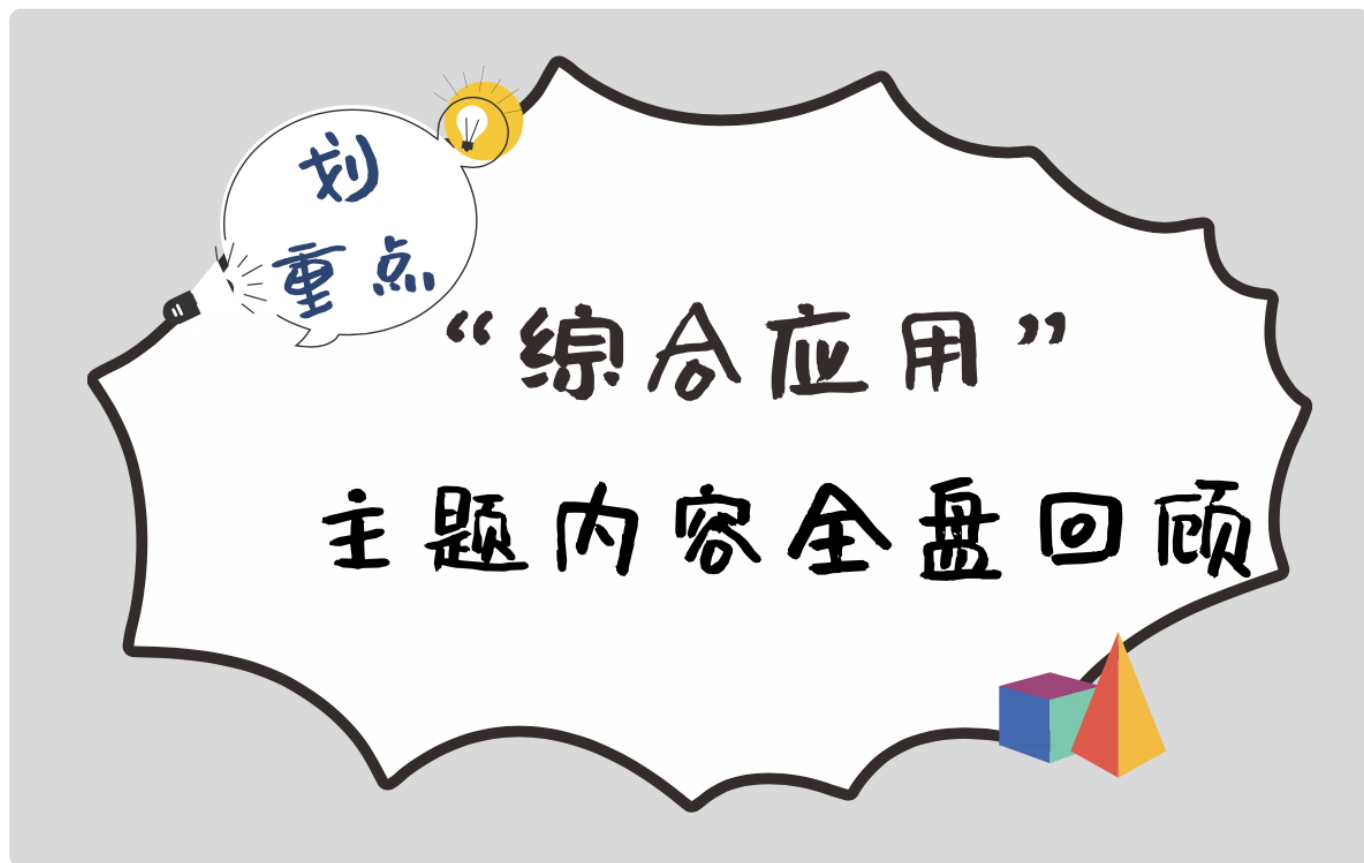


## 划重点 | “综合运用”主题内容的全盘回顾

2019-04-22 郑晔

10x程序员工作法

[进入课程 >](#)



讲述：郑晔

时长 00:41 大小 652.00K



又到了我们划重点的时间了，因为篇幅关系，“综合运用”这个模块最为短小精悍。

在这个模块中，我们把前面学到的各种知识综合起来，运用在实际的工作场景中，让你知道这些内容并不是一个个孤立的实践，在实际工作中，唯有将它们结合起来，才能发挥最大功效。

