总复习讲重新来“看书”

我们继续复习，在上一讲中，我从最佳实践的角度带领大家回顾了专栏里的一些内容。这一讲，我们换个复习的角度。在专栏进行的过程中，有一些同学注意到我引用了大量的书籍，提出让我把这些书做一个整理。

Wei 同学提到：

> 有一个小建议： 在每一个主题模块的小结中，把文章中提到的书籍做一个书单方便读者。

刘晓林 同学提到：

> 郑老师在专栏中推荐了很多非常好的书籍作为参考，可否考虑在某一期中，将这些参考书籍整理成一个书单，按照专栏的主题做个小分类，然后每本书简单点评两句作为领读内容。希望在专栏的结束语之前可以看到这个书单。

Y024 同学甚至在留言中帮我总结了一个小清单，而有人也在豆瓣上做出了一个豆列，罗列了专栏中提到的一些书。

在今天这一讲中，我就站在“看书”的视角，带着你进行一次复习。这些书大多是在我个人成长过程中，给我留下深刻印象的。

我希望你在结束这个专栏学习之后，开启的是另外一段学习历程，用这些书提升自己的水平，夯实自己的基础知识。学习了这个专栏之后，你拥有了一个新的知识结构，再来看这些书就会有一种全新的体验。

此外，在这次的内容中，我会提到几本专栏中没有提到的书，算是给你在学习路上的一个补充。我还制作了一个豆列，方便你去找到这些书。

编码实践

- 如果你想详细学习如何写好代码，我推荐你去读 Robert Martin 的《代码整洁之道》（Clean Code），这本书几乎覆盖了如何把代码写好的方方面面。
- 《实现模式》是一本关于如何写好代码的书，更具体一点是，编写别人能够理解的代码。它的作者 Kent Beck 是许多软件开发实践的开创者。但 Kent Beck 的写作能力一般，他的很多作品被埋没了。只有细细品味，才能体会到 Kent Beck 深厚

的功力。

- 我提升自己编码水平的理解是从《程序设计实践》（The Practice of Programming）这本书开始的，这本书的作者是 Brian Kernighan 和 Rob Pike，这两个人都出身于大名鼎鼎的贝尔实验室，参与过 Unix 的开发。
- 如果你想从日常开发中提升自己的效率，可以读一下《卓有成效的程序员》。假如你不曾思考过这个问题，这本书会让看到一些不同的工作方式，我也给这本书写过一篇书评。不过，这本书里的技巧太具体了，所以，有一些已经有些过时了。

设计

- SOLID 原则是一种面向对象软件设计原则。早在1995年，Robert Martin 就提出了这些设计原则的雏形，然后在他的《敏捷软件开发：原则、实践与模式》这本书中，比较完整地阐述了这五个原则，后来，他有把这些原则进一步整理，成了今天的 “SOLID”。有了设计原则做基础，这本书后面讲了设计模式，理解起来就容易多了。虽然书名是关于敏捷的，但这是一本讲设计的书。
- 设计和架构有什么区别？2017年，Robert Martin 出版了《架构整洁之道》（Clean Architecture），他在其中告诉我们，二者没有区别。所以，这也是一本关于设计的书，给出了 Robert Martin 对设计的最新理解。你可以把它看成《敏捷软件开发：原则、实践与模式》的修订版。
- 《设计模式》**不推荐阅读**，它是设计模式的开山之作，但它的起点是 Erich Gamma 的博士论文，其写作风格偏向学术，而且中文版翻译得也很一般。这里将它罗列出来只是因为其历史重要性。如果你想学习设计模式，现在有一些更容易入门的书，比如《Head First 设计模式》。
- Martin Fowler 的《企业应用架构模式》将软件开发当时常见的解决方案汇集成模式，今天看来很多模式已经习以为常，但当年出场可是技惊四座的。从这本书的名字你不难看出，它出版的年代是企业级开发盛行的年代。Martin Fowler 一直认为这本书没有写完，希望能够继续更新，但不知道何时能看到这本书的新版。
- 《Unix 编程艺术》也是一本讲软件设计的书，只不过，它选择的切入点是 Unix 中的设计，从中你可以学到“只做一件事，把它做好”、“文本化”等编程理念，有助于你改善日常的工作。这样的书，也就只有 Eric Raymond 这样沉浸编程几十年的人才能写出来。

