



## 08 | 法则三：架构师如何在一定时间内最大化自己的增量价值？

2021-12-28 郭东白

《郭东白的架构课》

课程介绍 >



讲述：郭东白

时长 24:28 大小 22.42M



你好，我是郭东白。上节课我们讲了架构活动中需要重视对商业价值的考量。作为一个架构师，必须要创造足够的商业价值，才能保障自己职业的长期。

那么你作为架构师，该如何为你的公司、部门或团队提供可量化的增量价值呢？主要有扩大收入与减少成本两种路径。今天这节课，我们就结合几个真实的案例来具体分析一下

领资料

### 如何寻找扩大收入的机会？

有的架构师不关注软件之外的事情，比如很少关心公司或部门的收入。这种性格虽然让他专注于软件工作，但从长期来看，如果不去思考如何通过技术为公司创造商业价值，那就很难保持或扩大自己在团队的影响力，职业发展也可能受挫。



所以哪怕没有人向你施加为企业增加营收的压力，你也要主动思考，你或你的部门，该怎么为公司创造更多的营收。

那么我们该从哪里寻找扩大营收的机会呢？除了之前提到的 [深度理解用户心智](#) 和 [商业模式](#)，以及我们经常遇到的不同的横向问题，比如稳定性、安全、可测试性、可维护性之外，还有其它方法吗？

你可能听说过 **“在小数据里看大机会，在大数据里看小机会”** 这句话，这其实适用于我们所有技术人。

前半句指的是从个性需求中抽象出共性的机会。也就是从解决一个具体问题的过程中得到启发，并从中找到一个可以规模化应用的机会。比如做用户调研时，你发现有一个用户总是截图之后再通过微信把商品发给另一个人。那么你就会意识到，可以开发一个分享工具，通过小程序和 DeepLink 来获得更多的社交裂变。

后半句指的是**靠数据来打磨用户体验。也就是通过数据分析找到机会点，然后通过产品和技术的改进，不断提升转化减少损失**。这是算法和数字化运营的常见办法。比如淘宝 App 的首页跳出率是 2%，就是通过数据分析和打磨而不断提升的。

我特别强调一下前半句。事实上，很多著名的开源框架都是这么来的。离我们最近的一个例子就是 Docker。

虽然现在的 Docker 已经没有它如日中天时那么辉煌了，但毫不夸张地说，今天整个云原生生态的存在，引爆点就是 Docker。但谁能想到，Docker 最初只是为了解决一个环境打包问题呢！回想一下，从你进入计算机领域以来，你见过几个程序员会把心思放在运行环境打包上呢？

这个案例给我的启发就是，很多非常成功的技术人，他们往往就是**看到了别人忽略的小痛点，或者在别人不去排查的小异常上执着探索，才最终跨越了现实的障碍**。

举一个我们团队的例子。我们曾在海外做社交裂变，玩法与拼多多的拉好友砍价类似。但刚开始时，业务逻辑出了一个 Bug，用户没有达到提现条件（比如说拉到 20 个好友下载 App）我们就让用户提了现。当然这个 Bug 造成了一定的资金损失。

我想既然钱已经损失了，那么分析一下用户行为也是好的。比如到底是哪些人比较喜欢薅羊毛呢？这些人薅了羊毛之后会对我们的产品产生情感连接，提高忠诚度吗？

于是我们通过用户标签做了简单的行为分析，然后惊奇地发现：小城市里的年轻女性非常特殊，她们发现了这个提现存在漏洞后，就做了大量的分享行为，主动拉新。在她们的分享推荐下，很多人都下载了 App，而且这些人还没有加入裂变来薅羊毛。

更让我们兴奋的是，在我们修复了这个 Bug 之后，这些年轻女性的分享拉新行为依旧没有停止，而且她们拉来的用户的留存和购买率，竟然和大盘的拉新、留存差不多。也就是说，我们激活的这些年轻女性，是能够为平台带来大量用户增长的超级分享者。换句话说，对这个小 Bug 的执着，最终帮助我们找到了超级分享者。

后来我们就放大了这个玩法，故意放出“薅羊毛”的机会随机发给某些用户，尤其是年轻女性，并在其中寻找超级分享者。

接着，我们通过用户画像中召回与超级分享者最相似的人群，再用增强学习的方法放大这个人群的拉新效果。这样一来，这个小发现就能被大量复制了。

通过这个方法，我们发现前 100 万新增用户所用的拉新成本，连之前十分之一都不到，而且增长的空间也很大。这个方法玩了半年，让这个国家的互联网买家渗透率增加了 7%。

