

## 二

# 46 架构重构内功心法第二式：合纵连横

46 架构重构内功心法第二式：合纵连横上一期我给你讲了我的架构重构内功心法的第一式：有的放矢，需要架构师透过问题表象看到问题本质，找出真正需要通过架构重构解决的核心问题，而不是想着通过一次重构解决所有问题。

今天我来传授架构重构内功心法的第二式：合纵连横。

## 合纵

架构重构是大动作，持续时间比较长，而且会占用一定的研发资源，包括开发和测试，因此不可避免地会影响业务功能的开发。因此，要想真正推动一个架构重构项目启动，需要花费大量的精力进行游说和沟通。注意这里不是指办公室政治，而是指要和利益相关方沟通好，让大家对于重构能够达成一致共识，避免重构过程中不必要的反复和争执。

一般的技术人员谈到架构重构时，就会搬出一大堆技术术语：可扩展性、可用性、性能、耦合、代码很乱……但从过往的实际经验来看，如果和非技术人员这样沟通，效果如同鸡同鸭讲，没有技术背景的人员很难理解，甚至有可能担心我们是在忽悠人。

例如：

技术人员说：我们系统现在的可扩展性太差了，改都改不动！

**产品人员想：咦，可扩展性，和扩胸运动有关吗？扩展什么呢？怎么会改不动呢？不就是找个地方写代码嘛……**

技术人员说：我们的可用性太差，现在才 3 个 9，业界都是 4 个 9！

**项目经理想：什么是 3 个 9，三九感冒灵？4 个 9 和 3 个 9 不就是差个 9 嘛，和可用有什么关系……**

技术人员说：我们系统设计不合理，A 业务和 B 业务耦合！

**运营人员想：咦，耦合，莲藕还是藕断丝连？A 业务和 B 业务本来就是互相依赖的**

## 呀，耦合为什么不合理呢？

上面的场景仅仅是个示例，并无嘲笑产品、运营和项目人员不懂技术的意思，而是说明有的技术术语并不是很好理解，在跨领域沟通时，很难达成一致共识。

除此以外，在沟通时还经常遇到的一个问题是**凭感觉而不是凭数据说话**。比如技术人员说“系统耦合导致我们的开发效率很低”，但是没有数据，也没有样例，单纯这样说，其他人员很难有直观的印象。

**\*\*所以在沟通协调时，\*\*\*\*将技术语言转换为通俗语言，以事实说话，以数据说话，\*\*是沟通的关键！**

以[专栏上一期的 M 系统]为例，我们把“可扩展性”转换为“版本开发速度很慢，每次设计都要考虑是否对门户有影响，是否要考虑对其他业务有影响”，然后我们还收集了 1 个月里的版本情况，发现有几个版本设计阶段讨论 1 周甚至 2 周时间，但开发只有 2 天时间；而且一个月才做了 4 个版本，最极端的一个版本讨论 2 周，开发 2 天，然后等了 1 个月才和门户系统一起上线，项目经理和产品经理一听都被吓到了。

以[上一期]的 S 系统为例，我们并没有直接说可用性是几个 9，而是整理线上故障的次数、每次影响的时长，影响的用户，客服的反馈意见等，然后再拿其他系统的数据进行对比，无论是产品人员、项目人员，还是运营人员，明显就看出系统的可用性有问题了。

## 连横

除了上面讨论的和上下游沟通协调，有的重构还需要和其他相关或者配合的系统的沟通协调。由于大家都是做技术的，有比较多的共同语言，所以这部分的沟通协调其实相对来说要容易一些，但也不是说想推动就能推动的，主要的阻力来自“**这对我有什么好处**”和“**这部分我这边现在不急**”。

对于“这对我有什么好处”问题，有的人会简单理解为这是自私的表现，认为对方不顾大局，于是沟通的时候将问题人为拔高。例如“你应该站在部门的角度来考虑这个问题”“这对公司整体利益有帮助”等。这种沟通效果其实很差，首先是这种拔高一般都比较虚，无法明确，不同的人理解也不一样，无法达成共识；其次是如果对公司和部门有利，但对某个小组没用甚至不利，那么可能是因为目前的方案不够好，还可以考虑另外的方案。

那如何才能有效地推动呢？有效的策略是“**换位思考、合作双赢、关注长期**”。简单来说就是站在对方的角度思考，重构对他有什么好处，能够帮他解决什么问题，带来什么收益。

以[上一期]的 M 系统为例，当时有另外一个 C 系统和 M 系统通过数据库直连共用数据库，我们的重构方案是要去掉两个系统同时在底层操作数据库，改为 C 系统通过调用 M 系

统接口来写入数据库。这个方案对 C 系统来说，很明显的一点就是 C 系统短期的改动比较大，要将十几个功能都从直接读写数据库改为跨系统接口调用。刚开始 C 系统也是觉得重构对他们没有什么作用，后来我们经过分析和沟通，了解到 C 系统其实也深受目前这种架构之苦，主要体现在“数据经常出错要排查”（因为 C 系统和 M 系统都在写同一个数据库，逻辑很难保证完全一致）、“要跟着 M 系统同步开发”（因为 M 系统增加表或者字段，C 系统要从数据库自己读取出来，还要理解逻辑）、“C 系统要连两个数据库，出问题不好查”

（因为 C 系统自己还有数据库）.....这些问题其实在 M 系统重构后都可以解决，虽然短期内 C 系统有一定的开发工作量，但从中长期来看，C 系统肯定可以省很多事情。例如，数据问题排查主要是 M 系统的事情了，通过 M 系统的接口获取数据，无须关注数据相关的业务逻辑等。通过这种方式沟通协调，C 系统很乐意跟我们一起做重构，而且事实也证明重构后对 C 系统和 M 系统都有很大好处。

当然如果真的出现了对公司或者部门有利，对某个小组不利的情况，那可能需要协调更高层级的管理者才能够推动，平级推动是比较难的。

对于“这部分我们现在不急”问题，有的人可能会认为这是在找借口，我也不排除这种可能性。但就算真的是找借口，那也是因为大家没有达成一致意见，可能对方不好意思直接拒绝。所以这种情况就可以参考上面“这对我有什么好处”问题的处理方法来处理。

如果对方真的是因为有其他更重要的业务，此时勉为其难也不好，还是那句话：换位思考！因为大部分重构的系统并不是到了火烧眉毛非常紧急的时候才开始启动的，而是有一定前瞻性的规划，如果对方真的有其他更加重要的事情，采取等待的策略也未尝不可，但要明确正式启动的时间。例如，3 个月后开始、6 月份开始，千万不能说“以后”“等不忙的时候”这种无法明确的时间点。

除了计划上灵活一点，方案上也可以灵活一点：我们可以先不做这个系统相关的重构，先把其他需要重构的做完。因为大部分需要重构的系统，需要做的事情很多，分阶段处理，在风险规避、计划安排等方面更加灵活可控。

## 小结

---

今天我为你讲了架构重构中的沟通和推动方法，希望你有所帮助。

这就是今天的全部内容，留一道思考题给你吧，有的人认为：架构师不是技术岗位吗，为何还要做这些事情，沟通和推动的事情让项目经理做就可以了！你怎么看这个观点？

[上一页](#)

[下一页](#)