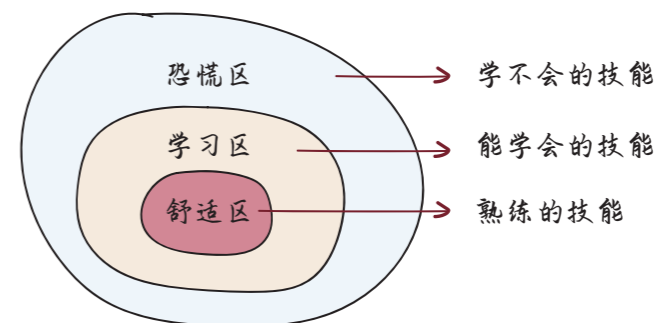
## 5.2 自我认知

“Know Yourself”是苏格拉底的哲学思想之一，不探讨高深的哲学，生活工作里也能应用到这一先贤的思想。



清楚赖以生存的工作能力的上下限，比如代码水平。上下限之间就是你的能力圈，在这个范围内可以尽可能地拓宽边界、进行提升。清楚了上下限就不至于在面对机会时过度自信搞砸事情，或犹豫不决错失良机。上下限跨度不要过大，过大会导致工作能力的不稳定，将能力范围稳定在中高度范围。

清楚自己的优劣势。上下限更多的是如代码能力等硬技能，优劣势就包含了交际、沟通、协作等软技能。保持优势一直是稳定优势的情况下，规避劣势并有意识地学习、提升，将从不同方面带来很大程度的好处。



但切忌出现放弃自己专长而去发展短板的情况，从这个意义上讲，超出能力圈做力所不能及的事而搞砸比陷入舒适区更可怕。

## 5.3 主动学习

现在几乎所有的“给大学生的建议”、“给职场人的建议”、“给XXX的建议”等人生建议都少不了主动学习这一条，好像下班后不再上个补习班就对不起自己的人生。



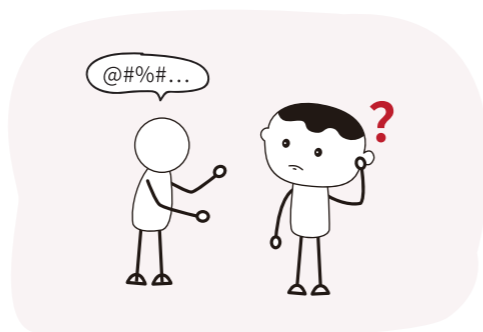
Take it easy,“主动学习”没有那么夸张，我们几乎无时无刻不在“学习”：从同事那种草了一款新软件、在微信文章中了解到行业趋势、看禅道的发音视频纠正了一次错误读音、看《程序员生存手册》学会了小美人鱼伸展……这些，都是一种学习。在这些小知识点上停留一下下，加以消化、吸收，再举一反三发散一下，这就是一次主动学习的过程。

主动学习与上述的认识自己一脉相承，扬长避短，积水成渊。而程序员本就处于飞速发展的行业，确实需要比其他行业更能把握时代的脉搏。

但当下生活诱惑颇多,手机、iPad、Switch、形形色色的 APP 都在榨取着你的时间,如何抵制这种快速反馈的廉价快乐,学会深度思考和学习?不畏浮云遮望眼,装备好一颗进取和自律的大心脏,以终为始、结果导向,主动学习,一点也不难。

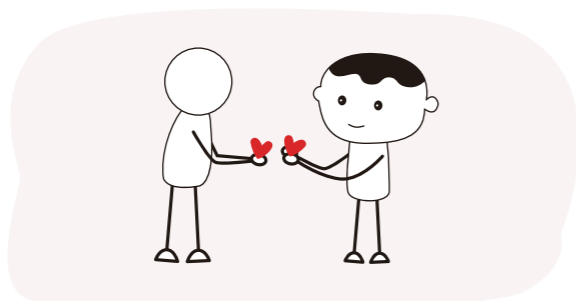
## 5.4 沟通能力

有效沟通包含两部分:倾听他人、表达自己。达成这两部分不仅是口头语言的交流,还有肢体、神态等辅助语言,以及,更重要的是毫无偏见、互相尊重、友善沟通的心态。



### 5.4.1 知彼解己

好多外行都吐槽互联网人“不说人话”,连相亲都用“水位同步”“对齐需求”“生态化反”这些黑话,那阿道可得解释一下:每个圈子都会采用“组块化策略”来讲话,以节约时间、提高沟通效率。这种“组块化策略”就像压缩包,具有解压能力的人可以轻松读懂,但没有掌握这些压缩包的圈外人却只能接收不能解压。于是就出现了“每个字我都认识可就是不知道什么意思”。



知彼解己是有效沟通的原则,首先了解对方,然后再争取让对方了解自己。所以,沟通交流之前,可以换位思考,简单判断一下对方是否会了解你的“压缩包”,如果判断对方可能不是那么容易解压,请务必采取“说人话策略”,即用深入浅出、用浅显的语言呈现复杂内容。