所以你看，**在别人容易忽视的痛点和异常点上，深度挖掘可能大规模复制的机会，不失为扩大收入的好方法。**

## 如何寻找减少成本的机会？

能赚钱固然好，但省钱对于一个公司，尤其是成熟公司来说，也同样重要。

先分享一个故事。Robert Scott 是一位伟大的探险家，他是人类第二支到达南极的队伍的队长。令人遗憾的是，他与他的四位队友都死在了从南极返回的途中。原因很复杂，但一个颇为重要的原因就是食物不足。他们只带了够 4 个人吃的食物，而队里却有 5 个人[1]。

这跟软件开发有什么关系呢？

一个公司在做商业模式探索的过程中，跟一个人要向前冲刺是一样的，都需要消耗能量。只不过对于一个公司而言，这个能量是钱。一旦钱耗尽，公司就饿死了。当然你可能会说，我家不缺钱。可以，难道你不缺时间吗？

所以作为一线的架构师，哪怕你不清楚公司的财务状况，但你的任何架构动作，都要考虑公司的研发成本，确保团队的规模在公司财务状况的允许限度之内。这也是经济学中所说的成本观念。

其实这种成本观念公司上下每个人都要有。对于一个公司而言，一切有限资源上的消耗都是公司在架构活动中要付出的成本，比如时间成本、人力成本、机会成本、计算成本等。

拿人力成本来说。很多做技术的人都有官本位思想，认为下属多多益善。似乎下属多了，他的管理水平就高了。相应地，工资和层级会更高。所以就会故意夸大和消耗更多的人力成本来换取自己的晋升。事实上，一个真正有经验的管理者很容易就能鉴别这种中饱私囊的蛀虫。面对这种情况，我的建议是，无论如何也不要参与到这种行为中去。

分享一个我的经历。08年美国发生次贷危机之前，我从耶鲁大学招了一个满分毕业的计算机系硕士生。他工作勤奋，产出也很好。但没过多久，公司就因为营收压力要强制裁员。由于他的入职时间最短，最终决定只能裁他。

他出生在印度一个普通家庭，从印度理工计算机系毕业后，又考到全球顶尖学府，这一路的艰辛可想而知。然而现实就是，在美国全职工作是拿绿卡留在美国的前提，我们一旦终止合同，那他只有几天时间去找工作。在经济危机发生的时候，这几乎是完全不可能的事情。

那两天公司上下都在裁员，当我叫他进办公室时，他已经预感到要发生什么，整个人都快虚脱了。他苦苦哀求，我都不知道该怎么回复他，只能沉默以对。这么多年了，我现在一闭上眼，还能想起他那一刻流露出的绝望眼神。

我分享这个故事，就是期望你知道，控制成本不是为了老板，而是为了我们自己。假设你总是以又大又笨重的系统去应对，那么在公司出现困境、经济出现低谷的时候，你的团队或企业可能就不复存在。相应地，架构师创造的软件也就不复存在，个人收入也会受影响。

从更理想化的角度来说，这么做也是为了自己的良心。我虽然无法预测未来，但在面对印度同事那一刻，内心还是非常自责。既然我不能给他一个稳定的任职机会，为什么还要让他加入？我的团队如果大幅盈利，那我不但不用裁员，说不定还可以收留其他员工。说到底还是我没做好。

我只是讲了人力成本的例子，其实时间成本、机会成本，包括技术的迁移成本，都是一样的。你作为一个架构师，要能省则省。

有些人会过分强调极客精神，事事追求完美。我觉得出发点是好的。**但在一个企业中，追求完美必须以成本可控为前提。**

## 性能优化案例串讲

关于第三条生存法则的内容我们就讲这么多，接下来分享一个我经历的性能优化的案例，来帮助你学以致用。

### 案例背景与分析

很多研发在做性能优化的时候，都会说“某个性能参数之前的 TP95 是多少秒，经过我优化之后，降低了多少秒”，以此来凸显自己的厉害。但这种玩法会让一个人逐渐忘记目标，只一心追逐性能上的极致。这就违反了我们今天讲的以商业价值为导向的架构原则。

我刚去 AliExpress 时，负责公司的全站架构。那时全站的性能非常差，但大家不知道这个差到底意味着什么，也不知道做性能优化到底要付出多大成本，又能带来多大回报。

