



软件危机的产生

授课教师：蓝天 电子邮箱：lantian1029@uestc.edu.cn

什么是软件危机

定义

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。


效率和质量下降

1968年NATO会议 (Garmisch , Germany) 提出 “软件危机”

- ❖ 项目超出预算
- ❖ 项目超过计划完成时间
- ❖ 软件运行效率很低
- ❖ 软件质量差
- ❖ 软件通常不符合要求
- ❖ 项目难以管理并且代码难以维护
- ❖ 软件不能交付

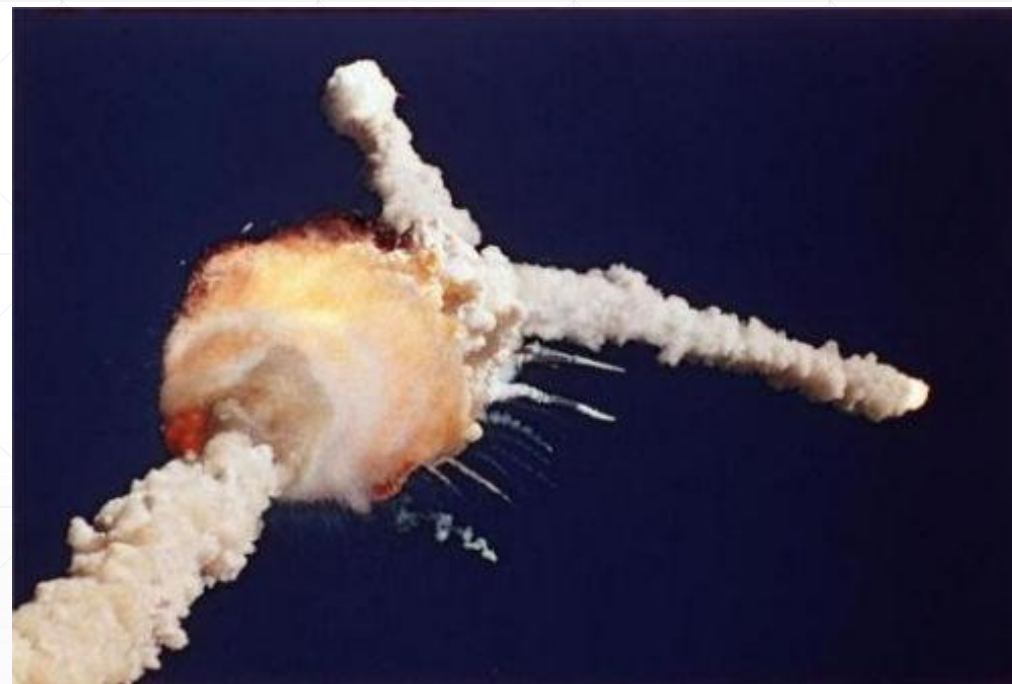
软件危机案例：OS/360

- 1961-1964年，IBM公司的 OS/360，共约100万条指令，花费了5000个人年，开发总投资5亿美元，达到当时美国研究原子弹的曼哈顿计划投资20亿美元的1/4，而结果却令人沮丧，错误多达2000个以上，系统根本无法正常运行。

| OS/360 | |
|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1965 | IBM Corporation, United States |
|  | <p>System/360 software included a new programming language (PL/1), revised versions of older languages (such as FORTRAN, COBOL, and RPG), and a very complex new operating system that would run on all machines in the 360 family. OS/360's complexity was highlighted in project manager Fred Brooks' 1975 classic book <i>The Mythical Man Month</i> as an example of the "Second System Effect." This phenomenon, he explained, leads designers of successful systems to overdesign and unnecessarily embellish their second system.</p> |
| <p>Fred Brooks with several System/360 machines, c. 1965</p> | <p>PConline.com.cn 太平洋电脑网</p> |

