代码重构！你敢吗？| 技术头条

CSDN 4月8日

以下文章来源于大飞码字 ，作者大飞码字

大飞码字

在国内知名互联网公司工作快十年了，平时喜欢思考一些事情，所以开个公众号跟大家分享...

作者 | 大飞

本文经授权转大飞码字

今天讲述一个代码重构的经历。

2014年，我从基础架构部门，转调到业务部门。技术负责人想让我搞定业务系统的稳定性问题。

当时的业务系统确实存在不少问题，不过我初来乍到，对整体系统不熟悉，就想在熟悉一段时间后再动手。没想到，后面是事情自己找上了门。

那是一个周六的早上，我当时不在广州，而是去了深圳，去一个同学家。当时跟我同学聊的尽兴，就一直没看手机，间隔了一个多小时后，我打开微信一开，工作群里有几百个未读。看到我们技术负责人的头像一直在闪动，就意识到应该是出大问题了。

原来，是一个核心的业务系统出了一个bug， 影响到了一个重要的商户。

他们本意是给一个用户推送一条特定消息，消息里面还包含了一些隐私信息。不巧，一个新来的同学因为一个新的需求，修改了那部分的代码，引入了一个bug，导致本来是发个一个特定用户的信息，发给了一堆人。

商户相当不满，后来是部门的公关出面，才将事情平息下来，经理那边也因为这个事情，拉我们到办公室批了一顿。

技术负责人也压力山大。我们几个人，在会议室里讨论了很久，最后大家都觉得如果要比较好的杜绝此类的问题，除了要加强各种测试等措施外，还有一个，就是要重构现有的代码。

因为这个系统是最核心的业务系统之一，而且几经易手，当时的代码已经变得极难维护，里面各种 if else 的分支，还有长达一千行一个的函数，注释不全，文档也不足，要想长期的维护下去，这个技术债是非偿还不可了。

大家面面相觑，虽然知道重构是最好的解决方案，但大家都不想搞呀。

后来，我考虑到，初来这里还是要做些事情才能得到大家认可的，就硬着头皮，把重构的这个任务给接了下来。

01 CSDN

确定重构的范围

接下这个任务后，我和项目组的成员就开始分析这个系统。

发现这个系统的业务流程很长，涉及到几十个子系统(微服务)，还依赖几个外部门的服务。如果全部重构下来，估计一年都做不完，而且风险极大，重构一年的系统，然后再上线，谁敢呀，而且到那时，说不定黄花菜都凉了。

觉得这样肯定不行。

我们就重新梳理了一遍，把整个系统划分成了三个部分，我们发现中间部分的修改最频繁，出问题的频率也最大，就决定先重构中间流程部分的代码。

项目规划部分，我们对项目进行了分期，中间部分的重构作为第一期，其他两部分可以作为二期，三期项目来做。一个是可以极大地减少压力，使得的事情更加容易把握，另一个是间隔一段时间有产出也能给团队带来信心。

设计好验证的方式

当确认好重构的范围后，接下来的事情，就是要考虑如何来验证重构后的代码了。

这个是重构代码最重要的一个部分，如果没办法验证重构代码的正确性，你是不敢上线的，就算硬上了，也会睡不好觉。

一般重构代码的验证，可以采用测试代码，测试用例覆盖的方法。(这部分可以参考 《重构》)。但我们发现，我们要重构的这个部分，不能采用这种方式来验证。

因为业务逻辑很复杂，而且涉及到太多的外围系统，一个是测试用例很难覆盖全面，另外一个是没有办法可以很好的隔离外部系统的依赖。

我们分析了整个系统，发现这个系统的功能是，接受商户过来一个请求，然后进行各种权限，角色等的判断，再根据各个参数去各个依赖系统拉取数据，最后组装出一个新的包，再把这个新的包发

送到隔壁部门的下游系统。

后来，我们想了双流程验证的方案。

我们将重构部分的代码，全部封装起来，然后提供一个新的接口，一个请求进来后，我们分别执行旧的业务逻辑，也将请求发给新接口。在流程的最后，我们将新旧流程构造出的字段，进行逐个字段的对比。新流程只验证正确性，不做实际的输出。