当时我们已经有了全站的埋点。也就是说，我们有办法获取任意一个页面上跳出率和加载时长的关系，有办法获取任意一个页面上流量分布和加载时长之间的关系。我们也知道如果针对一个页面做优化，比如 JavaScript 优化、内容的静态化、图片压缩、动态加载等，都可以用来提升页面性能。

但问题就在于，如果针对每个页面的优化都要做投入。再加上维持这些优化效果，就要对页面的变更做限制和性能监控，那么付出的成本会非常高昂。

要知道，AliExpress 是个跨境电商网站，当时有 14 个站点，每个站点的前端代码都有微小的差异。一到大促，就要根据国家和语言一次性发布数千个页面。如果按部就班一个一

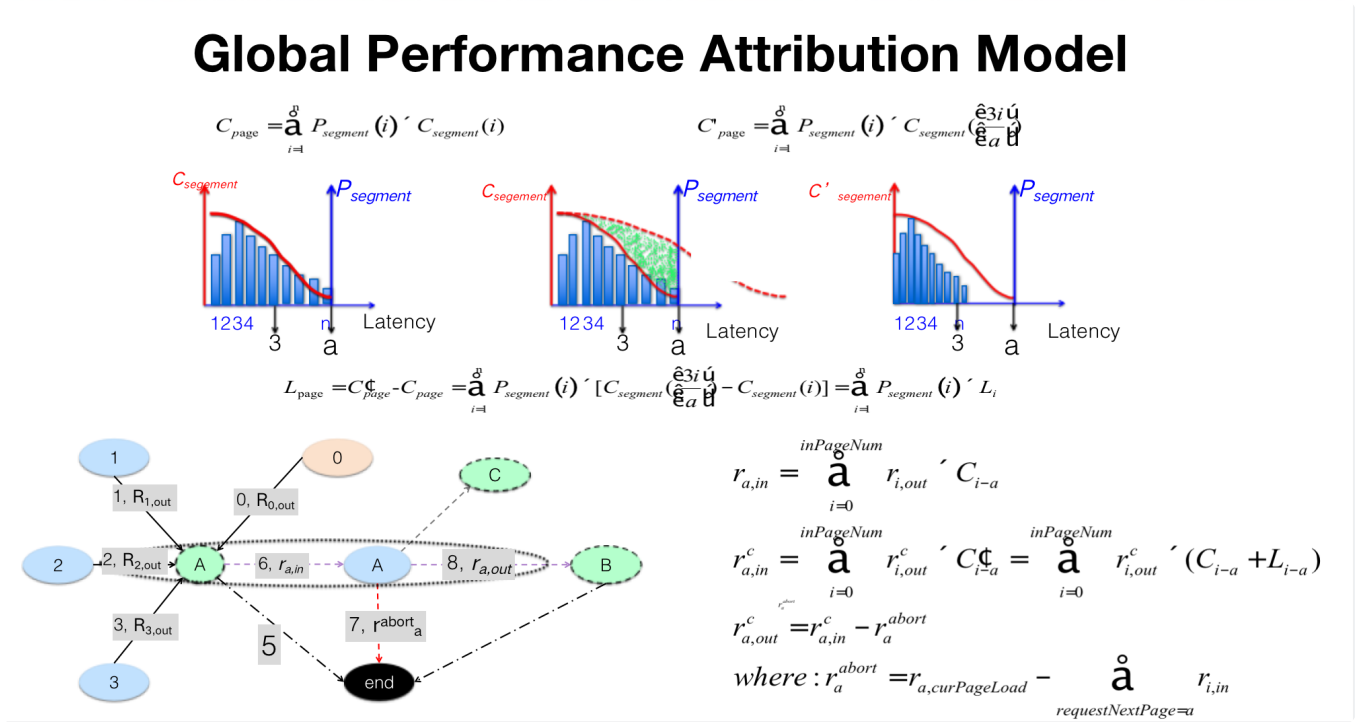
个页面去优化，哪怕配十几个全职研发，从头到尾做一年也跟不上发布的速度。

但结果呢？

我们只用了 6 个同学，兼职干了不到半年，就把全站的订单数提升了 10.5%。这是怎么做到呢？

我发明了一个方法，能够准确度量性能优化后每个页面的预期产出。而我们要做的，就是按预期产出除以预期投入成本，也就是预期回报的 ROI 来排序，再依次做优化。

