

47 | 架构重构内功心法第三式：运筹帷幄

2018-08-14 李运华 来自北京

《从0开始学架构》



在前面的架构重构内功心法“[有的放矢](#)”和“[合纵连横](#)”中，我提到架构师需要从一大堆问题中识别关键的复杂度问题，然后有的放矢地通过架构重构来解决。但是通常情况下，需要架构重构的系统，基本上都是因为各种历史原因和历史问题没有及时处理，遗留下来逐渐积累，然后到了一个临界点，各种问题开始互相作用，集中爆发！到了真正要开始重构的时候，架构师识别出系统关键的复杂度问题后，如果只针对这个复杂度问题进行架构重构，可能会发现还是无法落地，因为很多条件不具备或者有的问题没解决的情况下就是不能做架构重构。因此，架构师在识别系统关键的复杂度问题后，还需要识别为了解决这个问题，需要做哪些准备事项，或者还要先解决哪些问题。这就需要我今天要和你分享的架构重构内功心法第三式：运筹帷幄。

经过分析和思考，我们可能从最初的 100 个问题列表，挑选出其中 50 个是需要在架构重构中解决的，其中一些是基础能力建设或者准备工作，而另外一些就是架构重构的核心工作。有了这样一个表格后，那我们应该怎么去把这 50 个问题最终解决呢？

最简单的做法是每次从中挑一个解决，最终总会把所有的问题都解决。这种做法操作起来比较简单，但效果会很差，为什么呢？

第一个原因是**没有区分问题的优先级**，所有问题都一视同仁，没有集中有限资源去解决最重要或者最关键的问题，导致最后做了大半年，回头一看好像做了很多事情，但没取得什么阶段性的成果。

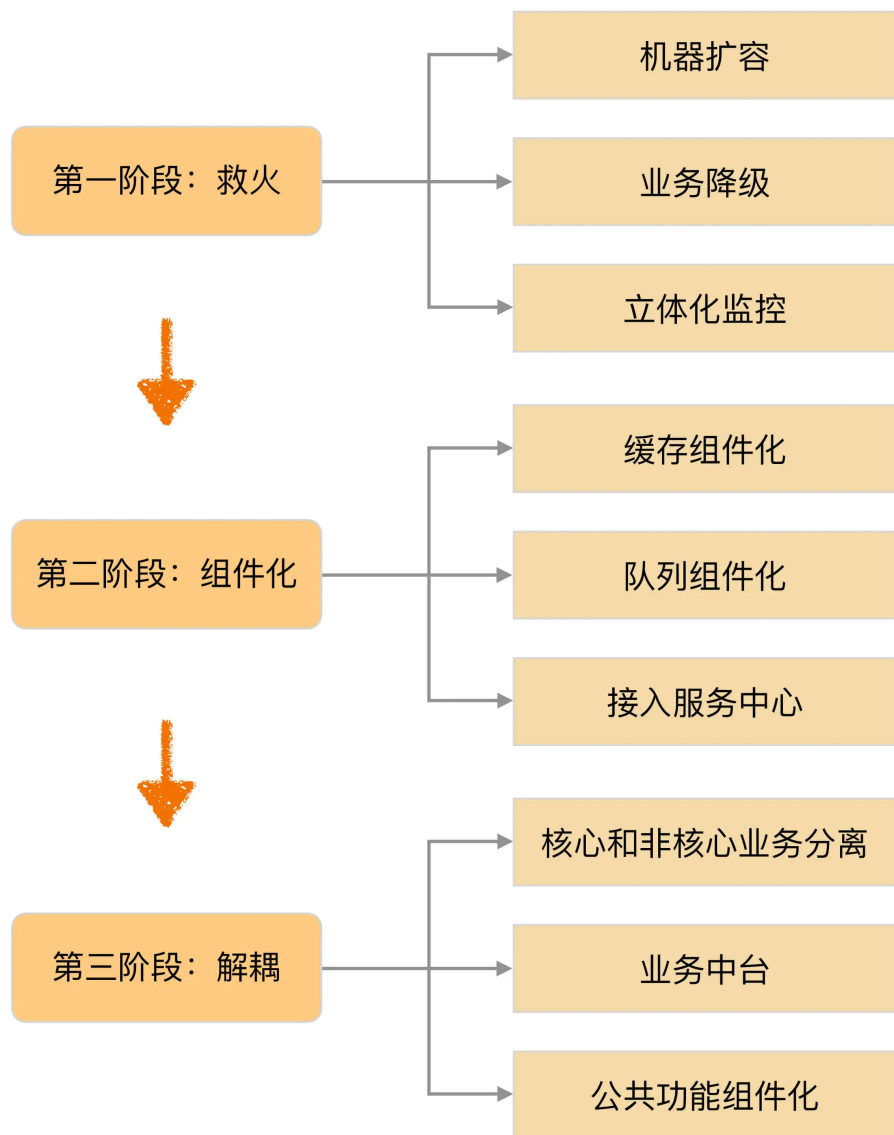
第二个原因是**没有将问题分类**，导致相似问题没有统筹考虑，方案可能出现反复，效率不高。

