

程序员，为什么千万不要重写代码？

程序员都有一颗工程师的心，所以当他们到一片新的场地想做的第一件事就是，将旧的一切推倒重来。是的，他们决不会满足于简单的增量劳动。

或许这种微妙的心理定位可以解释：为什么程序员进入新项目组后宁愿丢掉旧代码重新写，也不愿意修修补补。他们认为旧代码简直一团糟。

但是，事实上真是这样吗？你之所以认为旧代码一团糟，其实是由编程的一个基本定律决定的，那就是：写代码容易，读代码难。

为什么你觉得旧代码异常混乱？因为读代码更难。

这大概就是代码Reuse难以实现的原因。这就是你组里的每个人都喜欢用不同的功能将分割的字符串转换成一个数组。比起猜测旧的功能是怎样实现的，重新写一个自己的功能要简单和有趣多了。

作为这个公理的推论，你可以问问身边的程序员他们正在奋战的代码怎么样？「简直是一塌糊涂！」他们肯定会这样说。「我简直想推倒重来！」

为什么认为代码这么糟糕呢？「额，看看这个功能，竟然有两页长！完全不知道这些东西为什么在这里！完全不知道这些API是干什么的。」他们会这样回答你。

曾经，Borland的创始人 Philippe Kahn 当初就是向记者们吹嘘：Quattro Pro会比Microsoft Excel要好用得多，因为它是从头开始编写的，全部都是新的源代码！

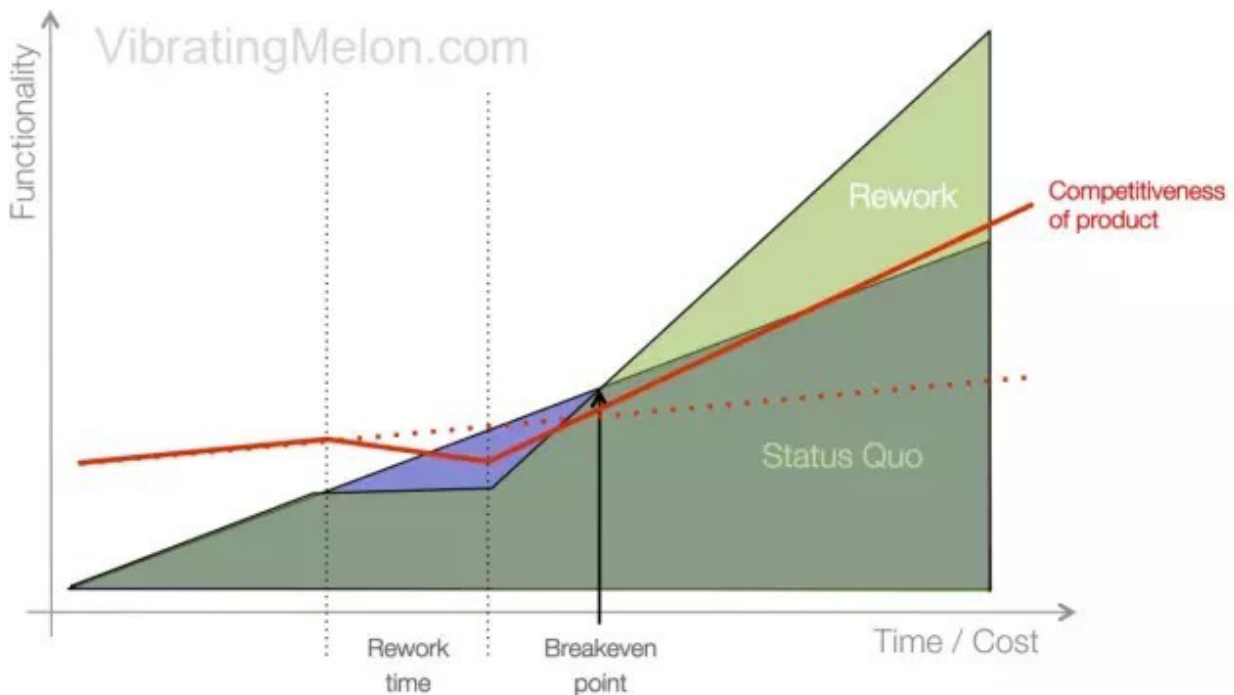
但是，认为新代码比旧代码好简直就是荒谬。旧代码是已经运行过的，测试过的。无数的bug在被发现前都上线运行过，发现之后程序员们可能在花了好些日子才修复了这些bug。这种修复可能是一行代码，也可能是几个字符，无数的时间和精力都花在了这些bug修复上。

当你决定抛弃这些旧代码从零开始的时候，你也丢掉全部前任努力的结果。

新代码一定比旧代码好？NO，重写可能会带来更大的风险。

对技术领导者来说，重写项目的代码也是一个异常艰难的决定。因为从公司层面说，重现代码甚至会威胁产品的市场竞争力。一旦决定重写代码，那么与竞品相比，你可能落后了2~3年——在软件行业，这时间可够长的。