具体的公式比较长，我就不在这里展开了。这个算法的核心原理展示在这张图里[2]：



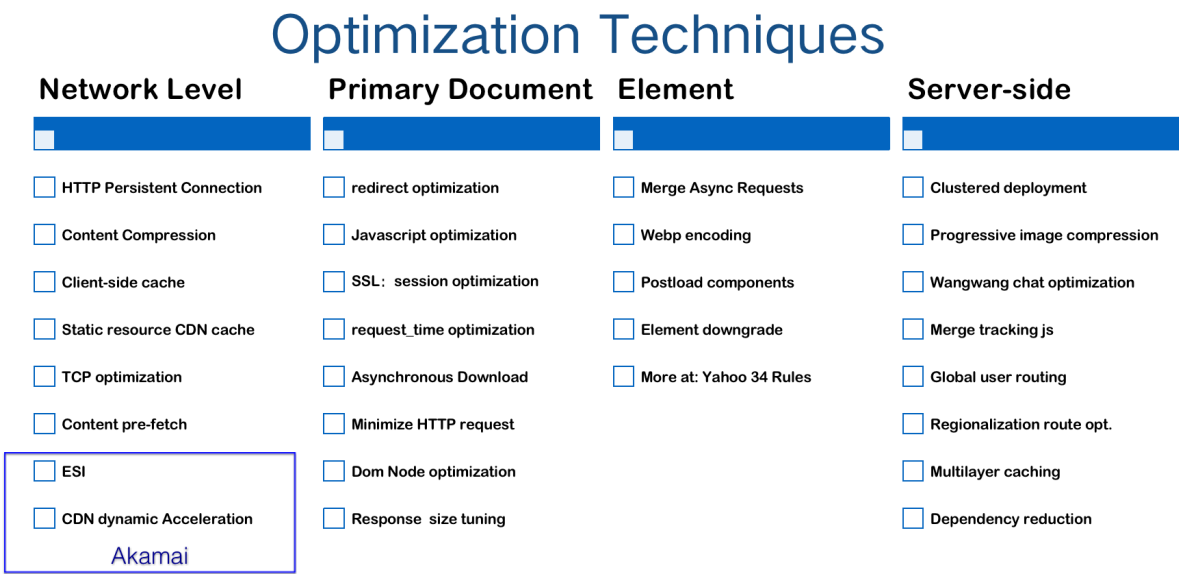
如图所示，我们根据每个页面转化率分布的直方图，以及预期的性能优化后的结果，预测出如果不做性能优化而损失的该页面的转化率。我们把这个预测值叫做**页面的性能损耗 Lpage**。

因为我们有全链路的转化漏斗和流量统计，所以如果优化某个页面，把性能损耗追回之后，那么这个优化对下游，以及对订单和 GMV 的预测，就可以通过大数据统计提前算出来。

所以我们就以优化订单数为目标做了架构规划，然后统筹我们可以做的一切优化动作。



如下图所示，本来完全不等价的优化动作，有的在网络层，有的前端，有的在后端。这下子就可以在一个指标上做比较了，因为我们的每一个优化动作最终都能被归因成了订单贡献。



之后我和团队把它做成了一个基于性能损耗的度量系统，共申请了 13 项专利。而这个理念也从最开始的指导架构规划，变成了性能归因、性能监控、转化分析和准实时的转化排查工具，并从 AliExpress 的一个部门逐步推广到阿里巴巴的其他部门。

