# 《人月神话——没有银弹》读后感

在通读了软件工程的圣经——《人月神话——没有银弹》之后，我对软件工程生产力和我在这门专业上应该着重做的事上有了深刻的理解，在之前知识点堆砌的乱七八糟的脑海里建立了一点大局感，应该适当的改变一些着重点，重视本质性的工作，而非附属性的工作；学习开发自己的思想，如何从抽象性的问题，发展出具体概念上的解决方案。在之前的学习中，这些东西并没有得到很深的锻炼，需要更加锻炼自己解决问题的能力。

### 在该论述当中，讨论到了次要和必要复杂度的差异。所谓次要复杂度是指由人们本身所产生的问题，而这类型的问题是可以被解决的。譬如说，撰写和最佳化组合语言的复杂度就是属于次要的，它可以借由高阶程序语言如Java来取代。必要复杂度则是从软件本身要解决的问题衍生而来，并无法被移除。虽说“工欲善其事，必先利其器”，但重点还是在“事”上，除非副属性工作能占到90%。过去我们通过一些方法突破解决了附属性的困难，如高级语言、分时技术、统一的开发环境。但本质性的困难往往没有银弹，我们更需要脚踏实地，日积月累的去克服改善这些问题。

我认为，在现有的所有体系中，都没有所谓的“银弹”，“银弹”只是人们想拥有一个一劳永逸的解决办法而针对一个具体事件想出来的临时的可行的某一个措施，它的效用时间是有限的，并且解决方法本身并不是一成不变的，而是随着时间与经历的增长在变化的。我们不能将问题的处理定格于某一时刻或者某一阶段或者某一特定问题的问题处理。软件工程也是变化的，就好像当我们拥有了瀑布模型外，发现它并不能满足我们一些特定的需求，又出现了增量模型、螺旋迭代模型、敏捷开发等新的模型，且还在持续的更新中。从这个角度来看，我们永远没有办法采用一个已知的问题解决模式处理最新的问题。这个就如同我国在发展社会主义的时候不能照搬苏联模式，只能自己去根据国家实情去探索新的道路，新的发展方法。我们可以在软件生产率上取得逐步的发展，而不是等待不大可能来到的突破。

最后关于Fred Brooks本人，他看待问题的方式角度，和对待评价者的取其精华弃其糟粕的风度，坚持自己的态度，也是我们值得学习的。