# 没有银弹——软件工程中的根本和次要问题

（阐释复杂性是软件工程的自身属性）

我们看看近十年来的情况,没有银弹的踪迹。没有任何技术或管理上的进展, 能够独立地许诺在生产率、可靠性或简洁性上取得数量级的提高

根本困难:复杂度、一致性、可变性和 不可见性。解决次要困难的突破：高级语言，分时，统一编程环境

可能的终极解决方案：Ada和其他高级编程语言，面向对象编程，人工智能，专家系统，“自动编程”，图形化编程，程序验证，环境和工具

解决的终极方案：购买和自行开发，需求精炼和快速原型，增量开发，卓越的设计人员

# 再论《没有银弹》 (“No Silver Bullet” Refired)

（反驳上一章中没有银弹的这一观点）

复杂性是层次化的。

Jone的观点——质量带来生产率

所以,无论采用对象类库还是函数库的方式,我们必须面对我们编程 词汇规模彻底扩大的事实。对于重用,词汇学习并不是思维障碍中的一小部分。

子弹的本质——复杂性是我们这个行业的属性

# 《人月神话》的观点:是或非?(Propositions of the Mythical Man-Month: True or False?)

（主要谈及20年后《人月神话》中需要补充和需要修补的地方）

20 年后的人月神话(The Mythical Man-Month after 20 Years)

概念完整性。

结构师。

将体系结构和设计实现、物理实现相分离。

体系结构的递归。

开发第二个系统所引起的后果:盲目的功能和频率猜测

瀑布模型是错误的!

增量开发模型更佳——渐进地精化

增量式开发和快速原型