



下载APP



## 加餐 | 国庆特别放送：什么是业务架构，什么是基础架构？

2021-10-04 叶剑峰

《手把手带你写一个Web框架》

课程介绍 >



讲述：叶剑峰

时长 12:27 大小 11.41M



你好，我是轩脉刃。

相信你的公司一定有架构师这么一级岗位，但是肯定不只一个，而且岗位要求还各不相同，有的岗位要求熟悉 Linux 底层原理，有的岗位要求对业务有较强的敏感度。不知道你对这些岗位真实的需求有什么了解？为什么同样叫架构师有这么大的差异呢？职业发展方向有什么差异呢？今天我们就聊聊这个话题。

### 架构和开发



想要了解架构师之间的区别，首先得对架构师的工作内容有了解，所以我们先对比一线开发，看看架构师的价值核心是什么。

所谓架构，和一线开发最大的区别就在于是否有系统设计工作。

**一线开发的工作内容是在获取到一个细分需求后，思考如何用代码实现这个细分需求。**如果是初级开发工程师，拿到手的甚至可能是一个已经定义好接口和交互的技术方案，要做的事情就是往这个技术方案中填充内容，让一个功能可以如期运行。

如果职别高一点，是高级开发工程师，工作要求会比简单实现功能更复杂，因为拿到手的是一个功能需求，往往需要进行技术方案设计，梳理拆分成一个个小的功能需求点，然后将功能需求实现出来。不过仍然是以编码实现为主。

而架构师的工作价值已经不是体现在编码实现上，而更多是体现在设计上。

架构师面对的是系统，这个系统或大或小，可能是一个复杂功能模块、一个复杂业务，也可能是一个公司级别的基础服务，**但都有一个特点，就是比较庞大和复杂。**如何将这个庞大又复杂的系统清晰地分层、如何设计流程、如何拆分子系统、每个子系统负责什么、难点系统的技术应该选择什么技术，这就是架构师最大的价值体现。

## 基础架构和业务架构

虽然工作内容类似，但各个公司在架构师岗位上的称谓多种多样，比如云平台架构师、研发架构师、数据安全架构师、系统架构师等等，但是不管怎么起称谓，架构师的本质就是两种：基础架构和业务架构。

基础架构师，主要负责的是对基础服务的架构设计。**所谓基础服务，就是和业务无关，基本上所有业务都会使用到的服务**，比如数据库、缓存、对列等。

这些基础服务的重要性毋庸置疑，它的稳定性往往决定着整个公司的业务稳定性。试想一下，如果你的公司数据库出现不可用，是不是所有的业务线都会受到影响呢？而基础架构做的事情就是设计出合理的一套技术方案，来保证这些基础服务的可用性和稳定性。

业务架构师，主要负责的是业务服务的技术方案设计。你应该听说过这么一句话，技术是为业务服务的。是的，技术最终的价值就体现在业务实现上，**而业务架构师，核心作用就是让技术更好地服务业务。**

在过去的工作中，我更多承担的是业务架构师的职责，所以对于这个岗位也可以再聊聊我自己的想法，给你一些职业发展的参考。

## 你的技术服务什么业务

作为业务架构师，我觉得首先要清楚的是，你的技术是服务什么业务的。每个业务都带有行业属性，所以业务架构师的一个必要条件是了解你当前负责业务的行业。这里面不限于行业的技术发展趋势、竞品对手的动向，以及自己产品的后续发展方向。

有的人可能觉得这个没有什么必要。但是这点其实非常关键，最明显的体现就在技术选型上。**因为实现一个功能的方法有很多种，但是最符合我们自身业务的技术选型才是最优的。**

以如何选型和设计数据库为例，在分层模型上是否要增加服务层、在模块划分上是否要单独增加某个模块服务，这些的设计都需要考虑到行业的需求。

之前在交通领域，我们要设计一个信号灯管理系统，就需要考虑到国内国外的信号灯行业标准、各个信号灯厂商的标准、后续是否可能有地理未知查询的需求等，才能设计出比较好的数据库方案。