我是如何践行生存法则的？

这个架构项目成功的关键，也正是我一直在遵照我这两节课介绍的生存法则。现在让我们一起通过这个性能优化的案例来检验一下，看看我如何用生存法则来指导架构活动。

另外，我们通过案例检验的过程，会重新总结并强调我们这两节课传递的一些核心观点。

第一，你作为一个架构师，在架构设计中要追求商业价值。

我们做性能优化时，不是单纯做性能指标的优化，而是一上来就以提升商业价值为目标。因此我们的优化目标是订单数，而不是一个技术指标。

第二，想要创造商业价值，就必须不断度量你创造的增量价值，这样才能确保自己处在价值创造的前沿。并且能够在一个相对未知的环境下，不断寻找自己的增值空间。

从这个项目的开始，我们就没想过要做全站的性能优化，而是发现了回报最大的单点，做针对性的优化。

为了做到这一点，我发明了准确度量性能损耗的公式，找到了部门层面的单一优化目标（订单数）。并把所有可能的优化动作，全部归因到这个单一的优化目标上去。

有了这个可度量的价值，我们就不再做地毯式的性能优化，而是做具有针对性的、回报最高的性能优化动作。

此外，我们也没有越做越宽。当发现性能优化的回报不够大了，我们就不再做性能优化了。而是换个赛道去创造价值：做现有网络的性能监控。

最好的证明就是当时我们和全球研究实力最强、监控能力最完善的 Akamai 合作。而且我们发现，我们对 CDN 网络的监控能力，在很多国家要远远胜过 Akamai。甚至到后来，我们干脆请 Akamai 的运维人员接了我们的部分报警。在此之后，我们又把应用方向再次扩大到业务转化问题排查等等。

### 第三，作为一个架构师，要最小化整个架构活动的成本。你要做的是：

确保最终方案的可行性；

寻找最优的实施路径，确保最终能够完成实施；

试图最大化最终解决方案的结构性，以最小成本放大你的产出。

我刚加入 AliExpress 时还没有很强的号召力，能调用的资源也极为有限。但当我发现性能优化是个突破口之后，就立即制定了一个可行的方案，而不是一个宏大的方案。

我当时仅仅把上面这个理念解释给了几个同学，然后我们就靠手算找到了回报率最大的几个页面，并且凭经验找到了投入产出比最大的优化点。这就确保了整个项目有非常强的可行性，同时也给了我们信心。

紧接着，我们迅速搭建了刚才提到的性能损耗的度量工具，验证了从工具中发现的优化点到订单回报的全流程。这样一来，不到一个月，整个项目的可行性就得到了验证。



但还是遵从同样的原则，为了确保投入最小化，我们仅仅升级了从优化点到订单的 A/B 能力。这样的话，我们的实施成本少，实施路径明确，所以项目可行性和合理性的风险就非常小。

最后，我还做了设计方案的结构化规划：先把相关理论和公式做了完整的推导，确保所有参与到项目的同学，都知道未来这个系统能给部门带来的核心技术价值，以及它对我们业务的支柱性作用。再把这些公式变成性能监控和性能损耗度量的工具。

这种结构化的思维方式使得我们在推广中的投入成本非常低，而且所有参与者都可以调用同样的工具来监控和度量性能的优化，以及实际产出。

可以看到，所有的核心能力，包括数据链路、监控、A/B 能力等等，在建设时都没有打折扣。虽然我们没有打算一上来就把整个系统构造完整，但我们还是为将来的扩展做了足够的考虑。这也是为什么我们的系统后来能够演变出新的能力的原因。

**第四，做架构和做业务一样，都不能靠饱和攻击取胜，而要靠对阶段性精确目标的最大化投入来取得进步。**

我们的系统虽然后来演变出了很多能力，包括保障业务稳定性、系统监控、业务问题和性能问题排查等等。但是在这个过程中，任何一个时期我们都只有一个目标。尤其是最开始，我的目标非常窄，就是通过性能优化带来订单量提升。我们甚至都不考虑优化带来的服务器成本降低。因为前者是放大收入，后者是节省成本，我们出手第一步只考虑放大收入，没有考虑节省成本这个附带目标。

**第五，不断寻找通过技术手段扩大收入的机会。**

这一点非常明显，整个案例都是通过纯技术手段带来订单增长的玩法。

**第六，不断寻找通过技术手段缩减成本。**

这一点要着重解释一下。这个案例并没有缩减成本，而是通过对性能的投入来获得商业回报的。

但我们案例执行中的每一步，都试图最小化我们在性能优化过程中的人力和时间成本，同时最大化商业回报。我们是怎么通过技术手段做到的呢？

在案例中，我们走的每一步都试图以最小成本去放大收入，技术上我们在不断计算性能损耗，在寻找最大的性能损耗和技术人日投入之间的比值来选择我们下一个优化点。

我们这个架构项目没有以节省成本为目标，是因为对于一个处在成长期的企业而言，挣钱永远比省钱更重要。我在 AliExpress 任职总架构师和 CTO 的前三年里，很少把注意力放在节省成本的手段上。

一个业务在高速增长的过程中，目标用户群体和用户心智，以及商业模式、供应链和运营手段，都在不断迭代。那么我们在快速奔跑的时候，最高优先级就是保障增长。哪怕增速慢下来，我们的优先级也依然是探索加速模式，重回高增长。而当一个业务到了成熟甚至是衰老期的时候，那就需要通过节省成本来扩大利润了。