### 重点复习

在这个模块中，我们学习到了一些新知识。

#### “学习区”学习模型

舒适区，舒适而缺乏成长。

恐慌区，超出能力范围。

学习区，有难度而可以达成。

在学习区练习才能得到足够的成长。

## **T 型人才，一专多能**

知识的广度。

专业技能的深度。

有“一专”，“多能”才是有意义的。

在这个模块中，我们还了解了一些重要的思路，让我们把工作做得更好。

## **进入新工作，从全面了解了解开始**

业务：做什么。

技术：怎么做。

团队运作：怎么与人协作。

从大到小，由外及内地了解工作。

## **面对遗留系统，稳扎稳打，小步前行**

### **基础理念**

烂代码只是现象，要了解根因。

能重构，先重构，大规模改造是迫不得已的选择。

小步前行。

### **实际操作**

构建测试防护网。

将大系统分解成小模块，逐步替换。

新旧模块并存，由分发模块调度。

建立好领域模型。

寻找行业对于系统构建的最新理解。

## 程序员的职业发展

程序员的焦虑来自于对未来的不确定性，这种不确定性是一个特定时代加上特定行业的产物。

快速发展的中国经济。

程序员在中国是一个新兴职业。

成为行业专家，制定高目标。

向大师学习，开拓视野。

找到好的问题，和高水平的人一起工作。

## 实战指南

了解一个项目，从大图景开始。

——《[38 | 新入职一家公司，怎么快速进入工作状态？](#)》

小步改造遗留系统，不要回到老路上。

——《[39 | 面对遗留系统，你应该这样做](#)》

在学习区工作和成长。

——《[40 | 我们应该如何保持竞争力？](#)》

## 额外收获

在这个模块的最后，针对大家在学习过程中的一些问题，我也进行了回答，帮你梳理出一个思路，更好地理解学到的内容：

推行新观念，找愿意改变的人，做具体的事。

Lead by Example.

外部系统应该用接口隔离，这种做法体现了接口隔离原则（ISP），也是防腐层概念的体现。

外部系统的测试，能模拟的就模拟，能本地的就本地。

## 留言精选

关于入职一家新公司，怎么快速进入工作状态这个问题，[西西弗与卡夫卡](#) 同学分享了他的方法：

有朋友正在转型，从乙方商业化产品的交付经理转向新公司的产品经理。原本得心应手的思维方式和习惯，遇到了巨大挑战。以前只需依据已有产品的功能出解决方案，能做就能做，不能实现就是不能实现，到某个时间交付什么功能很明确，考核是以交付签字为准。现在需要面对各方需求，自己想明白用户真正的问题是什么，最终要交付的价值是什么，没有一个实体的谁人来签字，只有不断地迭代。

借鉴领域驱动设计，可以采用以下方法。简单描述的话，是一个点、一个圈再加一个箭头线，是不是有点像丘比特？

一个“点”，指的是用户核心价值。这是最关键的一条，基本上只能靠自己想明白。想，不是闭门造车式的苦思冥想，可以是已有的领域经验，可以从书本中学习，可以是大家的各种吐槽，可以是自己从旁边观察用户的实践，还可以是自己变身为用户的实践。

有些人会纠结“点”想的对不对，迟迟不敢动手。其实一开始想得对不对不是那么重要，关键是要有这“点”，然后快速到市场上验证，根据反馈再调整。

一个“圈”，指的是围绕核心价值划出的范围，即领域驱动设计中的限界上下文。产品经理面临的一个现实是，各种人都会给你提需求，只要他们觉得和你有关，还时不时来问什么时候可以实现。

需求轰炸之下很容易焦虑，不光自己焦虑，所有的利益相关者都会焦虑。依据核心价值，框出需求范围，在和各方交流过程中可以有一种确定性，减少焦虑，利于行动。

大家（不光是研发团队，也包括其他需求方）就能明白，哪些和当前核心价值密切相关，我们优先考虑；哪些与核心价值有关但它不在我们的范围内，属于其他团队，需要他们协助；哪些有关系，但目前没想清楚价值大不大，并且代价可能很高建议先搁置。范围不是一成不变，它随着时间会发生变动，所以我们不要追求固定，只要保证在某个时间段内，大家一致认同即可。

一个“箭头”，指的是实现路径，箭头指向核心目标（核心价值）。目标（核心价值）和范围描绘的是终极，而从现实到终极还有很多路要走，可能的路径还有很多条。我们需要琢磨怎么走更稳当，怎么走代价比较低，路上关键的里程碑是什么。路径对不对是其次，重要的是思考过程，可以把关键点需要交付的价值、需要支持的资源等等梳理清楚。