第三个原因是会**迫于业务版本的压力，专门挑容易做的实施**，到了稍微难一点的问题的时候，就因为复杂度和投入等原因被搁置，达不到重构的真正目的。

以 X 系统为例，在我加入前，其实也整理了系统目前存在的问题，大的项包括可用性、性能、安全、用户体验等，每个大项又包括十几二十个子项。但是实施时基本上就是挑软柿子捏，觉得哪个好落地、占用资源不太多，就挑来做，结果做了半年，好像做了很多功能，但整体却没什么进展。

后来我们成立了一个“X 项目”，在原来整理的问题基础上，识别出架构的核心复杂度体现在庞大的系统集成太多功能，可扩展性不足；但目前系统的可用性也不高，经常出线上问题，耗费大量的人力去处理。因此我们又识别出如果要做架构重构，就需要系统处于一个比较稳定的状态，不要经常出线上问题。而目前系统的可用性不高，有的是因为硬件资源不够用了，或者某些系统组件使用不合理，有的是因为架构上存在问题。

基于这些分析，我们制定了总体的策略，如下图所示：



可以看到，真正的架构重构在第三阶段，第一阶段和第二阶段都是为了第三阶段做准备而已，但如果没有第一阶段和第二阶段的铺垫，直接开始第三阶段的架构重构工作，架构重构方案需要糅合第一阶段和第二阶段的一些事项（例如，业务降级、接入服务中心等），会导致架构重构方案不聚焦，而且异常复杂。

为什么最终采用这样一个策略呢？主要还是为了集中有限的资源，某个阶段集中解决某一类问题。这样做首先是效率高，因为阶段目标比较明确，做决策和方案的时候无须进行太多选择；其次是每个阶段都能看到明显的成果，给团队很大的信心。比如说第一阶段的“救火”，做完之后，系统很少有因为机器过载、缓存响应慢、虚拟机挂死等问题导致的故障了；完成第二阶段的事项后，因为组件、外部系统故障导致系统故障的问题也很少了。完成前两个阶段后，我们就可以安心地做第三阶段的“服务化”工作了。

S 系统的重构做法也是类似，但 S 系统当时面临的主要问题就是可用性不高，并没有系统耦合的问题，所以我们当时的策略是“先救火、后优化、再重构”。“救火”阶段做了扩容（防止资源不足导致系统被压死）和 Nginx 一键切换功能（故障时快速切换）；优化阶段将一些明显的可用性问题解决（包括性能问题等）；重构阶段将原来的单点数据库改为多中心。

总结一下重构的做法，其实就是“分段实施”，**将要解决的问题根据优先级、重要性、实施难度等划分为不同的阶段，每个阶段聚焦于一个整体的目标，集中精力和资源解决一类问题。**这样做有几个好处：

每个阶段都有明确目标，做完之后效果明显，团队信心足，后续推进更加容易。

每个阶段的工作量不会太大，可以和业务并行。

每个阶段的改动不会太大，降低了总体风险。

具体如何制定“分段实施”的策略呢？分享一下我的经验。

1. 优先级排序

将明显且又比较紧急的事项优先落地，解决目前遇到的主要问题。例如，扩容在 S 系统和 X 系统中都是最优先实施的，因为如果不扩容，系统隔三差五会出现响应超时报警，一会来个过载报警，一会来个大面积不可用.....这些问题耗费大量的人力和精力，也就没法做其他事情了。

2. 问题分类

将问题按照性质分类，每个阶段集中解决一类问题。例如，X 系统的第二阶段，我们将多个底层系统切换到公司统一的公共组件，提升整体可用性。

3. 先易后难

这点与很多人的直觉不太一样，有的人认为应该先攻克最难的问题，所谓“擒贼先擒王”，解决最难的问题后其他问题就不在话下。这样看起来很美好，但实际上不可行。

首先，一开始就做最难的部分，会发现想要解决这个最难的问题，要先解决其他容易的问题。

其次，最难的问题解决起来耗时都比较长，占用资源比较多，如果一开始做最难的，可能做了一两个月还没有什么进展和成果，会影响相关人员对项目的评价和看法，也可能影响团队士气。

