

开篇词 | 怎样成长为优秀的软件架构师？

2019-04-10 许式伟

许式伟的架构课

[进入课程 >](#)



讲述：姚迪迈

时长 10:49 大小 9.92M



你好，我是许式伟。从今天起，我想和你一起来聊聊架构的话题。

开始之前，我先来和你简单介绍下我自己。

我是 2000 年开始工作的，曾经做过 WPS 的首席架构师，也在盛大从事过技术研究方面的工作，后来在 2011 年创立了七牛云，现在我是一名创业者、CEO。但不管角色怎么轮换，我觉得我的另一面始终是一名程序员、架构师（如果你想了解更多我的经历，可以观看下面的短视频）。



0:00 / 5:54

让我们来想象一下，如果把信息世界看成一座大厦，把程序员看成这个世界的建筑师，那么，现在的你在负责什么样的工作呢？

当我们把程序员类比成建筑师时，按照能力水平来分，我觉得大体可以分为三个层次：搬砖师、工程师、架构师。

软件搬砖师之名对应到建筑行业的建筑工人，他们的编程能力和业务基本上停留在堆叠代码，按照要求去实现功能需求的层面。

只要能让程序跑起来，能正确地实现业务逻辑，就可以称为“会编程”的人。有时候，我们也会看见程序员自称为“码农”“搬砖的”，虽然二者的工种不同，但从基础工作的相似度来说，确实有可类比的成分。

很多外行的人都会觉得程序员是一个很神秘的职业，但实际上程序员的基础门槛并不算高。我自己从 2016 年 2 月开始至今，一直在教几位 8~12 岁的小朋友学习编程。这个实践经验告诉我：小学生完全有能力学编程。而且，并不是只有部分小学生可以，而是任何一位小学生都可以学会。

然而，只让代码跑起来是不够的。这个世界是不断变化的，作为程序员，我们更多的时间是用来维护代码：增加新的需求，对已有的功能进行调整，修改之前代码遗留下来的问题，优化性能等等。

这是因为一个软件诞生之后，后续就是需要花费大量的代价去维护它，演进它。一个人是完全维护不过来的，需要更多的人，很多的团队一起协作。如果面临了员工离职、岗位调整等情况，还会导致软件代码在不同人之间流转。

所以，一些有追求的程序员会关注代码的质量。代码质量的评判可以有这样一些基本维度：可阅读性（方便代码流转）、可扩展性 / 可维护性（方便修改功能，添加新功能）、可测试性（质量管理）、可复用性（简化后续功能开发的难度）。

这一类致力于不断提升软件代码的工程质量的程序员，我们可以称他们为软件工程师。

工程师不会简单把写代码看作一门工作，把任务交代过去就完事。他们会有“洁癖”，代码在他们眼里是一种艺术，是自己生命的一部分。

他们会把写出来的代码改了又改，直到让自己满意为止。阅读和维护软件工程师写的代码会有一种赏心悦目的感觉。

但是，大部分商业软件都是一项极其复杂的工程，它们远比很多传统的建筑工程复杂得多，无论是涉及的人力、时间还是业务的变数都要多很多。

人力上，大部分大型的软件系统都有几千甚至几万人的规模，而这几千几万人中，却没有两个人的工作是重复的，他们都是在从事着前所未有的创造性工作。

时间上，只要软件还在服务客户中，程序员们的创造过程便不会停止，软件系统仍然持续迭代更新，以便形成更好的市场竞争力。

这些都与传统建筑工程的模式大相径庭。一幢建筑自它完成之后，所有的变化便主要集中在一些软装的细节上，很少会再发生剧烈的变动，更不会持续地发生变动。但软件却不是这样，它从诞生之初到其生命周期结束，自始至终都在迭代变化，从未停止。

所以，光靠把控软件工程师的水平，依赖他们自觉保障的工程质量，是远远不够的。软件工程是一项非常复杂的系统工程，它需要依赖一个能够掌控整个工程全局的团队，来规划和引导整个系统的演变过程。这个团队就是架构师团队。

软件架构师的职责，并不单单是我们通常理解的，对软件系统进行边界划分和模块规格的定义。

从根本目标来说，软件架构师要对软件工程的执行结果负责，这包括：按时按质进行软件的迭代和发布、敏捷地响应需求变更、防范软件质量风险（避免发生软件质量事故）、降低迭代维护成本。

那怎么才能成长为优秀的软件架构师？软件架构师和软件工程师最根本的差别又在哪里？我认为关键在于四个字：掌控全局。

掌控全局，就是对系统的全貌了然于胸。从传统的建筑工程来说，建筑架构师并不单单要会画建筑图纸，而是要对地基构建、土质、材料、施工工艺等等所有有可能影响建筑质量的因素都要了然于胸。

