

# 人月神话读书笔记

## 推荐理由

《人月神话》，作为一本由被誉为“IBM System/360 之父”的布鲁克斯（Frederick P. Brooks）撰写的软件工程领域的经典著作，可以说从大一开始就被列入推荐书单。大一大二时看这本书时，只是为了应付任务草草了事，这次终于借由作业从头到尾完整地又看了一遍，结合实践经历果然与以前看时感觉完全不同。

全书一共分 19 个章节，看的时候总觉得知识点颇为零碎，CMMI 模型提出的项目、工程、支持过程三个维度都有所覆盖。用大量软件工程实践的事例和通俗的类比来探讨阐述了软件工程管理相关的一些观点。书中从开发人员的职业乐趣开始，通过巴比伦塔讲沟通的重要性，借鉴外科手术队伍来阐述团队的组建和分工等等。从最后的没有银弹，再次肯定了开发工作是一种高智力的脑力工作。其中提及的有些问题或者矛盾随着技术不断革新已经不再成为项目的瓶颈，例如削足适履中阐述的项目投资与磁盘空间和内存之间的矛盾，对于当今电脑硬件的发展水平已经不再成为主要关注点了。还有一些观点可能还没有遇到实际相关的问题，没有很强烈的认同感，或许再经历几年的磨砺再回头看看这本书会有不一样的感悟吧。

所以，下文我会详细谈一谈印象比较深刻或者比较有共鸣的几个观点或者问题，可能大一大二看的时候还不以为然，但随着接触的越来越多，很多以前认为是浪费时间并且怨声载道的步骤其实都是至关重要的，比如各种文档、计划、测试等等。

## 本书要点

### 焦油坑：

编程系统产品（Programming Systems Product）开发的工作量是供个人使用的、独立开发的构件程序的九倍。我估计软件构件产品化引起了 3 倍工作量，将软件构件整合成完整系统所需要的设计、集成和测试又强加了 3 倍的工作量，这些高成本的构件在根本上是相互独立的。

### 乐趣：

创建事物的快乐 开发对其他人有用的东西的乐趣 将可以活动、相互啮合的零部件组装类似迷宫的东西，这个过程所体现出令人神魂颠倒的魅力。

### 乐观主义：

所有的编程人员都是乐观主义者：“一切都将运作良好”。

### 人月：

向进度落后的项目中增加人手，只会使进度更加落后。向软件项目中增派人手从三个方面增加了项目必要的总体工作量：任务重新分配本身和所造成的工作中断；培训新人员；额外的相互沟通。

## 外科手术队伍：

同样有两年经验而且在受到同样的培训的情况下，优秀的专业程序员的工作效率是较差程序员的十倍。小型、精干队伍是最好的——尽可能的少。一位首席程序员、类似于外科手术队伍的团队架构提供了一种方法——既能获得由少数头脑产生的产品完整性，又能得到多位协助人员的总体生产率，还彻底地减少了沟通的工作量。

## 没有银弹：

为什么“没有银弹”，作者认为主要原因在于软件的特性：复杂性、一致性、可变性、和不可见性。笔者认为最根本的原因还是变化。正式可变性造成了软件工程的复杂性，非一致性，和不可见性。没有“银弹”，也就意味着没有一个软件是完美无缺的。这里引用软件测试领域的依据名言：我们只能通过测试证明软件存在 bug，却永远无法通过测试证明软件没有 bug。IT 领域也是优胜劣汰的，一个软件要想生存下来，就必须不停地修改和维护。

## 精编书摘

- 学习编程的最困难部分，是将做事的方式往追求完美的方向调整。
- 第二个谬误的思考方式是在估计和进度安排中使用的工作量单位：人月。成本的确随开发产品的人数和时间的不同，有着很大的变化，进度却不是如此。因此我认为用人月作为衡量一项工作的规模是一个危险和带有欺骗性的神话。它暗示着人员数量和时间是可以相互替换的。
- 无论多少个母亲，孕育一个生命都需要十个月。
- 向进度落后的项目中增加人手，只会使进度更加落后。（Adding manpower to a late software project makes it later）
- 实践是最好的老师，但是，如果不能从中学习，再多的实践也没有用。
- 唯一不变的就是变化本身。
- 适当使用可视化图形可以给工程师和程序员带来可观的成效。而且，这种效果并不仅仅局限于次要问题，开发人员思考和探索的质量也得到了改进。未来的成功系统的开发将围绕在可视化图形表达方式的周围。首先，我们会使用“合适的”实体和关系来形成概念，然后表达成一系列逐步完善的模型，不断地系统化阐明和精化设计概念。模型用若干可视化语言的适当组合来描述，它必须是多种语言的组合，因为系统模型具有若干方面的内容，每方面象变戏法般产生不同类型的思维图像。
- 解决软件构建根本困难的最佳方法是不进行任何开发。软件包只是达到上述目标的方法之一，另外的方法是程序重用。实际上，类的容易重用和通过继承方便地定制是面向对象技术最吸引人的地方。
- 体系结构的递归。对于大型系统，即使所有实现方面的内容都被分离出去，一个人也无法完成所有的体系结构工作。所以，有必要由一位主结构师把系统分解成子系统，系统边界应该划分在使子系统间接口最小化和最容易严格定义的地方。每个部分拥有自己的结构师，他必须就体系结构向主结构师汇报。显然，这个过程可以根据需要重复递归地进行。
- 软件工程的焦油坑在将来很长一段时间内会继续地使人们举步维艰，无法自拔。软件系统可能是人类创造中最错综复杂的事物，只能期待人们在力所能及的或者刚刚超越力所能及的范围内进行探索和尝试。这个复杂的行业需要：进行持续的发展；学习使用更大的要素

来开发；新工具的最佳使用；经论证的管理方法的最佳应用；良好判断的自由发挥；以及能够使我们认识到自己不足和容易犯错的——上帝所赐予的谦卑。

## 作者简介

**Frederick P. Brooks, Jr.**是北卡罗来纳大学 Kenan-Flagler 商学院的计算机科学教授，北卡罗来纳大学位于美国北卡罗来纳州的查布尔希尔。**Brooks** 被认为是“IBM 360”系统之父”，他担任了 360 系统的项目经理，以及 360 操作系统项目设计阶段的经理。凭借在上述项目的杰出贡献，他、**Bob Evans** 和 **Erich Bloch** 在 1985 年荣获了美国国家技术奖（**National Medal of Technology**）。早期，**Brooks** 曾担任 IBM Stretch 和 Harvest 计算机的体系结构师。在查布尔希尔，**Brooks** 博士创立了计算机科学系，并在 1964 至 1984 年期间担任主席。他曾任职于美国国家科技局和国防科学技术委员会。**Brooks** 目前的教学和研究方向是计算机体系结构、分子模型绘图和虚拟环境。