第三，刚开始的分析并不一定全面，所以一开始对最难的或者最关键的事项的判断可能会出错。

采取“先易后难”的策略，能够很大程度上避免“先难后易”策略的问题。

首先，随着项目的推进，一些相对简单的问题逐渐解决，会发现原来看起来很难的问题已经不那么难了，甚至有的问题可能都消失了。

其次，先易后难能够比较快地看到成果，虽然成果可能不大，但至少能看到一些成效了，对后续的项目推进和提升团队士气有很大好处。

第三，随着项目的进行，原来遗漏的一些点，或者分析和判断错误的点，会逐渐显示出来，及时根据实际情况进行调整，能够有效地保证整个重构的效果。

4. 循序渐进

按照前 3 个步骤划分了架构重构的实施阶段后，就需要评估每个阶段所需要耗费的时间，很可能会出现有的阶段耗时可能只要 1 个月，而有的却需要 6 个月，虽然这可能确实是客观事实，但通常情况下，按照固定的步骤和节奏，更有利于项目推进。我的经验是每个阶段最少 1 个月，最长不要超过 3 个月，如果评估超过 3 个月的，那就再拆分为更多阶段。就像 X 项目，我们先划分了阶段，每个阶段又分了任务子集，当任务子集比较小的时候，多个任务子集可以并行；当任务子集比较大的时候，就当成一个独立的里程碑推进。

小结

今天我为你讲了架构重构中分阶段有序推进的技巧，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，如果一个架构重构项目最后规划要 2 年才完成，你会怎么处理？

欢迎你把答案写到留言区，和我一起讨论。相信经过深度思考的回答，也会让你对知识的理解更加深刻。（编辑乱入：精彩的留言有机会获得丰厚福利哦！）

© 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。

精选留言 (36)



空档滑行

2018-08-14

2年的重构周期太长了，基本等于重新设计实现一个系统，就算架构师的沟通能力再强，也很难说服公司。

我觉得首先要看下自己的项目是不是计划做的太大，先把非关键功能重构砍掉；第二步剩下的功能拆分成最多半年一个周期，计划详细程度递减。一年后的计划都不需要对外公布，然后半年的计划一个里程碑，按文章里说的划分优先级，划分难易度然后推进。

作者回复：很好的思路👍

共 3 条评论 >

👍 53



xxl

2018-08-30

2年？江湖再见！



👍 39



陶邦仁

2018-08-14

将两年时间的重构拆分成多个子重构，促使重构快速见效，树立团队信心！

作者回复：通常如果说要重构两年，我的建议是重写可能更快😁😁

共 3 条评论 >

👍 18



何磊

2018-08-14

如果一个项目真的需要两年重构。要么是评估有问题，要么是这个项目不适合重构，只能重写。

作者回复: 英雄所见略同 😊😊👍



👍 12



凡凡

2018-08-15

两年时间太久，在互联网更新迭代这么迅速的场景下更加不适合，尤其创业公司，生命或许都维持不到两年。一般都要罗列重构点，按照难度从小到大，效果从大到小排列，然后安排合适的迭代计划。迭代过程中逐步建立信心，信任，适当调整计划。如果真的心里有个两年的计划，一定不要一下子全抖出来 😞

作者回复: 这算是厚黑术么？ 😊😊 我建议还是说出来，真要两年重构，那就直接重写了



👍 10



波波安

2018-09-05

两年时间太久了。有可能公司业务都发生了很大的变化。重构的规划可能并不满足新业务的发展。

作者回复: 正解

共 2 条评论 >

👍 6



小神david

2021-03-14

内功心法讲得非常好 ~ 收益很多 ~ 👍

作者回复: 架构的事情，不单单是写代码，也不单单是就技术论技术，很多人之所以觉得架构难，就是没有意识到这一点。



👍 5



成功

2018-09-07

二年的重构要看项目规模，规模不大，就重开发。如果项目本身的规模是3—5年，像雷达，战斗机控制系统等，用2年重构也值得。



👍 4



wmg

2018-08-14

两年的时间用于重构，时间太久了，对于这个瞬息万变的时代，不确定因素太多。即便一个系统真的烂到要两年时间重构，我倒觉得不如重建，那样还会省去很多约束，也许周期会更短。

作者回复: 是的，两年重构还不如重建



👍 4



王志

2021-05-18

两年来搞笑的吗？大部分系统重新开发半年就搞定

作者回复: 哈哈，看来大家都知道这个是坑：)

共 2 条评论 >

👍 1



谭方敏

2020-03-15

救火阶段：机器扩容，业务降级，立体化监控。

组件化阶段：缓存组件化，队列组件化，接入服务中心。

解藕阶段：核心和非核心业务分离，业务中台，公共功能组件化。

将要解决的问题根据优先级，重要性，实施难度等划分为不同的阶段，每个阶段聚焦于一个整体的目标，集中精力和资源解决一类问题。

这样做的好处在于：

- 1) 每个阶段都有明确目标，做完之后效果明显，团队信心足，后续推进更加容易
- 2) 每个阶段的工作量不会太大，可以和业务并行。
- 3) 每个阶段的改动不会太大，降低了总体风险。

