**《人月神话》摘要：【焦油坑】~【画蛇添足】**

**by全柯（斜体字为我自己的理解）**

**一、焦油坑**

（指：以前开发大型项目，会出现各种问题，使得项目像是动物在焦油坑中挣扎一样）

**1、编程系统产品**

4个概念：

——程序（成本x）：可以有作者在开发用的平台上运行，通常车库产品（*a+b problem*）。

——编程产品（成本3x）：可以任何人运行、测试、修复、扩展，可以多平台运行的程序，经过了测试，有文档。

——编程系统（成本3x）：大系统中一个组件单元，强调接口并且保证输入输出与接口一致。

——编程系统产品（成本9x）：真正有用的东西，开发目标。

**2、职业的乐趣**

创造的快感、对他人有用、组装零件精妙、学习的乐趣、易于驾驭。

**3、职业的苦恼**

必须追求完美、目标和资源由他人设定、枯燥debug、差错越多查出新错越难（二次收敛）、竞争激烈。

**二、人月神话**

（讲的是时间进度管理）

**1、乐观主义**

程序员普遍会假设“一些都将运作良好”，“在理应的时间内可以完成”，然而实际上这不一定，而且大项目中“顺利”的概率很小。

**2、人月**

不应用人月作为单位，人数和时间不可以兑换，因为人员之间需要相互交流，并且测试有次序无法并行。各部分关系复杂则人越多需要的时间越多。

**3、系统测试**

时间规划：1/3计划1/6编码1/4构件测试1/4系统测试。有半数时间测试

**4、空泛的估算**

**5、重复产生的进度灾难**

如果已经落后于进度了，再增加人手，还会更加落后（Brooks法则）。应重新安排进度。

**三、外科手术队伍**

（用外科手术团队来类比开发团队，从而清晰划分职责）

**1、问题**

一方面，最好程序员比最差程序员效率高10倍，而部分之间沟通成本巨大还容易出问题，所以最好由少数精英开发。

另一方面，大型项目如果只由少数人开发则需要很多年，过时了。

**2、Mills的建议**

大型项目每个部分由一个类似“外科手术”的团队解决，期中一个人设计分解问题，其他多数人做支持工作。外科医生（首席程序员，定义功能性能设计程序）、副手（后备外科医生经验较少，参与思考讨论评估）、管理员（管人事法务）、编辑（根据外科医生的设计草稿写文档维护文档）、两个秘书（帮管理员和编辑）、程序职员（维护管理文件和技术日志）、工具维护人员（维护编辑调试工具可靠）、测试人员（规划测试步骤，搭建测试平台）、语言专家（专门研究技术）

**3、如何运作**

上述团队中1~2个人思考系统，7个人解决问题，从而概念一致。外科医生&副手掌握所有部分代码，出现意见分歧时由外科医生协调一致。

4、团队的扩建

大型项目中保持相同比例，即团队中<1/7的人负责当外科医生。

**四、贵族专制、民主政治和系统设计**

（讲概念的完整性）

**1、概念一致性**

（欧洲建筑类比）多人分配任务使概念差异、不一致。作者认为概念完整是最重要的。

**2、获得概念的完整性**

人们普遍强度功能，忽略理解复杂度。易用性需要设计的一致性和概念上的完整性。

**3、贵族专制统治和民主政治**

概念完整性要求极少互有默契人员（系统结构师——贵族）实现。不过其他人也可以通过编制外部技术说明来发挥创意。

**4、在等待时，实现人员应该做什么？**

Blaauw提出的创造性活动三个阶段：体系结构、设计实现、物理实现。实际中大多可以并发进行。

五、画蛇添足

（）

**1、结构师的交互准则和机制**

结构师负责估计预算。预算超的话需要削减设计or建议低成本实现方法，相当于挑战开发人员。结构师需要：1只能建议不能支配2能提出建议也能接受别的方法3建议时低调平静4准备放弃提过的建议。

2、自律——开发的第二个系统所带来的后果

结构师在第一个项目谨慎，很多想法打算以后实现，所以第二个项目容易画蛇添足很多修饰，而再以后就经验丰富了。第二次开发存在对于技术细化的趋势，不同于功能增强。

所以结构师需要有意识自我约束，并且听从经验更加丰富结构师的决定。