## 故事的番外

你肯定想问我最后那个被裁的印度同事怎么样了。当他走出我的办公室后，我整个人都要崩溃了。我在我们整个楼跑上跑下，挨个敲每个研发管理者的门，告诉他们这个同事有多优秀，裁员对他来说打击会有多大。我跑了一下午，没有得到任何回复，原本都绝望了。

但是下班前有个主管来找我，说他团队里有个同事知道来龙去脉后，决定在那天下午提前退休，把自己的位置让给印度同事。这位印度同事后来在这个团队工作了很多年，我们也一直保持着友好的朋友关系。

这个世界还是有很多善良的人的，我期望你能记住这个故事，善待你周围的人。

## 小结

这两节课我分享了一个新的架构师生存原则：**你作为一个架构师，必须要创造足够的商业价值，这样才能保障你在企业长期存在的意义。**

为企业创造商业价值，说得更直白一点，就是你要为企业赚到钱。

拉长时间来看，我们每个人，都会被我们所在的企业以商业价值的视角来审视。与其让别人审视你，还不如自己审视自己，在日常的每一天都去看自己的不足，从商业价值视角上看到自己可以提升的地方。这就是我们这两节课反复强调的事情。

因此，在日常的架构工作中，就要从创造商业价值出发，理解自己所在团队或企业的商业模式，理解自己为企业创造的增量价值。然后通过技术手段，最大化自己或团队商业价值创造。

具体怎么做，其实我们不可能在两节课里穷尽所有的办法，但我们的确穷尽了所有的路径，那就是寻找扩大收入和减少成本的机会这两种。前者需要你不断探索、度量你的增量价值。而后者则需要你关注成本，平时能省则省，看准了机会之后再对精确目标做最大化投入，以取得最大的回报。而最终能做好这两点，还是要靠你日常养成的习惯。

## 思考题

能否分享一个你经历过的通过技术创造商业价值的案例呢？请注意，在分享这个案例的过程中，我建议你多强调这个技术突破点是怎么被发现的，具体的技术细节反倒可以讲得概括一些。

注释：

[1] Apsley Cherry-Garrard, The Worst Journey in the World – Antarctic 1910–13

[2] 郭东白、李彦超、桑植：“一种性能优化结果预测方法及装置。” 申请 / 专利号：CN201610550297.8；申请日期：2016-07-13

“

为整个企业创造商业价值，说得更直白一点，就是我们要为企业赚到钱。拉长时间来看，我们每个人，都会被我们所在的企业以商业价值的视角来审视。与其让别人审视你，还不如自己审视自己，在日常的每一天都去看自己的不足，从商业价值视角上看到自己可以提升的地方。



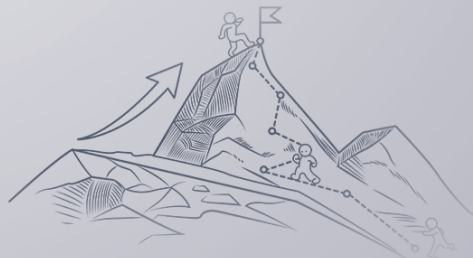
## 郭东白的架构课

郭东白 / 车好多集团 CTO、前阿里 P10  
浙大兼职教授和博导

”



识别二维码  
免费试读



如果今天这节课对你有帮助，欢迎你点击课程右上角的分享并赚钱按钮，把课程转发给你的同事或朋友，大家一起交流、进步。我们下节课再见！

分享给需要的人，Ta订阅后你可得 **20** 元现金奖励

 生成海报并分享

 赞 4  提建议

© 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 07 | 法则三：架构师如何找到自己的商业模式？

下一篇 09 | 法则四：为什么要顺应技术的生命周期？

## 精选留言 (22)

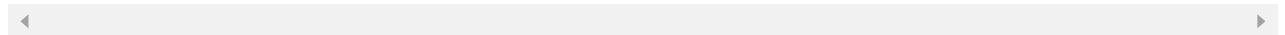
 写留言

**Right**

2021-12-28

这节内容看完，有被老师的能力给震撼到。能力强到离谱外，更难得可贵的是老师经常强调善良这一点，德才兼备。  
谢谢老师带来这么好的课程！

作者回复：谢谢！



共 2 条评论 >

 10



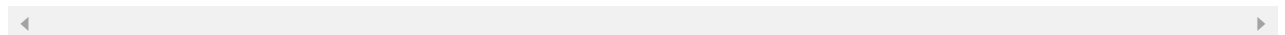
**罗均 - Jun**

2021-12-28

一如既往地老师的课程，震撼、感恩、不可思议！至心为那位印度精英的福报感到开心，更为老师的真诚而感动！

关于这节课的作业，学生有一个非常低端的“创新”，不知道能否入老师法眼，其实主要是想感谢一下淘宝。在汽车制造商的自动化测试中，需要模拟CAN总线和以太网环境给targ...  
展开 ∨

