

结束语 | 放下技术人的身段，用极限思维提升架构能力

2020-01-31 许式伟

许式伟的架构课

[进入课程 >](#)



许式伟

七牛云CEO

你好，我是许式伟。

我们一起度过了 **296** 天，学习了 **87** 篇文章，
阅读了约 **357,003** 字，收听了约 **15.7** 个小时的音频。

放下技术人的身段，用极限思维提升架构能力。



讲述：姚迪迈

时长 10:17 大小 9.43M



你好，我是七牛云许式伟。

这个专栏从去年 4 月份至今已经有 10 个月左右的时间，到了要和你说再见的时候了。感谢你的一路相伴，感谢你的坚持。也希望这些内容能够对你有所帮助。

从工程角度来说，架构师的存在几乎是一种必然。传统项目工程也有架构师的角色，只不过软件工程有其特殊性，它快速变化，充满了不确定性，所以架构师的重要性的比重会被进一步放大。



但是如何才能成为优秀的软件工程架构师？

传统的架构图书往往从架构思维开始。但是，我认为它们错了。这里面最关键的问题在于：

架构并不是“知识点”。

架构思维的确非常非常重要。但是，熟读架构思维并不足以让人成为一名优秀的架构师。

关于这一点，我经常拿中国传统的武学文化做类比。武功招式可以精确传授，是“知识点”，掌握了就是掌握了，理论上可以做到分毫不差。但是，架构不是武功招式。它更像内功，它不是 0 和 1，没有清晰的掌握和没有掌握这样泾渭分明的区别。

在架构能力上，没有最好，只有更好。

这是为什么我们的架构课并不是从架构思维开始，而是采用双线结构。它基本上围绕着以下两个脉络主线来展开内容：

如何从零开始一步步构建出整个信息世界；

在整个信息世界的构建过程中，都用了哪些重要的架构思维范式，以及这些范式如何去运用于你平常的工程实践中。

这两大脉络相辅相成。

首先，我们通过还原信息世界的构建过程，剥离出了整个信息世界的核心骨架，这也是最真实、最宏大的架构实践案例。

其次，我们结合这个宏大的架构实践来谈架构思维，避免因对架构思维的阐述过于理论化而让人难以理解。

最后，架构就是对业务系统的正交分解。因此，整个信息科技的演化过程，自然而然形成了分层：基础架构 + 业务架构。

基础架构的产生是对业务架构不断深入理解的过程。越来越多的共性需求从业务架构抽离出来，成为信息科技的基础设施。

作为架构师，我们需要坚持对业务进行正交分解的信念，要坚持不断地探索各类需求的架构分解方法。这样的思考多了，我们就逐步形成了各种各样的架构范式。

这些架构范式，并不仅仅是一些架构思维，而是“一个个业务只读、接口稳定、易于组合的模块 + 组合的方法论”，它们才是架构师真正的武器库。

这个武器库包含哪些内容？

首先，它应该包括信息科技形成的基础架构。努力把前辈们的心血，变成我们自己真正的积累。光会用还不够，以深刻理解它们背后的架构逻辑，确保自己与基础架构最大程度上的“同频共振”。

只有让基础架构完全融入自己的思维体系，同频共振，我们才有可能在架构设计需要的时候“想到它们”。

这一点很有趣。有些人看起来博学多才，头头是道，但是真做架构时完全想不到他的“博学”。

从体系结构来说，这个基础架构包含哪些内容？

其一，基础平台。包括：冯·诺依曼体系、编程语言、操作系统。

其二，桌面开发平台。包括：窗口系统、GDI 系统、浏览器与小程序。当然我们也要理解桌面开发背后的架构逻辑，MVC 架构。

其三，服务端开发平台。包括：负载均衡、各类存储中间件。服务端业务开发的业务逻辑比桌面要简单得多。服务端难在如何形成有效的基础架构，其中大部分是存储中间件。

其四，服务治理平台。主要是以容器技术为核心的 DCOS（数据中心操作系统），以及围绕它形成的整个服务治理生态。这一块还在高速发展过程中，最终它将让服务端开发变得极其简单。