## 你的技术广度和深度如何

其次，你需要有足够的技术广度。业务架构师在进行技术选型的时候，要从多种可行性方案里进行选择，如果你没有足够的技术广度，对各种技术的优缺点的了解度不够，那么你的视野本身就被局限在一定范围内，大概率会做出错误选择。

但是如果所有技术都只有浅层了解，也是不够的。以前有个总监在给我们做培训的时候说，**技术人员最终的技术能力模型应该是一个大 T 字形，即在某个领域有足够的深度，而在多个领域有足够的广度。**

因为各个技术领域，深入到底层逻辑都有很多相似的地方，也就是说如果你在特长领域的钻研越深，想横向扩展技术广度也会越容易。这一点我一直铭记于心，也分享给你。

## 业务架构师和一线

最后我觉得非常重要的一点，业务架构师一定不能脱离一线。

如果不是在一线长期摸爬滚打过来，很难有接地气的设计。而在实现阶段，如果不时刻关注代码的质量，进行足够的代码检查，实现是有极大可能偏离设计的。

这种因为设计脱离一线，导致系统出现问题的例子真是比比皆是，你在实际工作中一定也遇到过不少。在工作过程中遇到了不合理的现象，其实都可以考虑是否是设计出现了问题，比如微服务过度导致两个人维护多个业务。

所以说，业务架构师不可能长时间脱离一线，否则就根本没有办法把控整个设计以及设计的落地实现了。

我可以分享一个我自己的习惯。**不管你能力有多强，接手或者到了新的一个业务中，前面 3 个月尽量不要做大的架构级别的修改**，因为不深度了解业务，没有足够时间了解一线的代码逻辑，是不可能做出好的架构调整的。

## 职业发展方向

刚才从工作内容的角度聊了聊基础架构和业务架构的区别，我觉得可以从职业选择的角度再谈谈这两个架构在之后发展方向的差别，毕竟在现在或者今后的某个时间，你在职场中一定会遇到这两个方向的职业选择问题。

不管你愿意从事基础架构还是业务架构，两者都是有淘汰周期的，都需要进行技术更新。这个是首先要说清楚的。所以如果你抱着哪个岗位更稳定的想法做选择，就不太靠谱了。

基础架构虽然听名字是底层，仍然有可能被淘汰。比如云和微服务的出现，对于之前的服务器运维方向的基础架构工作是一个很大的打击，现在的基础运维很多都新加了服务编排等工作。

而业务架构的更新淘汰更是常事，或许你在某个行业深耕多年，但由于各种政策原因，大行业都变化了，想不被淘汰，你的行业知识就必须重新补充了。

总而言之，你需要明白的是，没有什么会是一成不变的，只有不断更新自己，才可能跟上变化。清楚这一点，我们再继续聊。

基础架构的同学更大可能是往技术专家方向发展。他们**对技术的成就感更多来源于为某个软件或者某种语言增加了特性**，比如会追求成为 Apache PMC、微软的 MVP 等，他们的

研究是有可能改变某个技术行业的。所以如果你想走这个方向，必须有热衷于某个技术行业的觉悟。

而业务架构的同学更多可能是往业务管理方向发展，他们对技术的成就感更多来源于**创造出某个比较流行的产品**。比如有的业务架构同学就希望在教育行业有所发展，能设计并实现出改变教育行业的产品。

两种职业发展方向并没有优劣之分，而且不管哪个方向做到顶尖的人都是市场上非常稀缺的人。你在做职业选择的时候，更多的是要看清楚自己的兴趣所在，只有把自己的兴趣和工作相匹配，你的职业生涯才比较快乐。

## 系统架构

聊了那么多道理，最后再讨论一下做架构在术上的东西，希望能给你现在或者之后的工作一些参考。

这里我推荐一本书《系统架构 - 复杂系统的产品设计与开发》，作者 Edward Crawley 是麻省理工学院航空航天学位以及工程系统学的教授。之前说过，架构师的工作就是把一个复杂的系统进行拆分设计，这本书告诉带给读者的，就是如何做的一套思考并创建系统架构的方式，即一些思考系统的原则和定律。

