其他人的BUG

在软件行业，经常看到有的公司管理让一个人修补另一个人代码里的BUG。有时候有人写了一段代码，扔出来不管了，然后公司管理让其他工程师来修复它。我想告诉你们，这种方法会很失败。

首先，让一个人修复另一个人的BUG，是不尊重工程师个人技术的表现。久而久之会降低工程师的工作积极性，以至于失去有价值的员工。代码是人用心写出来的作品，就像艺术家的作品一样，它的质量牵挂着一个人的人格和尊严。如果一个人A写了代码，自己都不想修复里面的BUG，那说明A自己都认为他自己的代码是垃圾，不可救药。如果让另一个人B来修复A代码里的BUG，就相当于是让B来收拾其他人丢下的垃圾。可想而知，B在公司的眼里是什么样的地位，受到什么样的尊重。

其次，让一个人修复另一个人的BUG，是效率非常低下的作法。每个人都有自己写代码的风格和技巧，代码里面包含了一个人的思维方式。人很难不经解释理解别人的思想，所以不管这两人的编程技术高下，都会比较难理解。不能理解别人的代码，不能说明这人编程技术的任何方面。所以让一个人修补另一个人的BUG，无论这人技术多么高明，都会导致效率低下。有时候技术越是高的人，修补别人的BUG效率越是低，因为这人根本就写不出来如此糟糕的代码，所以他无法理解，觉得还不如推翻重写一遍。

当我在大学里做程序设计课程助教的时候，我发现如果学生的代码出了问题，你基本是没法简单的帮他们修复的。我的水平显然比学生的高出许多，然而我却经常根本看不懂，也不想看他们的代码，更不要说修复里面的BUG。就像上面提到的，有些人自己根本不知道自己在写什么，做出一堆垃圾来。看这样的代码跟吃屎的感觉差不多。对于这样的代码，你只能跟他们说这是不正确的。至于为什么不正确，你只能让他们自己去改，或者建议他们推翻重写。也许你能指出大致的方向和思路，然而深入到具体的细节却是不可能的，而且不应该是你的职责。这就是我的教授告诉我的做法：如果代码不能运行，直接打一个叉，不用解释，不用推敲，等他们自己把程序改好，或者实在没办法，来office hours找你，向你解释他们的思想。

如果你明白我在说什么，从今天起就对自己的代码负起责任来，不要再让其它人修补自己的BUG，不要再修补其他人的BUG。如果有人离开公司，必须要有人修补他遗留下来的BUG，那么说话应该特别特别的小心。你必须指出需要他帮忙的特殊原因，强调这件事本来不是他的错，本来是不应该他来做的，但是有人走了，没有办法，并且诚恳的为此类事情的发生表示歉意。只有这样，程序员才会心甘情愿的在这种特殊关头，修补另外一个人的BUG。