作者回复：很赞！其实慢慢的这种信心就会让你尝试更多的事情。继续！



共 2 条评论 >

 9



**罗均 - Jun**

2021-12-28

花了点钱，在X技术网上买了本课注释中老师申请的专利，花了几个小时阅读，终于对老师

的附图模型有点粗浅的理解了。

感觉老师的这套模型，简直就是data-driven operation的text book乃至Bible，需要再投入多点时间，再套用一些身边实际案例，才能再深入一点理解。

有个敏感的问题想请教老师，就是如果某公司在实际运营做性能优化时，也套用了老师...  
展开 ▾

作者回复：这种专利很难识别侵权的。Akamai的人还用了这个技术做了一个系统去卖钱呢。一般大公司很少打侵权官司，主要用来放手用的。

我们做了准实时的Dashboard。排查业务数据阴跌的时候很有效

共 5 条评论 >

👍 6



**bright21**

2021-12-28

支付宝的条码付，跨界的改变支付方式，改变了社会的支付习惯。在2012年左右的时候二维码的应用并不是很流行，基本都是在商品上使用作为一个商品的标示。当时在部门总监的支持下组建了一个小团队进行创新，头爆后大家觉得条码是一个很好的信息载体，可以作为支付账户进行支付相关的应用，这个就是现在的扫码付的雏形，支付宝做了跨界创新并申请了专利，但由微信发扬光大，做到现在人人在用。

展开 ▾

作者回复：赞！



👍 3



**AlfredLover**

2021-12-28

物超所值，有种追剧的感觉。其它课程都是等有空再看

作者回复：谢谢！



👍 3



**术子米德**

2021-12-29



\* 📦：无论公司是否给营收的压力，作为架构师，都要主动思考，如何能够帮助公司挣得更多

\* 😊：这句话听起来不错，做起来极难。回想我自己的所作所为，在给别人讲道理的时候



候，也会搬出类似的话，在自己做事的时候，总算在懒惰成性和内心各种小啾啾之间游...  
展开 ∨

作者回复：非常赞啊。你的留言虽然长，我也是非常喜欢，每次都会逐字逐句阅读。

这两句心得我很喜欢，一个总是审视自己内心的人最终会变得很纯粹。赞！

“近几年一直在问自己，做技术，持续做技术，到底是来自内心的喜欢，还是仅仅在这个行当而已。”

“一个软件系统的质量，是每个清爽易懂的代码角落。只有把代码写的简洁明了，显而易见，瞄一眼就懂，才能让缺陷在很早的阶段就暴露无遗。”

共 4 条评论 >

👍 3



一尾

2021-12-29

想问下老师，如何在已有软件体量较大的情况下，满足或者平衡大量新功能需求开发与保证软件/代码质量两者的？

作者回复：你的意思是一个请求非常高的系统有大量新功能研发吗？一般体量很大的系统相对稳定啊，业务一般不会对基础功能有大量的变更？不知道你们是什么场景，为啥会有大量的新需求？

Anyway，比较常见的招数是：先看一下是不是可以拆分一下服务或者模块，这样更改如果局限在单个服务上做发布对稳定性有帮助。其次大量的自动化测试和灰度发布很有效。

💬

👍 1



聪明的傻孩子

2021-12-29

这个课程给我一种追剧的感觉，实在是觉得非常精彩。作业分享一个我遇到的情况：之前在一家中型企业做研发，这家公司是一家做方案部署的公司，提供一整套方案给其他单位；但是运维人员每次都需要花费大量时间去重复其中的一些流程，然后当时的leader对于所有流程进行了一次了解，将原来的十几个流程简化为五个；其中的指标就是运维人员在现场部署所需要支持的时间，一开始一个项目基本会耗费三到五个月，在精简了流程...  
展开 ∨

作者回复：“指标就是运维人员在现场部署所需要支持的时间”这个很关键！正确的目标指引非常重要

**肖春**

2021-12-28

能力和人品真不是一般人能做到的！强

作者回复：谢谢！

**Helios**

2022-01-01

以前看过老师的分享，其中有一条是“有良知”，这几次课知道其中的含义，力排众议要为公司负责，尽自己能力要为员工负责，将心比心，老师做到了大家都知道的并且能做而不做的，致敬！

感觉大家应该和我一样，因为印度小哥的感动差点把正文忽略了。...

