代码重构的实战经验和那些坑 - 文章

2012年冬，我在一家创业公司的小团队里搞软件开发。彼时我们有一位真实的企业客户，且软件的第一版也已发布。开发按进度完工，在发布时我欣喜若狂，也非常骄傲，看着系统服务于每天几百万的独立用户，并发送出数千万条短信真是太令人满意了。到了第二年夏天，公司拿到了真实收入，我的职位变成了开发主管，公司又招了些新人，正待蓬勃发展，一切都很美好。然后我们做了一个巨大的决策失误：决定重写软件——从头开始。

为什么我们觉得有必要从头重写软件呢？

在第一次编写系统代码时，我们的时间表十分紧迫，必须与时间赛跑，在计划时间内赶完进度。因此无论是设计讨论，还是审查会议都没花太长时间——我们没有时间浪费在这上面——只能匆匆完成一个功能、快速测试，然后赶着去做下一个。我们与别的公司共享办公空间，我还记得其他公司的软件开发都会花很长时间做设计、讨论架构，再花上数周讨论设计模型。

除了设计仓促，原本的系统写得不差，总体来说架构也不错。其中有些意大利面条式的代码，是公司之前做概念验证时留下的，因为这些代码能用，再加上工期紧张，当时我们没有去碰。但后来我们不考虑执行优化改进，却决定要从头重写代码的原因在于：

- 老旧代码很糟糕，很难维护；
- “单一整体式的java架构”对我们的未来发展不利，无法支持有6千万移动用户以及多站点部署的大型运行商；
- 我想要尝试炫酷的新技术，比如Apache Cassandra、虚拟化技术、二进制协议、SOA等等。

结果很不幸：我们说服了全公司以及董事会，实现了愿望。

代码重写之旅

正式的开发时间是从2012年春天开始的，我们将2013年1月末设定为发布时间。由于计划太过庞大，我们需要更多的人，于是在印度聘请了顾问与几个远程开发者。但是，我们没有充分预期到维护原本系统、进行新的开发工作与理解客户需求这些并行起来的工作量。

还记得我在文章最开始说过，我们有一个真实客户么？这位客户是南美最大的移动运营商之一。在我们开发的系统投入使用后，他们开始对变更和新功能提出要求，因此我们只能继续更新原来的系统。但是，由于这个系统将会被废弃，在更新时我们总有些敷衍了事，尽可能找借口拒绝了客户许多的新功能需求。结果导致了工期拖延，没能在原定的deadline完成进度。事实上，我们的进度拖延了整整8个月。

不过我们还是先说说结果吧：当项目终于完工时，新系统看起来非常棒，满足所有需求。我们做了负载测试，结果显示新系统能很容易地支持超过1亿的用户，配置集中，查看图表的UI工具也很美观，是时候

废弃旧系统，改换新系统了……

但是客户拒绝了升级的请求：原本的系统已经获得了广泛应用，他们的用户已经开始依赖旧系统了，他们完全不想冒风险。长话短说，浪费了几个月之后我们收效甚微。该项目正式宣告失败。

何时需要重写代码

Joel Spolsky强烈反对重写代码，他建议大家都不要这样做。不过我不是特别认同：有时候逐步优化与重构非常困难，唯一读懂代码的方式就是重写。此外软件开发人员喜欢编写代码，创造新东西——阅读别人写的代码，尝试理解他们的代码与“思维抽象”会很无聊。不过，优秀的程序员也是优秀的维护者。

如果你想要重写代码，一定要出于正确的理由，并有着合适的计划。比如：

有时候在发布新版很久之后，老旧代码仍需维护，维护两个版本的代码需要耗费大量工作，在开始重写前请根据项目规模评估所需的时间与资源。

想想其他失去的机会，并比较任务的优先级。

重写大型系统比小型系统风险更高，考虑一下能否逐步重写。我们同时执行了以下几项工作：切换到新的数据库、使用“SOA”架构、更换为二进制协议，其实本可以逐步执行这些更换。

考虑开发者的偏见。在开发者想要学习新技术或新语言的时候，他们会想要使用这些来重写某些代码。不过我不反对这样做，这也是良好环境与文化的标志，但应当将它与风险和机遇做比较。

Michael Meadows对何时有需要进行“大型”重写有着很好的看法：技术上

- 组件的耦合度很高，无法单独对某个组件进行修改。重新设计单个组件会导致一连串的变化，不仅会影响到相邻的组件，甚至间接影响到所有的组件。
- 技术堆栈太过复杂，未来状态设计需要变更很多的基础架构。出于这个原因执行完全重写十分必要，逐步重新设计在这种情况下没有优势。
- 重新设计单个组件无论如何都会导致对该组件的重写，在现有设计中没有可以插入新功能的地方。这种情况下逐步重新设计没有优势。

政策上

- 赞助商无法理解逐步重新设计需要对项目进行长期投入。不可避免的是：大多数公司对于在逐步重新设计上继续耗费预算没有兴趣。在完全重写代码时，这种现象也很难避免，但赞助商更愿意继续投入，因为他们不想用着半成品的新系统与部分过时的旧系统。
- 系统用户更习惯使用“原本的界面”：在这种情况下，政策上不会允许修改系统的重要部分（前端）。但如果完全从头开始重写，则会绕过这个问题。用户还会坚持使用“相同的界面”，但这次你反击的理由更为充足。要记得：逐步重新设计的总成本总是要高于完整重写代码，但一般来说对企业的影响更小一些。在我看来，如果重写理由充足，公司又有超级优秀的开发者，那么就开工吧。

放弃正在开发的项目很危险：浪费大量的时间和金钱重复实现已有功能，同时还会放弃实现新功能的机会，有可能激怒客户并导致工作计划推迟。如果你正在重写代码，那是你的权力，不过请确保这么做的理由正确，同时了解风险也做了相关计划。