另外，**西西弗与卡夫卡** 同学还对于程序员如何保持竞争力的问题给出了非常不错的建议。

补充我的一些做法。工作中不要满足当前需求，要经常从自己上级主管甚至老板角度来审视自己的工作，思考业务的终极目标，持续琢磨扩展边界，挑战工作难度。

平时多看书多思考，除了钻研某个领域，还要多有涉猎，拓展领域，成为终身学习者。

适当运动维持健康，你有更多体力和更强抗压能力的时候，就可以超过不少人。

保持竞争力除了上述之外，要保持乐观，相信大多数事都有解决方法，在多数人都容易放弃的时候，你的坚持，就是竞争力。

对于新入职一家公司的场景，**Y024** 同学分享了他快速进入工作状态的方法：

1. 我会在权限允许的范围内，时不时的到处翻翻 ftp、内部 wiki 等资源，星星点点构建全貌（业务、技术、团队）。

2. 梳理系统数据流。去年很火的电视剧「大江大河」里，宋运辉初入职场的的方式就很值得借鉴：先走通全部流程，有个全貌，利用图书馆、师傅等资源再自己动手各个击破并绘制流程图，最终实践检验认知，以技术说话融入团队。

（他就每天只要天气晴朗，绕着设备上上下下、里里外外地跑。一个星期下来，全部流程走通；两个星期不到，原理搞通，仪表能读，普通故障能应付；第三星期开始，他可以开出维修单，但得给师父过目；第四星期起，谁

有事请假他可以顶上，坐到仪表盘前抄表看动态做操作。师父说他学得很快。

第四星期起，没人可以让他顶替时候，他在仪表室后面支起绘图板。先画出工艺流程图，经现场核对无误，又让师父审核后，开始按部就班地根据液体走向，测绘所有设备的零件图、装配图、管段图等。

这工作最先做的时候异常艰难，首先是绘图不熟练，很多小毛病，尤其是遇到非标零件，还得到机修工段测绘，有时一天都绘不成一个小小非标件。如果车间技术档案室有图纸还好，可以对照着翻画，可档案室里的图纸残缺不全，前后混乱，想找资料，先得整理资料。

资料室中年女管理员乐得有个懂事的孩子来帮她整理，索性暗暗配把钥匙给宋运辉，要是她下班不在的时候，让宋运辉自己偷偷进来关上门寻找资料。

机修工段的人本来挺烦这个宋运辉，说他一来维修单子多得像雪片，支得他们团团转，有人还趁宋运辉上班时候冲进控制室指桑骂槐，被寻建祥骂了回去，差点还打起来。但后来集中一段维修高峰后，维修单子又少了下去，上面还表扬跑冒滴漏少很多，一工段和机修工段各加一次月奖，可见设备性能好转。

再以后遇到维修，他们不能确定要用什么零件，打个内线电话给控制室问宋运辉，一问就清楚。双方关系渐渐变得铁起来。基层有时候很简单，只要拿得出技术，别人就服。 )

另外，Y024 同学还很认真地整理了专栏提到的部分图书：

郑老师拍案惊奇书单及简评，最近各大书店有活动，可以借机囤起来了。

## 1. 重构

作者: Martin Fowler

<https://book.douban.com/subject/1229923/>

严格说来，我并没有完整的读完这本书，不过，正如作者自己所说，这样的

书原本就不指望能够读完，因为有一大部分其实是参考手册。正是我读过的部分让我知道了重构，让我知道这么做可以把代码写得更好。

## 2. 敏捷软件开发

作者: Robert C·Martin

<https://book.douban.com/subject/1140457/>

这是一本名字赶潮流，内容很丰富的书，这本书让我开始理解软件设计，从此不再刻意追求设计模式。

## 3. 测试驱动开发

作者: Kent Beck

<https://book.douban.com/subject/1230036/>

读的是英文版，因为当时中文版还没有出版，所以，我不敢说，我通过这本书很好的理解了测试驱动开发，但它却为我打开了一扇门，让我知道了一种更好的工作方式。

## 4. 修改代码的艺术

作者: Michael Feathers

<https://book.douban.com/subject/2248759/>

这是一本讲解如何编写测试的书。至于这本书的具体内容，我的评价是实用。如果说不足，那么，这本书缺乏一个列表，就像 Martin Fowler 为《重构》所做的那样，出什么样的问题，应该采用怎样的手法进行处理。