理解了这些基础架构，再加上你自己所处行业的领域知识，设计出一个优秀业务系统对你来说就只是轻车熟路而已。

这也是为什么这个架构课的内容结构是目前这个样子组织的。因为消化基础架构成为架构师自身的本领，远比消化架构设计原则，架构思维逻辑要难得多。

消化基础架构的过程，同时也是消化架构思维的过程。

把虚的事情往实里做，才有可能真正做好。

当然提升架构能力，不完全与成为架构师这件事情等同。

架构能力其实是一种属性，并不是只有架构师需要架构能力。软件开发工程师、SRE、甚至包括产品经理，都需要具备架构能力。

而架构师这个特殊的岗位，则是因为软件工程的需要而产生的。它从更全局的视角来把控工程的演进方向，以确保整个业务系统经历几年甚至几十年的迭代，仍然可以快速适应变化，而不至于老化。

成为架构师并不是一件纯技能的事情。

架构师需要放下技术人的身段，学会“共情”。与用户共情，理解用户的所思所想。与开发人员共情，理解技术人的所思所想。与公司共情，理解公司的发展诉求。

架构师需要学会“认同他人，反思迭代自己”。不要在不了解背景的情况下，随意推翻别人写的代码，而理由可能仅仅是不符合你的个人风格。当然反过来完全看不到项目的问题同样要不得，但这往往是受限于个人能力。要提升自己的架构水平，需要在实践中不断反思，不断在自我否定中成长。

不过我们今天把话题的重心收敛到架构能力上。怎么才有意识地通过训练来提升自己的架构能力？

实践对架构能力不可或缺。

在现实中，不少技术人员连函数规格都想不清楚。他们关心你是怎么“实现”的，但是却不关心“接口规格”是什么样的，接口规格是否符合函数的“业务语义”。

要提升架构能力，首先得做到规格为先，而不是实现为先。不要动不动问怎么实现的。要首先谈这个规格合不合理，是否存在多余的依赖。进一步来说，要多去谈这个函数（或软件实体）的业务范畴合不合理，是否应该换一个切分的姿势。

其实 review 自己的代码也是一种极佳的架构能力的提升手段。对自己刚刚写完的代码，去 review 它，从中找出问题。如此反复训练，就能实现自我能力的提升。

这其实是最高效的自我提升的方式。如果团队其他成员 code review 发现了你的问题，你得反思一下为什么自己发现不了。

很多人追逐实现新的业务系统，通过做新系统来找到满足感。但是实际上对架构师来说，恰恰是反复打磨既有系统是更加锻炼人的。如果你一年前实现的系统今天仍然很满意，那就需要警醒，因为这一年你在原地踏步。

在架构能力上，没有最好，只有更好。

这里我想分享一段我自己 review 自己代码的特殊经历。事情发生在我大学期间，当时的电脑相对我们大部分学生的购买力来说，还是非常昂贵。所以我和另外 4 个同学花了 7500 元合买了一台电脑。

结果就是，我们 5 个人轮流使用这台电脑。这意味着，我一周平均只能用一天多一点时间。再刨除上课时间，我真正能够上机的时间并不多。

而当时的我对编程非常着迷，所以我绝大部分的上机时间都花在编程上。作为物理系的学生，正常来说我学的编程语言是 Fortran。但我很快就把 Fortran 课程自学完了，并从老师口中和 Fortran 课程的附录中了解到了 C 语言。

于是我找物理系高年级的同学搞到了 Turbo C 2.0，开始翻遍学校图书馆的图书自学 C 语言。

为了能够高效利用一周只有一天多的上机时间，我尝试把程序写到纸上，并且提前进行 code review，确保尽可能多地发现程序中的错误，以减少上机过程中的调试时间。

在一次数学建模竞赛里，我和另外两位同学（廖唯桢和程胜峰）一组，其中用到了 Dijkstra 的最短路径算法。看完算法逻辑的介绍后，我直接一遍写成最终的代码，没有经过任何调试过程。

这让在旁边看着的同学廖唯桢觉得很神奇，问我是不是之前实现过 Dijkstra 算法。但其实于我而言，这不过是长期养成自我 review 代码习惯的结果而已。

这个习惯持续了三年之久。这三年里，我开始的时候都是先把代码写到纸上并完成 review，然后再到电脑上。但是到后期这个习惯就变了，我不再需要把所有细节都提前写到纸上，而是只需要提前准备好骨架：整个程序串起来的思路是什么。