工程实践

- Kent Beck 有一本知名的软件工程之作《解析极限编程》（Extreme Programming Explained），它介绍了一种软件开发方法：极限编程。但更重要的是，今天很多主流的软件开发最佳实践都是从这里出来的。这本书可以理解成诸多最佳工程实践的总纲。
- Martin Fowler 在1999年写下软件行业的名著《重构：改善既有代码的设计》（Refactoring: Improving the Design of Existing Code），把重构这个小圈子实践带到了大众视野。2018年底，Martin Fowler 时隔近20年后，又写出了《重构》第二版。把他对这些年行业发展的新理解融入到重构实践中。重构应该有个目标，这个目标就是“重构成模式”，而这也是一本专门的书：《重构与模式》（Refactoring to Patterns）。
- 《测试驱动开发》是 Kent Beck 为世人展示 TDD 做法的一本书。它好的地方需要自己体会，Kent Beck 并没有显式的讲出来，比如：任务分解。
- Jez Humble 和 Dave Farley 的《持续交付》（Continuous Delivery）让持续集成再进一步，将生产环境纳入了考量。乔梁，他是《持续交付》这本书的中文版译者，而且在这本书出版近十年后，他自己写了《持续交付 2.0》，把自己多年来关于持续交付的新理解整理了进去。
- 说到遗留代码和测试，我推荐一本经典的书：Michael Feathers 的《修改代码的艺术》（Working Effectively with Legacy

Code），从它的英文名中，你就不难发现，它就是一本关于遗留代码的书。如果你打算处理遗留代码，也建议你读读这本书。这本书我也写过书评，你可以了解一下我对它看法。

领域驱动设计

- Eric Evans 2003年写了《领域驱动设计》，向行业介绍一下 DDD 这套方法论，立即在行业中引起广泛的关注。但实话说，Eric 在知识传播上的能力着实一般，这本关于 DDD 的开山之作，其写作质量却难以恭维，想要通过它去学好 DDD，是非常困难的。所以，在国外的技术社区中，有很多人是通过各种交流讨论逐渐认识到 DDD 的价值所在，而在国内，DDD 几乎没怎么掀起波澜。

- 2013年，在 Eric Evans 出版《领域驱动设计》十年之后，DDD 已经不再是当年吴下阿蒙，有了自己一套比较完整的体系。Vaughn Vernon 将十年的精华重新整理，写了一本《实现领域驱动设计》，普通技术人员终于有机会看明白 DDD 到底好在哪里了。所以，你会发现，最近几年，国内的技术社区开始出现了大量关于 DDD 的讨论。

- 因为《实现领域驱动设计》实在太厚，Vaughn Vernon 又出手写了一本精华本《领域驱动设计精粹》，让人可以快速上手 DDD，这本书也是我向其他人推荐学习 DDD 的首选。

产品与需求

- 精益创业是 Eric Ries 最早总结出来的。他在很多地方分享他的理念，不断提炼，最终在2011年写成一本同名的书：《精益创业》。如果说精益创业是理论，《精益创业实战》这本书则给了你一个操作流程。

- Mike Cohn 是敏捷理念的一个重要传播者，我们在讲测试金字塔时，提到了他的著作《Scrum敏捷软件开发》（Succeeding with Agile）。敏捷开发有两大流派：一派是工程实践，另一派是管理实践。如果你对 Scrum 这类管理实践感兴趣，可以读一下这本书。

- 如果你对用户故事这个话题感兴趣，推荐阅读 Mike Cohn 的两本书《用户故事与敏捷方法》（User Stories Applied）和《敏捷软件开发实践 估算与计划》（Agile Estimating and Planning）。

开发文化

- 软件行业里有一本名著叫《人月神话》，这算是软件开发领域第一本反思之作。今天，我们讨论的很多词汇都出自这本书，比如，没有银弹、焦油坑等等。虽然这本书出版于1975年，但其中提到的问题，依然困扰着今天的程序员。

- 开源概念的提出者 Eric Raymond，他的《大教堂与集市》推开了开源大门。今天开源软件已经成为程序员日常工作的一部分，但如果没有 Eric Raymond 这些人的努力，我们还必须与复杂的企业级软件搏斗。了解一下开源的历程，可以帮助你更好地理解今天的幸福。

- 程序员应该如何做，Robert Martin 也写了一本书《程序员的职业素养》（Clean Coder），其中对大多数程序员最重要的一点建议是，说“不”。

软件开发拾遗

- 高德纳的《计算机程序设计艺术》肯定是一套程序员都知道，但没几个人读完的书。算法的讲解经过几十年已经有了很好的发展，如果学算法，肯定有更好的选择。如果你想看图灵奖获得者如何从根上思考问题，不妨找来这套书来翻翻。