对于如何面对遗留系统，**毅** 同学提到：

1. 了解原系统已实现的功能，没有文档就在心中划分好内部功能模块；
2. 各模块的边界及关联，对于业务交叉点先思考通信机制；
3. 看代码，通常是瓶颈优先，业务上是先复杂后简单；
4. 选定切入点；
5. 正式改造时先把原有功能抽象出来使用现有实现，改造的过程完成前不会受影响；
6. 改造完成后切换但新实现进行测试；
7. 稳定后替换旧实现；
8. 重复 4-7。

Wei 同学对于“T 型人”的说法感触很深：

“T 型人”这个太说到点了，到底是做“专”还是做“广”，哪条路线一直是我思考的方向；工作上跟大牛工作过，给我感觉几乎是全能的，我一直都想像他们那样，做一个多面手，但是如何做广，这一直是困扰我的一个问题。

我是 dev 出身，但是现实遇到的问题往往跟数据库，发布的平台相关；这样说下来，各种相关领域，数据库、k8s、网络协议、DNS，都需要大量时间去积累；有时候什么都懂一点，反而让自己应该定位什么角色感到迷茫了，掌握的水平不足以让自己去应聘 DBA、Ops，但是只是应聘 dev 似乎又有点“浪费”，跟那些熟悉最新语言 / 框架的对比起来没特出竞争力。

今天学习“T 型人”这个概念，让我好好思考了自己到底应该怎么定位。我首先是一个 developer，这个是根；对语言特性的熟练掌握，各种 best practices，例如课程中提到的 TDD 等应该熟练应用起来；然后在这上面拓展，学习架构知识，多思考对不同系统应该怎么设计，老师提到的 DDD 会认真学习应用；再有软件最终还是给用户使用，而不是单单提交代码。相关的数据库、k8s、监控运用根据实际遇到的问题再学习解决。

最重要的是，在学习区终身学习和工作！

对于如何持续保持竞争力的问题，enjoylearning 同学提到：

程序员如何保持竞争力很重要，在这个年轻人学习能力不断提升的 IT 行业，作为老程序员经验阅历眼光以及技术前沿判断力就显得越来越重要。

说起来这个职业是一个需要终身学习的职业，年龄不重要，能力才重要，是不是让自己永远呆在学习区更重要。

对于技术推广，desmond 同学的理解也很棒：



技术推广，不要先推广最难的部分，先推广能让对方感到最明显好处的部分。取得对方的信任，是友好沟通的基础。

感谢同学们的精彩留言。我们的专栏更新已经进入尾声阶段，后续我会为大家做一些对整个专栏进行全盘复习的内容，敬请期待。

感谢阅读，如果你觉得这篇文章对你有帮助的话，也欢迎把它分享给你的朋友。



# 10x 程序员工作法

掌握主动权，忙到点子上

郑晔

火币网首席架构师  
前 ThoughtWorks 首席咨询师  
TGO 鲲鹏会会员



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 [答疑解惑 | 如何在实际工作中推行新观念？](#)

下一篇 [总复习 | 重新审视“最佳实践”](#)

## 精选留言 (5)

写留言



enjoylear...  
2019-04-22

3

很好的总结，程序员成长之路，其实还想问问程序员要不要面向薪资编程，要不要转语言

转技术方向，一门心思扎下去好不好，要不要转型到机器学习或大数据比较热门有潜力的方向。

展开 ▾

作者回复: 薪资、语言，甚至热门方向都是偶然复杂度，你的专家技能才是本质复杂度。



**Y024**

2019-04-23

👍 1

给意犹未尽的童鞋加点料：

1.郑老师以前博客，因为博客巴士被关闭了，不幸中的万幸是还可以通过这个备份链接访问（不过难度 qiang 了点）

<https://web.archive.org/web/20150924210902/http://dreamhead.blogbus.com/>

...

展开 ▾

作者回复: 哈哈，都被你找出来了。



**hua168**

2019-04-22

👍 1

这么快就结束了🙄，舍不得...

老师有微博吗？

关注新技术是看infoQ和雷达就行了？

如果深入某些技术的话，是否看官网比较好？

作者回复: 看以后是不是还有机会再来分享。



**WL**

2019-04-25

👍

收货很多感谢老师

展开 ▾





Jxin  
2019-04-23



西西弗与卡夫卡大佬，对整个专栏做了很好的补偿拓展。特此感谢，除了从老师这，也从大佬的回复中受益良多。

作者回复: 确实，西西弗与卡夫卡 的留言一直让我印象颇深。