分段实施策略

- 1) 优先级排序

将明显且又比较紧急的事项优先落地，解决目前遇到的主要问题。

2) 问题分类

将问题按照性质分类，每个阶段集中解决一类问题

3) 先易后难

4) 循序渐进

划分阶段，每个阶段又分了任务子集，当任务子集比较小的时候，多个任务子集可以并行；当任务子集比较大的时候，就当成一个独立的里程碑推进。

如果架构重构规划要2年，还是可以按照优先级排序，问题分类，先易后难，循序渐进的方式去推进了。

共 1 条评论 >



Jun

2019-12-22

先易后难也要慎重权衡。很多时候简单的方案都是hacky的，不利于长期目标。

作者回复：你要是能一眼看出长期目标是什么，而且确信你的判断一定准确那当然是直接考虑长远好了，但很多时候复杂度就在于你无法准确判断



钱

2019-09-05

课后思考及问题

如果一个架构重构项目最后规划要 2 年才完成，你会怎么处理？

两年时间比较长，期间研发、测试、产品，甚至我自己都可能换环境。太不可控，如果是我，我会这么做：

第一切割任务，粒度细到可控的范围，比如：每个月应该重点解决的问题有哪些

第二给任务排个优先级，先重点搞定重要紧急的

第三人员流动的根本原因一是钱少二是不开心，如果给不了钱要让大家开心点，工作氛围要轻松愉快点，节奏上要有张有弛

第四做好监控，有些任务如果延期，需要及时调整任务安排

第五任务安排到合适的人员，这个很关键，发挥每个人的长处，调动大家的积极性，千万不要明显的厚此薄彼，比如：说页面不重要没价值，后端逻辑才是核心，做页面的没啥价值，把人分成得力的和不得力的，当然，领导都会这么分吧！但可以不说不出来，想开除谁等开除时再说，不要告诉其他同事，这样会严重影响士气，让人想写不易维护的代码，自己发现bug也不想解决。

看评论2年周期重构不如重写，重构我的理解是要重写的只是原来的业务要覆盖，自己的代码逻辑变化是对研发来说的，对于业务是解决了他们的问题原来的业务继续支持 😊

本文核心观点

1-1：将要解决的问题根据优先级、重要性、实施难度等划分为不同的阶段，每个阶段聚焦于一个整体的目标，集中精力和资源解决一类问题



👍 1



云学

2018-08-16

看来架构重构和代码重构类似，先写测试用例来保证系统稳定，功能不能破坏，然后抽取接口，规范与周边模块的交互，最后对模块内部动手术，风险一直在掌控之中



👍 1



若水清菡

2018-08-15

第三阶段 解耦 那边是业务平台吧，不是业务中台吧

作者回复：是中台，当时的规划是把共性的地方抽取出来，其他业务也可以用



👍 1



艺超(鲁鸣)

2022-07-27 来自北京

重构如果需要两年的时间，大概率是原有系统架构设计出现了大问题。2年这个时间段，真的很难说服人，毕竟很多人换工作的频率基本都是2年。

作者回复：正解



初见

2022-06-30

2年重写吧，我经历过2次，都是原来系统是外包的，代码不规范，功能只是能用的程度，用户一多基本就垮了，做法是梳理业务后，进行重写，逐个拆分出业务重写，仅保留业务入口在老系统上，慢慢的，新系统建成，老系统只剩前端入口，最后直接切走，整体换到了新系统。

作者回复: 这样的经历很难得呀, 可以思考总结一下经验



Geek_9f2339

2022-03-31

记录一下内功心法的个人理解

有的放矢: 找问题, 找原因, 列方案

找到当前系统的问题所在, 根据问题分析出最合适的解决方案。

合纵连横: 找帮手

说服问题所涉及的团队, 人员。列出利害关系。让我们不是孤军奋战。

运筹帷幄: 确定方法步骤

对问题做出优先级, 执行计划等 (个人觉得这个可能是在合纵连横之前需要确认的事情, 因为这部部分是你说服架构重构的一个重要的证据)。

作者回复: 事实上第二步更多的是利益交换, 而不是方案是否可行



方人其

2022-02-15

将问题按照性质分类?? 一般分成哪些类? 参考下

作者回复: 一级的常见分类有: 性能类, 可用性类、可扩展类、管理类、流程类、测试类等, 还可以继续分类, 例如性能类可以分: 数据库类、缓存类、外部系统依赖类等



Geek_7de4c5

2021-12-08

如果重构要两年, 那么老板一定会问你, 完全重新开发要多久。大概率会重新开发。

作者回复: 哈哈, 是的, 2年也够重写了:)