展开 ∨

**Geek\_fd0943**

2022-01-01

在经济学中，我看到了边际效用是具有终极威力的判断法则。在space x,我看到马斯克如何用成本效用法则运营一家公司，在ibm 我看到阿达姆用相似的眼光制定了计算机系统改进的优先级。这一次我看到冬白大佬用这样的思路演进了阿里的业务。以前我对这个策略唯一不确定的是 $\delta$ 回报/ $\delta$ 投入排序模型并没有考虑到投入的头寸，就像在交易风险模型中，风险的核心是投入的体量。但今天这个疑惑也释然了。这很大程度上只是目标的粒度的拆...

展开 ∨

**白彦彬**

2021-12-31

这节课看到学长系统化解决问题的高超本领，让我回想到读博士时候，在老教授指导下做科研的经历；大道至简，追本溯源，层层递进，建立量化分析模型，再设计验证案例，之后扩展至其他应用领域。我想系统化解决问题能力的培养，是读博的首要收获，而不是发表了多少文章。

展开 ∨

作者回复：的确是，至少你知道这个世界有一种“笨”办法是行的通的。

**qinsi**

2021-12-30

从老板的角度来看，业务量已经起来又不差钱的情况下做优化可能都不算个事儿。俗语有云：An engineer can do for a dollar what any fool can do for two.

作者回复：不过我好像还没碰到过觉得自己不差钱的老板。

**neohope**

2021-12-30

一说看数据，想起来一个小例子。去年做预算的时候，发现一个系统存储费用特别高，后面分析的业务的存储特点，发现近半年的使用率远远高于之前的使用率。于是将存储做了分级管理，半年前数据逐步迁移到了慢速存储，而且在只读的情况下慢速存储表现还不错，到现在为止业务方都不知道换了存储。这个系统体量比较小，今年也能省出一个人的成本；而且随着数据量逐年上升，后面能省更多成本。

展开 ∨

作者回复：这个漂亮！数据驱动，按需做Information Lifecycle management.

"到现在为止业务方都不知道换了存储!"

**王建设**

2021-12-30

通篇读来，非常享受。印度同事的故事大家还是最为关心哈哈。通过技术创造商业价值的案例，我能想到不多。倒是处于“互联网企业”，是技术“服务”商业价值的一些例子。比如多年前通过优化一款游戏加载速度来提高广告商业转化率，一开始商业目标就是非常明确，用商业目标来倒推性能优化点。刚开始还有些技术理性主义，感觉商业化太浓厚了，确实不去考虑公司最终的目标--商业价值。在自己的思维中容易走极端，太技术理想，不注重...

展开 ∨

作者回复：嗯。或许有一天，我们都会到一个低欲时代，在此之前，商业价值还是一个企业的重要目标。





**大林 Kralite**

2021-12-30

别的架构课是“术”，老郭的架构课是“道”啊！看了很多同学留言说有温度，很有同感。

作者回复: 哈哈



**一尾**

2021-12-29

感谢东白老师~ 这是一个有温度的课程~~

作者回复: 谢谢！



**Alex**

2021-12-28

十分感谢老师，这是我学习了这么久最有深度的一门架构师课程。

作者回复: 谢谢！



**ACK**

2021-12-28

非常赞同：能够准确度量性能优化后每个页面的预期产出。  
与  
高效能人士的七个习惯：要事第一  
计算机课程：阿姆达尔法则  
有异曲同工之妙。

展开

作者回复: 嗯，的确类似



**我是谁**

2021-12-28

这个法则真的让我耳目一新。公司的CTO或产品经理在设计系统架构或产品的时候，基本上只是站在技术角度或老板想法考虑的，至于能否给客户和公司创造价值，那是不用考虑的。所以也无需考虑各种成本。包括技术带头人，解决技术问题也总是希望面面俱到，缺乏针对性。

从老师这节课里体会到，无论哪个层面，都需要从创造收入和降低成本两方面来考虑考...  
展开

作者回复: 是的