掌控全局，并不是无所不能，不是成为全栈，怎么做到掌控全局？核心在于对知识脉络的体系化梳理。这是架构能力构建和全面提升的关键。这种方法不单单是在软件工程中适用。

比如学数学，我个人非常喜欢做的一件事情是自己去推导书上所有的公式。每一个公式我都亲自推导而来。

这样做的核心意义在于，我在尝试从 0 开始，去构建整个精彩纷呈的数学世界，整个数学发展史在自己的笔下重新演绎了一遍，来龙去脉清清楚楚。有时候你甚至会推导出还没有学到的公式，但是在后面学到了。这种体验非常有趣而又让人满足。

是的，掌控全局的前提是：在自己心中去重新构建出整个世界。在这个过程中，你不需要一上来沉浸在某个技术的实现细节（除非它影响了你对这个世界构建过程的理解），但是你知道整个世界的脉络，知道整个世界的骨架。

这个时候，你对这个世界的感觉是完全不同的，因为，你已经成为了这个世界的构建者。

而架构的本质，不也正是构建和创造么？

作为一个软件行业的从业人员，我们可能接触各种各样的技术书籍。有讲编程语言的、讲数据结构与算法的、讲操作系统的、讲编译原理的、讲架构设计的，还有领域技术类的（比如数据库、存储、大数据、人工智能之类）。

大部分类别的技术书，多多少少都能够找到几本经典著作。但是，架构设计很可能是个例外，当我想推荐一本经典的架构设计书时，我并不能非常快速地想到应该推荐哪本。

从个人经验来说，我接触过的与架构相关的图书，大概有如下这些分类。

架构思维类。这类图书通常从一些著名的架构理论讲起，比如开闭原则、单一职责原则、依赖倒置原则、接口分离原则，等等。这种图书的问题在于过度理论化。计算机科学归根到底属于工程技术类，实践第一。

设计模式类。这一类图书则一下子进入架构的局部细节，每个模式的来龙去脉并不容易理解。就算理解了某个具体的模式，但是也很难真正做到活学活用，不知道还是不知道。

分布式系统架构设计类。这类图书通常从服务端的通用问题如一致性、高可用、高并发挑战等话题讲起，讲大型业务系统面临的挑战。这些知识是非常有价值的，但无法延伸到通用业务架构，对大部分企业的架构实践并不具备真正的指导意义。

重构类。这类图书主要讲怎么把坏代码一步步改进到好代码。我认为这是最实用的一类。但在没有优秀架构师主导的情况下，大部分公司的代码不可避免地越变越坏，直到不堪重负最后不得不重写。实际上，一个模块最初的地基是最重要的，基本决定了这座大厦能够撑多久，而重构更多侧重于大厦建成之后，在服务于人的前提下怎么去修修补补，延长生命。

这些架构类的图书并没有达到我个人的期望。因为它们都没有揭开架构设计的全貌。

我自己在职业生涯中前后大概做过十几次的架构类演讲，这也是我最为重视、重复次数最多的一类演讲。但同样地，这样零星的演讲对于传递架构设计思想来说，仍然远远不够。

所以一直以来，我就心存着这样一个念头：“要写一本不一样的架构类图书”。这个念想，也正是今天这个专栏的由来。

这个专栏的内容组织算是我的一次尝试。它和今天你看得到的大部分架构书并不太一样。我基本上围绕着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。

我想，每个程序员都有一颗成为架构师的心。所以，从内容设计来说，我希望这是一个门槛最低的架构设计专栏，也希望它可以帮助到想成为架构师的初学者，达成自己的目标。

在行文上，我会尽量避免深奥的术语，尽可能以通俗易懂的文字，来描述信息世界构建者们的所思所想。如果你在阅读的过程中遇到了理解上的障碍，非常欢迎你来给我留言，我将尽可能地根据你的反馈，做出必要的调整。

如果你已经成为了架构师，我也希望可以为你规避一些错误的经验。在过去的工作经历里，我看到不少架构师都会倾向于把架构看作一项纯技术性的行为。他们的工作流程是这样的：产品经理根据用户的需求做出产品设计，然后架构师再依据产品设计给出实现，也就是软件的架构设计方案。

在我看来，这其实是个误解。架构关乎的是整个复杂的软件工程，它关乎实现它的人，它又因团队的能力而异。

同时，架构也关乎用户需求，作为架构师，我们不只是要知道当前的用户需求是什么，我们还要预测需求未来可能的变化，预判什么会发生，而什么一定不会发生。预测什么不会发生最为重要，只有做到这一点，才能真正防止架构的过度设计，把简单的事情复杂化。

谈了这么多，那么，应该怎样成长为优秀的软件架构师？**我想，一靠匠心，二靠悟心。**架构设计并无标准答案，但我仍然希望把我这些年的所思所想分享给你，更希望这些内容能给你一些启发。

拼课微信：1716143661

许式伟的架构课

从源头出发, 带你重新理解架构设计

许式伟
七牛云 CEO



新版升级: 点击「 请朋友读」, 20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。

下一篇 01 | 架构设计的宏观视角

精选留言 (184)

写留言



郝林 置顶
2019-04-10

132

许大专栏必买, 好好学习!