你理想中的新代码会带来产品功能的提升▼

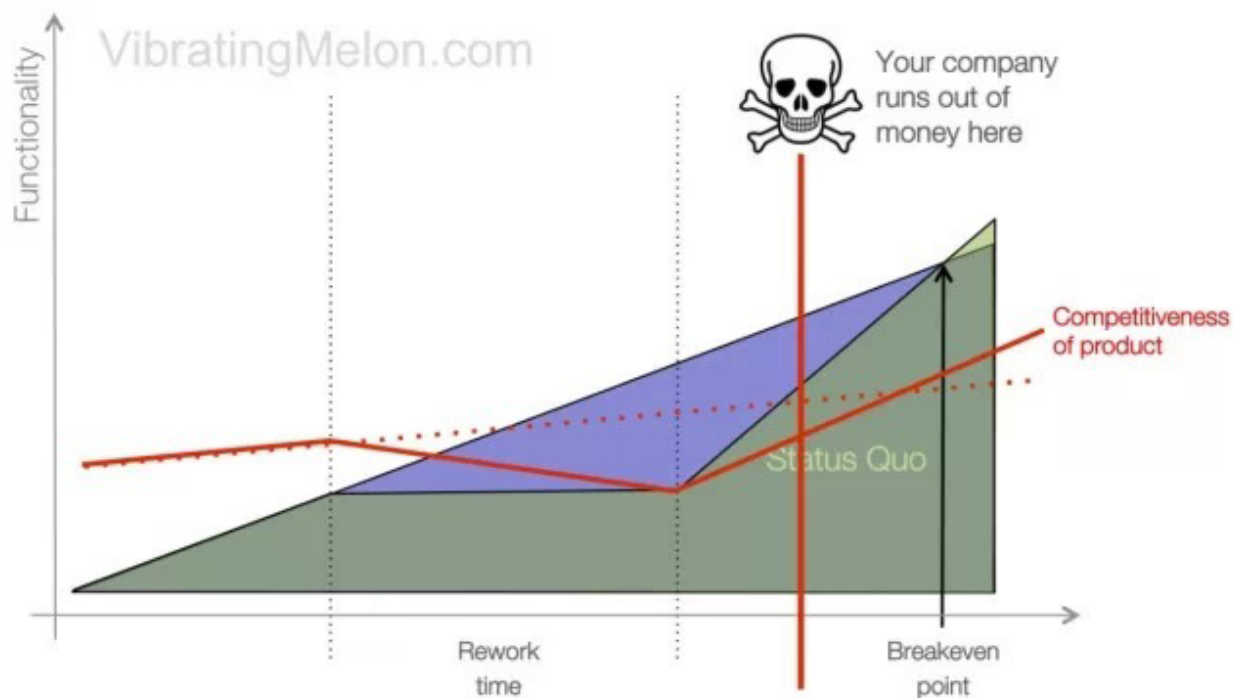


但事实上，即便重写的新代码可以实现旧代码的所有功能和需求，但是为产品带来的市场竞争力只有边际提升。因为重写用的新

技术、新语言、新框架并没有给产品带来质的飞跃。

更不用说在重写的漫长过程中可能会遇到一些意外情况，比如：

1、缺钱：资金链的断裂



2、缺人：核心程序员离职

最终导致效果不佳：达不到原产品应有的所有功能和需求，白白浪费了时间和金钱，也丢掉了市场竞争力。



所以重写代码意味着，你在把自己置身于非常危险的境地，可能几年后你也写不出比以前更好的代码。你只是花了一大笔钱把已经存在的代码又写了一遍。

当你觉得眼前的旧代码很烂时，该怎么办？

你觉得旧代码写的很烂，那又怎样呢？它们已经上线，已经在实际运行中经受住了考验。所以当你发现前任留下的代码乱七八糟的时候，不妨冷静下来，从以下三个方面入手理解代码、改善代码：

1、代码的结构有问题

如果一段网络代码突然弹出了自己的对话框，应该是UI代码需要被处理。这些问题可以被解决掉，你要一次次小心地移动代码，重构，改变接口。还需要一位细心的工程师立马仔细地检查这些改变是否有问题，从而不打扰到其他人。事实上，甚至比较大的结构变化也可以不扔掉代码来完成。

大牛程序员Joel Spolsky回忆说，曾经在某个项目中，他和他的团队花了好几个月重新架构在一点上：把代码动来动去、清理、创建有意义的基类，并创建了模块之间的完美接口。但是他们始终非

常小心翼翼，并没有产生新的bug、也没有丢掉任何旧代码。

2、代码的效率不高

曾经，Netscape的渲染代码被传非常缓慢。但事实上，这只会影响该项目的一小部分，这部分是你可以优化甚至重写的。你完全不必重写全部代码。优化速度的1%工作量，会让你获得99%的爆炸性提高。

3、代码写得很丑

有些代码真的写的很丑，比如Joel曾参与一个项目，开始用下划线做开始的成员变量约定，但后来改用更标准的「M」。所以一半的功能用「_」开始，一半用「M」开始，这看起来真的很丑陋。但这个问题5分钟就能解决，而不用从头开始写全部的代码。

最后，你要记住，从头开始再写一遍并不意味着你会写出比以前更好的代码。因为你没有参与到上一个版本的创建，所以你其实根本就不算有经验。一旦你准备推倒重写，你可能会再犯一遍版本一犯过的错，甚至会产生更多的新问题。

面对糟糕的旧代码，Keep Calm & Carry On!

在大型商业项目中，推倒重来是非常危险的行为。当然，如果你是在做实验，想到新算法可以随时重写。

如果你跳槽、或刚接手一个新项目，面对看上去异常混乱的旧代码，请冷静下来，忍住推倒重写的冲动，想想上面这些经验之谈。

参考资料：

Joel Spolsky www.joelonsoftware.com

Jeremy Burton vibratingmelon.com