

# 杀死一个程序员只需要改三次需求

许滨麟

1.在软件开发过程中，越早发现问题更容易使项目走上正确的轨道。而在软件开发中，编码属于较后的部分，需求的确定属于较前的部分。这也恰恰证明了编码更改的不容易性。小的需求变动可能只需要改几行代码，但大的需求变动可能需要重新调整系统架构乃至完全重新开始。从某种程度足以让程序员崩溃，所以应该在需求分析阶段就应正确确定真正的需求。对于撸起袖子就开干的方式，看似简单高效，便于直接沟通，能够快速迭代。却不知，发现没有一份正规且实时更新的功能需求设计文档，会付出三四倍的代价来弥补。

2.然而需求的传递总是会伴随着偏差。有一个关于撕纸的实验按照下面的步骤进行操作：把纸对折，再对折把左上角撕下来，旋转180度，把右上角撕下来，把纸展开，完成。你的作品是什么样子？中间开洞了吗？边上呢？角上呢？如果再做一次，你能完成同样的作品吗？然而往往结果不同对于有不同习惯的人而言，有的人会沿长边对折，有的人会沿短边对折，在对折时，有的人会把纸旋转90度，有的人不会，在旋转180度时，有的人会沿中心旋转，有的人会把纸翻面。由于每次对折都会可能产生两种不同结果，在撕第一个角时纸的朝向有四种可能性，旋转180度时有两种可能。所以仅仅两个撕角的位置，就至少有  $2 \times 2 \times 4 \times 2 = 32$  种不同的可能性。如果把所有细节都完整地记录下来的话，需求应该是这样的：拿着纸的短边，沿长边对折，再次拿着短边，沿长边对折，让折过两次的角朝向右下方，没有折过的角在左上方（必要时翻面），把左上角撕下来大约一公分的扇形沿中心旋转180度（不要翻面），使撕过的角处于右下方把右上角撕下来大约一公分的扇形，把纸展开，完成同样信息传递的过程中很有可能会漏掉一些自己认为应该是常识，无需再进一步说明的内容。比如把一张纸对折，我们很容易想当然地认为，应该是沿着长边对折，但事实上并非所有人都是这么理解对折的。

3.当然作为程序员也要学会拒绝不合理的需求。