- 《快速软件开发》（Rapid Development），**不推荐阅读**。在这本书中，作者首次提出了解决集成问题的优秀实践：Daily Build，每日构建。通过这个名字，我们便不难看出它的集成策略，即每天集成一次。它其中很多实践在当时是先进的，但今天看来有些落伍了。如果你只想从中收获一些理念性的东西，可以去读读。

- 《C 程序设计语言》、《Unix 编程环境》等出自贝尔实验室大师级程序员之手，他们的书都值得一读，其中的内容今天看来可能有些过时，但他们解决问题的方式和手法却值得慢慢品味。

- 我在讲淘宝技术变迁时，提到了《淘宝技术这十年》，这本书算不上经典，但可以当做休闲读物。

技术之外

- 管理大师彼得·德鲁克有一本经典著作《卓有成效的管理者》，虽然标题上带着管理者几个字，但在我看来，这是一本告诉我们如何工作的书，每个人都可以读一下。
- 尤瓦尔·赫拉利的《人类简史》或《未来简史》，是我第一次学到“大历史观”这个说法，历史不再是一个个单独的历史事件，而是一个有内在逻辑的发展脉络。
- 《从一到无穷大》是一本著名科普著作，它向我们介绍了20世纪以来的科学进展。作者乔治·伽莫夫既是热宇宙大爆炸模型的提出者，也是生物学上最早提出“遗传密码”模型的人。虽然这本书出版自1947年，但以现在社会的整体科学素养，还是有必要读读这本书的。
- 史蒂芬·柯维（Stephen Richards Covey）的《高效能人士的七个习惯》，其中的理念我在专栏两个不同的地方提到过，一个是讲以终为始时，那段关于智力创造的论述，另一个是讲优先级时提到的艾森豪威尔矩阵。这本书值得每个人阅读，很多程序员欠缺的就是这些观念性的东西。
- 很多程序员都是科幻小说迷，编程和科幻，这两个都是需要想象力的领域。刘慈欣的《三体》，不说它给 IT 行业带来的丰富的词汇表吧，作为科幻小说来说，它就是一流的，值得阅读。它会让你仰望星空，打开思维。如果你对科幻小说有兴趣，推荐阅读阿西莫夫的《银河帝国》系列，这是科幻小说界的扛鼎之作，你会看到，一部出版于1942年的书里就有大数据的身影。
- 对于程序员来说，最好的工作状态就是进入心流，它会让你忘我工作。如果你对心流的概念感兴趣，可以去读米哈里·契克森米哈赖的著作《心流》，这位作者就是心流概念的提出者。

好，今天的复习就到这里，你有哪些经典的书可以推荐给这个专栏的同学呢？欢迎在留言区写下分享你的想法。

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。

精选留言

Wei

谢谢郑老师总结！

郑老师的博客也值得一看，大师成长之路啊，现在虽然下线了，但是我会想办法整理一下的：）

老师能否也介绍一下看技术书籍的方法心得？ 有时候太长或者不熟悉领域的书，效率比较低，我在想要边读边做笔记了。

2019-04-27 23:35

风翱

乘着书香节，在当当上购买了一些书籍，准备开始另一段历程。

2019-04-26 10:47

作者回复

加油！

2019-04-27 19:19

dreamhead

如果你数一下推荐书的数量，也许会发现我推荐的另外一本书。

2019-04-26 08:43

西西弗与卡夫卡

补充，马大叔的《分析模式》，国内少有人提及，我认为是一本长期被低估的书。最大的价值是把OO抽象提到了新的高度。看完我发现：很多看似不同的事物，都有类似的关系。听起来是一句放之四海而皆准的“废话”，只有结合许多实实在在的例子

以及思考，才有可能真正转变我们看待世界的方式

2019-04-26 08:37

作者回复

《分析模式》确实被低估了，可能是因为和大多数理解的技术有距离。

2019-04-27 19:59

desmond

《企业应用架构模式》对我帮助很大，不过此时市场上可能更需要一本《互联网应用架构模式》，郑老师能否给马丁福勒传个话，^_^

2019-04-26 08:26

作者回复

小声地说，老马不一定擅长。

2019-04-26 08:35

小小

《原则》，《代码大全》，王福强《Spring揭秘》

2019-04-26 08:24

智超

正想着用脑图整理一下自己已读和想读的书单呢。最近也对moco 做了tech spike

2019-04-26 07:03

北天魔狼

2019-04-26 06:28

hua168

辛苦老师整理了

2019-04-26 03:09

Zapup

这篇是读得最快，也是收藏最快的一篇

2019-04-26 00:40

作者回复

欢迎收藏和转发！

2019-04-26 07:54