为了保证验证的效果，验证要在线上进行，所以还要再结合后面的灰度流程。

尽一切努力，搞清重构代码的逻辑

当我们确定好验证方式后，接下来就是正式的工作了，重构代码。重写代码本身是不难的，但遇到的麻烦是，几乎没有文档，注释也很少，通过看代码只是搞懂了百分之五十左右的逻辑，还有一大部分的逻辑，无法理清楚。

后来，我们想到一个办法，把代码版本管理系统的log 全部拉出来。通过log我们找到了各部分逻辑不清晰的代码的负责人，然后一个一个的去跟他们聊，跟他们请教。运气好的是，大部分的人员都还在，中间还跟产品经理聊了不少，终于，把整个的逻辑搞懂了百分之九十几。

因为有了上面的双流程验证和下面灰度逻辑，我们觉得，可以开始上线验证了。

灰度，一定要灰度

接下来，就要开始我们的灰度验证流程了。因为故障的影响很大，所以我们灰度的特别小心。

我们内部有灰度系统，但内部系统的灰度粒度比较大，为了保险我们需要更小粒度的灰度，所以我们自己写了灰度的逻辑代码，直接嵌入到了系统里面。

一开始的时候，极度小心，几乎是一个商户，一个商户灰度的。灰度完后，我们每间隔一段时间，就分析一遍log和监控，看看有没有隐藏的问题。

最终，我们确实在这个灰度的过程中，发现了不少的问题，不过因为涉及的用户很少，都没有造成大的影响。

这种极小范围的灰度，大概持续了一周左右的时间，后面慢慢加快了灰度的速度。大概花了一个月的时间，覆盖了全部的用户。

中间过程，几乎没有出现什么大的问题，可以说是比较成功的一次重构。

05 CSDN

控制好各方预期

最后一个点，跟技术无关，是关于相关人员的预期，包括上级的预期，同级的预期，下属的预期。

我当时知道这个项目有难度，自己心里也没底，所以跟上级说去试一试，后来谈成可以在过程接受两次中等故障。当然最后结果比预期好，没有一次中等故障，只有过两次小故障。

同级这块，也是跟大家说，尽力去试试，不过确实不是很有把握，也算是降低了他们的预期。

对于下面的兄弟，我是跟大家说，这是一件可以稳固我们团队地位的事情，拼死也要拿下这一仗。后面大家都很齐心，一起完成了这个在当时看来挺难的一个任务。

这个策略，是我第一年工作的时候，我导师告诉我的，内紧外松。这样外面对你的预期是比较低，内部却很拼命的做，最后的结果，往往比较容易超出大家的预期。

我觉得这是一个很好的策略。

结语

最后，我们顺利完成了这次的重构任务，也做出了我们在新团队的影响力。后面再来回顾，发现我们做对了不少的事情。没有一上来就开干，因为信心不足，反而是小心翼翼，也因为信心不太足，在不断的降低外界的预期，最后一步一步，紧遵流程，获得了不错的结果。

作者：大飞。十年互联网人，资深架构师，技术leader。

【End】

热 文 推 荐

☞下过富士康工厂、做过华为外包，这位程序员是如何花 6 年逆袭成为技术大佬的?

☞苹果高通 5G 开战!

☞我是技术男，也曾创业过，也拿过风投……

☞身为程序员的父母，你年薪多少才能让“码二代” 不输在起跑线上

☞搜狗员工吐槽“统计加班时长裁人”，CEO 王小川破口大骂：“赶快滚”

☞漫话：如何给女朋友解释什么是系统可用性？ | 技术头条

☞EOS现状: 72%应用涉赌被列为高危，说好的诗和远方，你竟沦落成了这样？

☞ICPC 2019国际大学生程序设计竞赛，中国高校未能夺冠

☞刺激！我31岁敲代码10年，明天退休！

```
System.out.println("点个在看吧！");
console.log("点个在看吧！");
print("点个在看吧！");
printf("点个在看吧！\n");
cout << "点个在看吧！" << endl;
Console.WriteLine("点个在看吧！");
Response.Write("点个在看吧！");
alert("点个在看吧！")
echo "点个在看吧！"
```

点击阅读原文，输入关键词，即可搜索您想要的 CSDN 文章。

你点的每个“在看”，我都认真当成了喜欢