我大学期间写过很多高代码量的程序。其实第一个 C 程序就不短，是一个仿 DOSKEY 的程序。后来也做过汇编语言的 IDE。这是因为学汇编的时候，发现没有好的汇编语言集成环境，于是就自己做了一个。至于为什么学汇编？是因为我想写一个 C++ 编译器，感受一下语言实现者的体验。另外，我也尝试在 DOS 操作系统下实现了一个图形界面库，并用它做了图片查看器和 MP3 播放器。

在代码量非常大的时候，人的脑容量就完全无法把这个实现装到头脑中。这时“规格重于实现”背后的意义就完全体现出来了。通过规格串起整个业务系统，以此把业务系统装到脑子里，这就是很朴素的架构“骨架”思维。

这不是一个假想实验。

它是我的亲身经历。这段经历启发我意识到极限思维对架构能力提升的重要性。

架构没有最好，只有更好。在极有限的上机时间里，在没有电脑的情况下，我们只能选择把更多的逻辑装进脑子里。

这个过程还可以更进一步。我们不断训练自己对不同业务领域的架构范式的理解。直至最终，我们头脑中可以装得下整个信息科技的骨架。

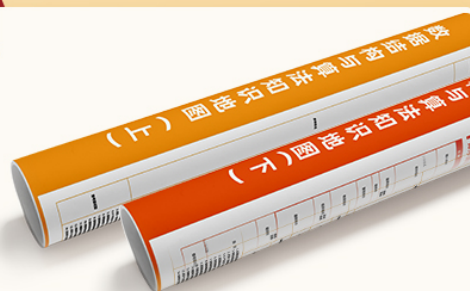
到那时，单就架构能力而言，你就是最顶级的架构师了。

备注：我在文末准备了一份调研问卷，也欢迎你点击下方的图片参与调研，期待你的反馈。

＝ 在 2 月 10 日前提交问卷，将有机会 ＝

得

算法地图



或得

极客时间课程阅码

价值 **¥99**



更多课程推荐

数据结构与算法之美

为工程师量身打造的数据结构与算法私教课

王争

前 Google 工程师



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。

精选留言 (14)

写留言



leslie

2020-02-01

10个月不容易：从开始坚持到最后；结束语早已看完，尾笔却一直不知如何去写；老师在付出的同时我也在坚持学习，最忙的时候一周5-6门的极客时间课程完全只能不断的压缩本就不多的睡眠时间。

借用老师的结束语“放下技术如的身段，用极限思维提升架构能力”：这个就像在问自己“什么是极客”？曾经一段时间陷入过这种困惑，为此看了不少极客时间的视频甚至问过...
展开



2



Jeff.Smile

2020-01-31

约一个半月跟完了，起初看许老师的课很其他人一样感觉很不同，许老师并不是像其他架构课一样讲一些看起来很前沿的技术细节，而是从0到100完成了架构世界的进化史，这比了解任何技术细节更重要，而且跟进课程的过程中解答了许多我们作为程序员进阶过程中的困惑，非常感谢许老师，期望再出新作！

展开



2



南山

2020-01-31

神往的境界，鼓起勇气向老师学习，迈进

展开



2



Jxin

2020-01-31

- 1.代码写在纸上再复现到机器上。栏主的坚毅非常人能及。
- 2.感谢分享，受益匪浅。

展开



2



Dimple

2020-01-31

感谢老师这么多个月的付出，新年快乐。武汉加油

展开 ▾



👍 1



py

2020-02-01

实践提炼到理论，理论反过来指导实践，相辅相成

展开 ▾



忆水寒

2020-02-01

这篇文章是最大的收获

展开 ▾



吴

2020-02-01

感谢

展开 ▾



梦醒十分

2020-01-31

受益匪浅!谢谢许老师!祝新年快乐!希望继续出新的课程!

展开 ▾



汪旭光

2020-01-31

感谢许老师的精彩分享!

展开 ▾



M凯雯

2020-01-31

打卡

展开 ▾





K战神

2020-01-31

全程跟了下来，有点不舍，
发现自己的架构观是多么的薄弱，
希望能够看到许大的精彩书籍

展开 ∨



aiueo

2020-01-31

完结撒花。从桌面应用程序到单体服务架构，服务端架构到服务治理，一路走来，收获满满。



Aaron Cheung

2020-01-31

规格为先 许老师的教诲非常受益

要提升架构能力，首先得做到规格为先，而不是实现为先。不要动不动问怎么实现的。要首先谈这个规格合不合理，是否存在多余的依赖。进一步来说，要多去谈这个函数（或...

展开 ∨