### 5.4.2 提升交流意愿

很多程序员更喜欢自己与代码待着,内向不是问题,但现在社会分工精细,人与人协作频繁,保持封闭拒绝交流确实无益于各方面的提升。很多价值要通过业务上的沟通交流才得以量化。所以,打开自己。



比如用直视、点头等动作来认真倾听、回应;看着对方眉心到鼻梁附近的“三角区”来进行眼神注视;开会、电话等需发言场合提前理清逻辑、分列要点。

## 5.5 领导能力

领导力不是当了领导才需要的权力,而是一种前瞻性的统筹能力。领导力与职位无关,只意味着看问题的眼光能否从宏观和全局出发,以整体、均衡、多面的思路应对工作、生活乃至世界。

管理者的本质是公司相信你能为公司解决更大的问题,从而愿意分配一部分人力资源以帮助你更高效地解决问题。既要你深入一线工作,又要有一定的带队能力,让团队发挥更大价值。

有无领导力的表现,对个人来说在于能否敢于授权与勇于担责,对团队来说在于有没有结果和有没有状态。有领导力的人懂得授权激励,让专业的人做专业的事;没领导力的人事无巨细、亲力亲为。有领导力的人能调动团队积极性,引导每个人发挥潜在优势,并达成一致的团队目标;没领导力的人徒有严格的等级关系,团队毫无凝聚力,更别提共同完成任务。

要想简单高效地打造领导力,只需要重点把握四个关键词:

目标: 为团队确定简单明确的目标  
沟通: 和员工进行直接明了的沟通  
授权: 对团队成员要进行充分授权  
反馈: 对行动结果要做到正确反馈



## 5.6 决策能力

复杂的决策之所以难,在于很多决策并没有统一的标准、无法被量化,甚至还有可能需要承担责任或后果。那么决策能力就是将所有标准统一,并将其量化,根据可能带来的结果做出最终选择的能力。

决策三步走:第一步,列出标准;第二步,分配权重;第三步,量化选项。将所有的分数,根据真实权重进行加权汇总,得到一个最终评分。

示例: 阿道的一位女性同事小蝉最近正为择偶头疼,李大帅和张有才,她有些纠结。



### 1 列出标准。

这两个人需要对比外貌、性格、收入、工作前景、家庭等因素。

### 2 分配权重。

小蝉是颜控,外貌权重得占3;  
小蝉自身经济独立,也不拜金,现有收入比重较小,权重为1;  
其他因素性格、工作前景、家庭权重都为2。

### 3 量化选项。

	李大帅	张有才
外貌	+3	0
性格	0	+2
收入	+0.5	+0.5
工作前景	0	+2
家庭	+0.5	+1.5
总分	4	6

结果。

就总分来看,结果很明显,量化结果是选张有才。

经过阿道的决策建议,小蝉已经委婉拒绝了李大帅,开开心心地和张有才去约会啦!



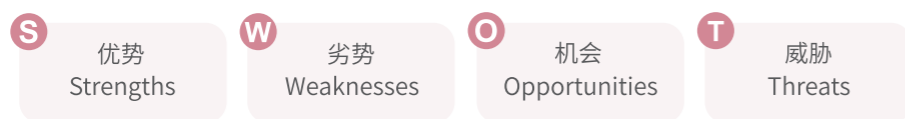
从决策的三个步骤出发给大家几个决策模型,在列出标准、分配权重时可以有所帮助:

#### MECE分类规则

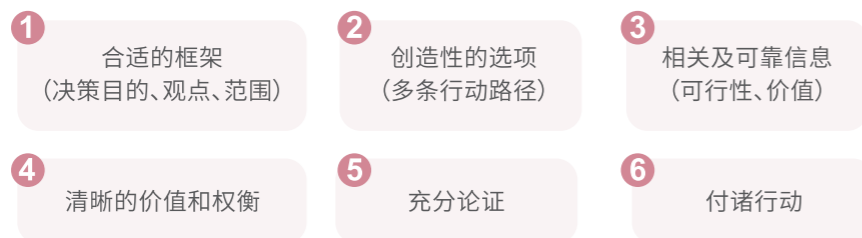
即完全穷举,相互独立,可以用MECE原则考虑“列出标准”时的每一条的合理性,以使各标准不存在遗漏也不会互相干扰。

## SWOT分析

S (Strengths)是优势、W (Weaknesses)是劣势、O (Opportunities)是机会、T (Threats)是威胁,将与研究对象密切相关的各种主要内部优势、劣势和外部的机会、威胁等列举出来,并以此判断每个因素在权重中所占的比例。



## 斯坦福六步法商业决策



自我管理方面,除了以上职场必备技能,还有一些结合当下现实情况的可能落地性没那么强、但确实是“掏心窝子的话”想要跟大家分享。

时代发展飞速,科技日新月异,带给我们极多便利的同时也颠覆着传统生活方式。算法的便利让大数据推送完美契合自己的喜好,可我们也日复一日被困在同质化的“信息茧房”中,只能看见井口的那一片天空;日渐被资本家鼓吹的消费主义裹挟,初中课本学的“量入为出适度消费、避免盲从理性消费”已被抛之脑后,分期付款、提前消费成了生活常态;段子、漫画、短视频无处不在,让人捧腹的同时侵蚀着深度阅读和理性思考能力。没人能做到“众人皆醉我独醒”,但在纷杂喧嚣的大环境里留一份理智,不随波逐流并不难。

