

需求变更的范畴与形状，是决定对应软件变更实施成本高低的关键。这就是为什么有的代码变更的成本与其实现的功能改变不成比例。这也是为什么第二年的研发成本比第一年的高很多，第三年又比第二年更高。

从系统相关方（Stakeholder）的角度来看，他们所提出的一系列的变更需求的范畴都是类似的，因此成本也应该是固定的。但是从研发者角度来看，系统用户持续不断的变更需求就像是要求他们不停地用一堆不同形状的拼图块，拼成一个新的形状。整个拼图的过程越来越困难，因为现有系统的形状永远和需求的形状不一致。

我们在这里使用了“形状”这个词，这可能不是该词的标准用法，但是其寓意应该很明确。毕竟，软件工程师们经常会觉得自己的工作就是把方螺丝拧到圆螺丝孔里面。

问题的实际根源当然就是系统的架构设计。如果系统的架构设计偏向某种特定的“形状”，那么新的变更就会越来越难以实施。所以，好的系统架构设计应该尽可能做到与“形状”无关。

哪个价值维度更重要

那么，究竟是系统行为更重要，还是系统架构的灵活性更重要？哪个价值更大？系统正常工作更重要，还是系统易于修改更重要？

如果这个问题由业务部门来回答，他们通常认为系统正常工作很重要。系统开发人员常常也就跟随采取了这种态度。但是这种态度是错误的。下面我就用简单的逻辑推导来证明这个态度的错误性。

- 如果某程序可以正常工作，但是无法修改，那么当需求变更的时候它就不再能够正常工作了，我们也无法通过修改让它能继续正常工作。因此，这个程序的价值将成为 0。
- 如果某程序目前无法正常工作，但是我们可以很容易地修改它，那么将它改好，并且随着需求变化不停地修改它，都应该是很容易的事。因此，这个程序会持续产生价值。