我看完受益匪浅，这里也分享我认为非常有启发的几点。

歧义原则：系统架构的早期阶段充满了歧义。架构师必须解决这种歧义，以便给架构团队定出目标，并持续更新该目标。

架构师角色原则：架构师的角色是解决歧义，专注创新，并简化复杂度。

系统在初期的时候，最大的问题其实是充满了歧义，由于系统设计未成型，所有人对一个系统的理解大不相同，所以在沟通交流的时候是有很多歧义的。而**架构师必须解决这种歧义**，将所有人对系统的认知先统一起来，为整个架构的设计定出目标，并且不断更新团队认知，减少歧义，持续更新目标。

架构决策原则：我们要把架构决策和其他决策分开，并且要提前花一些时间来谨慎地决定这些问题，因为以后如果要想变更会付出很大的代价。

架构师在系统设计的时候会有很多决策要做，非常重要的一点，你需要**区分架构决策和其他决策，而且必须提前花时间来决定，是否需要考虑这些架构之外的决策。**

比如我们做架构设计的时候，是否要考虑团队的技术能力结构。以音视频业务的语言技术决策为例，目前这个领域最成熟的语言还是 C++，那么不管团队的技术能力结构如何，可能你就只能在架构上选择这个语言了。

而这个决策一旦做下去了，后续的影响可是非常大的，可能会导致整个团队的人员替换、整个业务的重构等。后期要想改变，会付出很大的代价。

Conway 定律：设计系统的组织，总是会产生出与该组织的沟通结构相同的设计。

还有一个我读了之后想明白很多事情原理的 Conway 定律。不知道你接手一个新的业务的时候，是不是会很唾弃，为什么这里这么设计、为什么这个地方需要拆分服务。这个定律就是告诉我们，**其实在设计的时候，我们会很倾向于把系统设计的符合当时的组织结构。**

比如用户这个信息，如果组织结构分为两个部门，一个部门负责乘客，另一个负责司机，那么服务就会很自然至少分为乘客服务和司机服务。但是司机和乘客实际上都是自然人抽象出来的模型，在只有一个司乘人员的部门中，可能就只会分为一个服务了。

了解这个定律以后，每次在接手新业务的时候，我都会去了解一下当时这个业务所在的人员组织结构，有时候对业务的架构会有更深刻的了解。

产品进化原则：系统必须进化，否则就会失去竞争力。

2 下 1 上原则：要想判断出对 level1 所做的分解是否合适，必须再向下分解一层，以确定 level2 中的各种关系。

书的内容很多，更多的内容需要你自行阅读。还记得第一节课使用思维导图来读取源码的技巧吗？其实我们也可以使用思维导图来读一本书。

这里附上我读这本书时制作的思维导图，你也可以边看这本书，边对照参考。也附上我之前在组内分享这本书的时候制作的 PPT，都放在项目 GitHub 上的

[geekbang/natinal\\_day](https://github.com/geekbang/natinal_day)分支上了，希望你能从中受益。

## 小结

今天咱们聊了基础架构和业务架构工作内容和职业发展的区别，相信你对这两个职位和如何设计系统架构有了更深的了解了。

你在过去的工作经历中，经历过哪些系统架构方面的问题，或者有过哪些系统架构的经验呢？欢迎在留言区分享你的思考和经历，我们下节课见~

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 1  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 09 | 自研or借力：集成Gin替换已有核心(下)

下一篇 加餐 | 阶段答疑：这些代码里的小知识点你都知道吗？

## 精选留言 (1)

 写留言



鸭补一生如梦

2021-10-06

《系统架构 - 复杂系统的产品设计与开发》

《架构师的自我修炼：技术、架构和未来》

《业务架构 应用架构 数据架构 实战》

请问这三本书各自适合的阶段和层次如何？

展开 ∨