越是在一个稍微动乱点的时代,静才越有价值。

## 5.7 时间管理

无论是硬核知识技能还是软技能,最终都是要我们自己学习掌握,并落实到实践。而在这个过程中,我们都会遇到一个共同难题——时间管理。

时间管理是一个大课题,阿道主要分享介绍一些时间管理的具体小技巧。

### 5.7.1 每日待办列表

整理每日待办列表是一种管理安排我们每天时间的有效方法。它可以让我们有限的时间更加条理化,在合适的时间做合适的事情,有目的地完成目标。列待办事项看似简单,但里面有很多小技巧 and 注意事项哦,这里分享给大家参考。



### 5.7.2 告别拖延

虽然兴致勃勃地给自己列了一堆每日待办,但你是否有过这样的场景:明明自己还有很多待办事项没做完,却忍不住地要先看看短视频、逛逛朋友圈、刷刷微博……最后无奈地告诉自己,这些事情等明天一定要完成。造成这种现象的罪魁祸首就是我们的通病——拖延症。

关于拖延,不管你已经喝过多少碗文字鸡汤了,问题依旧在那里。因为你需要的并不是说教和念叨,也不是鼓励和提醒,而是切实的方法和行动。下面我们一起来看看造成我们拖延的原因有哪些以及有什么解决方案。

如何告别拖延症?



拖延原因	解决方案
制定的目标、任务太大、太艰巨。	<ol style="list-style-type: none"> <li>1.目标虽重要,但不要贪多哦。一段时间内,集中精力完成一个目标,避免目标捆绑。</li> <li>2.目标与过程足够清晰具体,可将其分解成多个关键结果。不要只盯着总目标,确保每一期迭代都有一个主要目标,实现持续交付。</li> </ol>
总是看轻未来的收益,而妥协于眼前的诱惑。	<ol style="list-style-type: none"> <li>1.加重对未来收益的感知,比如给自己设定一个完成目标时的奖励。</li> <li>2.环境中设置触发点,排除身边可触及的干扰因素(如电脑里的游戏、手机里的短视频 APP 等等)。</li> </ol>
过度完美主义,将热情消耗在准备上,缺乏执行力。	<ol style="list-style-type: none"> <li>1.降低事情开始的难度,并缩短准备的时间。</li> <li>2.不必设定过于完美的计划。执行过程可以通过敏捷的思想,不断地迭代完善,逐步缩短与目标的距离。</li> </ol>



受篇幅和内容所限,阿道就此搁笔。这个手册的内容是自己学习、生活、经历的经验总结,当然这只是一家之言。出刊目的是希望这本手册能为你的职场“生存”带来一些助力,或者在“生存”之外的启发和思考。所以不必拘泥于其中的某个方法、某条建议,灵活运用、提升自己就是最大的收获。

手册之内,是脚踏实地、致知力行,合上手册,更多的星辰大海、诗和远方等你去探索!

#### 主创团队的话:

《程序员生存手册》是我们对这种形式的第一次探索和尝试,很期待你来分享收到这本手册、打开这本手册、直到阅读到这里的心情和感受。不管何种心态,我们期待着你任何形式的反馈,可以是文字、图片甚至是视频、Vlog、漫画……无论感谢还是吐槽,我们都一并接受。因为禅道不仅想带给程序员专业的工具,还想带大家更多的进步和思考。秉持着开源共创的心态,我们在 GitHub 上创建了《程序员生存手册》这个公共项目库,欢迎有想法、有经验的大家一起来进行我们后续的迭代,让手册不断完善、优化。

之后这一系列还可能延续,比如《部门经理生存手册》《创业者生存手册》等等,也想听听你们的期待。所以,通过以下任何形式来找我们吧:



GitHub  
期待与您携手共创



阿道  
快来和阿道讨论《程序员生存手册》

## 致谢

这本手册到这里就正式结束了，从有想法到交付印刷历时10周、多次迭代，期间得到许多帮助，在此一一感谢。

感谢对手册进行内容审校的老师，他们不仅在前期对敏捷、CMMI等专业内容进行深入浅出的讲解，还认真对初稿内容进行审校、点评，保证了内容的专业性。

内容审校：



感谢各社区在手册编写过程中提供的内容、技术、资源合作等方面的支持：



(按首字母顺序排序)

感谢禅道用户，也就是收到这本手册的你们。朝暮与年岁并往，然后一同行至天光，禅道的旅程由我们一同走过。你们的支持给着我们前行的动力，你们的互动和反馈让生存手册这颗种子落地生根。恕无法一一展示你们的姓名，但我们拥有共同的身份：禅道人。期待一起走更长更远的路，去看更亮的光。

以上，依然难免遗漏，诚惶诚恐，再次感谢大家。