展开



Jesse 置顶
2019-04-10

45

看老许的专栏 脉络清晰而沉稳, 语言不急不躁, 看得出 很多都是干货。
日拱一卒 不期速成。跟着老许一步一步成为架构师。

作者回复: 日拱一卒, 不期速成。说得太好了, 学架构如同练内功, 不期速成是最正确的姿态。



mark

2019-04-10

👍 50

许老板是国内第一个推广go语言的人

展开 ▾



大晶

2019-04-11

👍 21

35的一线程序员，很想成为架构师

展开 ▾



小明

2019-04-10

👍 13

搬砖党前来升级

展开 ▾



halweg

2019-04-10

👍 12

买了就是会了

展开 ▾



糊李糊涂

2019-04-10

👍 10

我的go语言领路人！

展开 ▾



郭高法

2019-04-10

👍 10

我是一个高级工程师，我是这么自己定位自己的。能把一个模块设计好，写出好的代码来。

公司总会给一些系统总体设计的工作，领导的栽培吧。虽然，设计的系统最后在大家共同努力下开发完成，能用吧。事后细想，总觉的不是自己想要的系统，没有一点成就感那是假的，但是，没有那种行云流水的快感，希望在学习课程和讨论中，能摩擦出思想的火...

展开 ▾



kisnotunde...

2019-04-10

👍 9

后端一枚，根据这些年的学习经验，方向和基础都很重要
半码农半工程，来学习许老师写下的经验，努力往架构师前进



geewonii

2019-04-13

👍 7

天赋平平无奇的前端搬砖工一枚。

看到这个课程总觉得自己接触这部分还太早。

看完推文才明白，是自己给自己设限了。...

展开 ▾



甘陵笑笑生

2019-04-13

👍 6

8-12岁小学生学习编程 希望分享一下教学内容和经验 这样一边搞架构 一边教小孩 谢谢

作者回复: 建议先让小孩学 Google blockly。

汉化版本网址: <https://playground.17coding.net/>



chapin

2019-04-11

👍 6

从13年学习go语言就开始关注许老板了，在我的眼中，许老板是布道师，开拓者，CEO，令在下着实佩服。未曾想到不差钱的许老板能来极客时间开课，大概是分享精神吧。支持一个下。

展开 ▾



永光

2019-04-16

👍 5

码农思考的是：功能怎么实现，业务逻辑怎么实现。（任务怎么完成）

工程师思考的是：怎样把自己的工作，干的完美，并从自己的工作中找到乐趣。（完成任务的同时提升自己）

架构师思考的是：怎样规划，让别人也尽量干得完美。（完成任务的同时帮助他人）

展开 ▾



浅蓝深蓝

2019-04-10

👍 5

匠心和悟心都没有怎么办

展开 ▾



A1

2019-04-10

👍 5

架构是一种行为模式，一种思维框架，虽然看起来针对具体事物缺乏针对性，但对于人建立大局观是有帮助的。现代企业，尤其是信息化企业，主要决策者更需要有架构思维，把可扩展的架构同组织和过程结合起来看，才能更敏捷更高效。



小奋

2019-04-14

👍 4

产品一枚，想了解架构，没有技术基础能听懂吗

展开 ▾

作者回复: 首先，产品经理和架构师是一体两面，我个人非常建议产品经理学习架构思维。其次，我会努力让没有技术基础的人听懂，但这确实实践上难免有疏漏，欢迎遇到问题提问。最后，我认为任何人都应该具备基础的编程能力，尤其是产品经理。而正如我序言中所说，这并不难，小学生都能够做到。



一路向北

2019-04-11

👍 4

19年，期待自己在技术方面的最大提升

展开 ▾



coder

2019-04-11

👍 4

许大大作为七牛的CEO，应该属于日理万机的那一类，所以我们这些作为nobody的人比较好奇，是什么促使您能在来 极客时间 去share自己的技术理念？😊😊😊

展开 ▾



sol

2019-04-10

👍 4

预判什么不会发生是个有意思的话题。

展开 ▾



Yu

2019-04-10

👍 4

像七牛go大佬学习

展开 ▾