

outline

- 软件
- 软件危机
- 软件工程

案例分析1： IBM360



- IBM360的操作系统：1963~1966年，5000人一年的工作量，近100万行源程序。
- 项目负责人F. D. Brooks：“正像一只逃亡的野兽落到泥潭中做垂死的挣扎，越是挣扎，陷得越深，最后无法逃脱灭顶的灾难。……程序设计工作正像这样一个泥潭，……一批批程序员被迫在泥潭中拼命挣扎，……谁也没有料到问题竟会陷入这样的困境……”。

人月神话：“没有一种单纯的技术或管理上的进步，能够独立地承诺在10年内大幅度地提高软件的生产率、可靠性和简洁性”。

1、软件活动包含根本任务和次要任务

根本任务：打造构成抽象软件实体的复杂概念结构；

次要任务：使用编程语言表达这些抽象实体。

2、现有解决方案致力于解决次要任务

3、结论：没有银弹

无论这些方案多么完善，都不可能在根本上解决问题。



案例分析2: Ariane 5

- June 4, 1996
- 欧洲宇航局，阿丽亚娜5型火箭
- 首航飞行了大约40秒后开始偏离航向
- 火箭通过远程控制被销毁，5亿美元
- 嵌入式软件，惯性参照系统（SRI）
- 异常被监测到了，但并没有被适当处理
- numeric exception (integer overflow)



软件危机的产生



上世纪六十年代
(1960年代),
软件开发和维护
遇到了一系列严
重的问题

- 需求不明确
- 软件存在大量缺陷
- 软件开发成本和项目周期失控
- 软件难以维护
-

**软件
危机**

软件危机的概念

- 软件危机是指在计算机软件的开发和维护过程中所遇到的一系列严重问题。
- 这些问题绝不仅仅是不能正常运行的软件才具有的，实际上，几乎所有软件都不同程度地存在这些问题。

Key points:

- how to develop new software
- how to support old software