软件危机案例：美国火箭爆炸

- 谣言：1963年在美国，由于一个FORTRAN程序的循环语句“DO 5 I=1,3”误写为“DO 5 I=1.3”，“,”被误写为“.”，**一点之差**导致飞往火星的火箭爆炸，造成了1000万美元的损失。
- 实际情况：根据NASA的报告，是1962年的飞往金星的宇宙飞船天线发生故障，致使地面无法控制火箭，于是火箭上的计算机开始控制火箭，然而在**导航系统软件中又存在一个bug**（打印错误，在平滑半径导数表达式“ $R\text{-dot-bar sub } n$ ”中漏掉了代表平滑的“bar”），致使火箭迅速偏离航道。于是发射场的安全官员下令摧毁飞船。

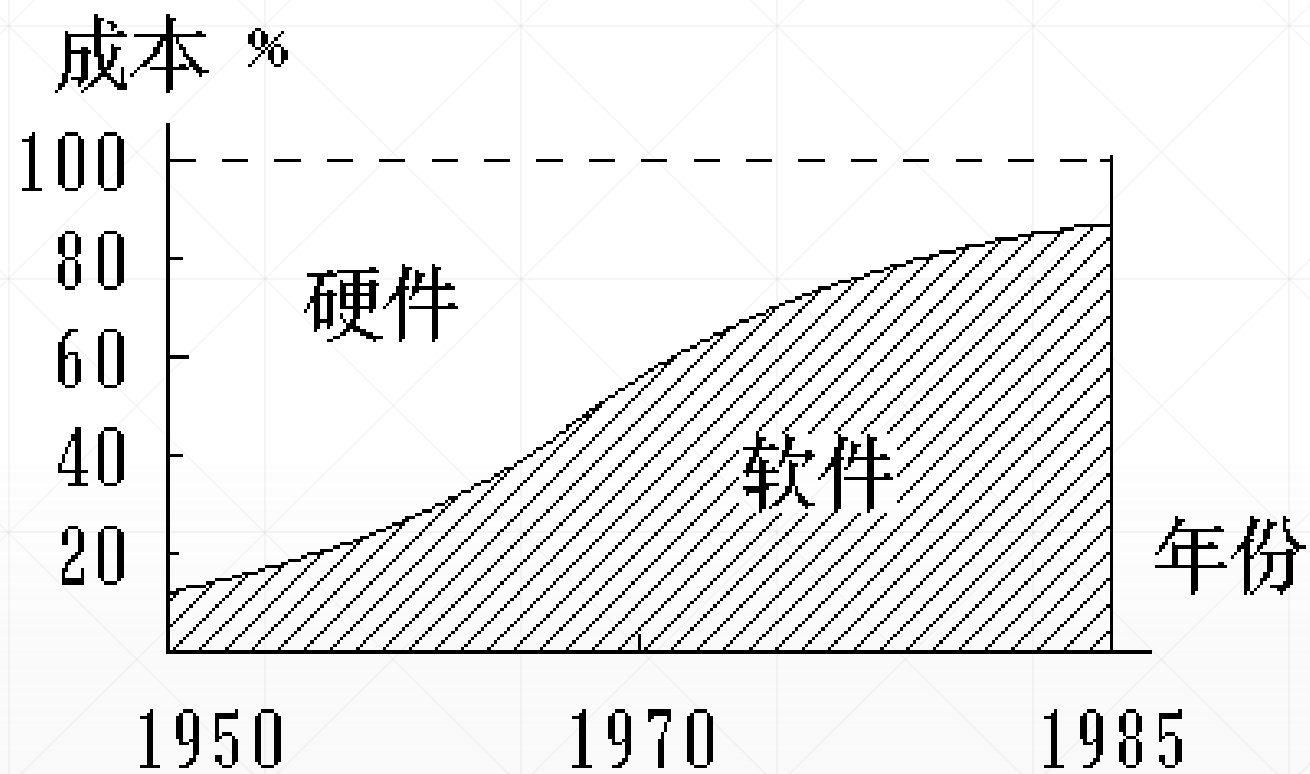


软件危机案例：美国银行信托软件系统

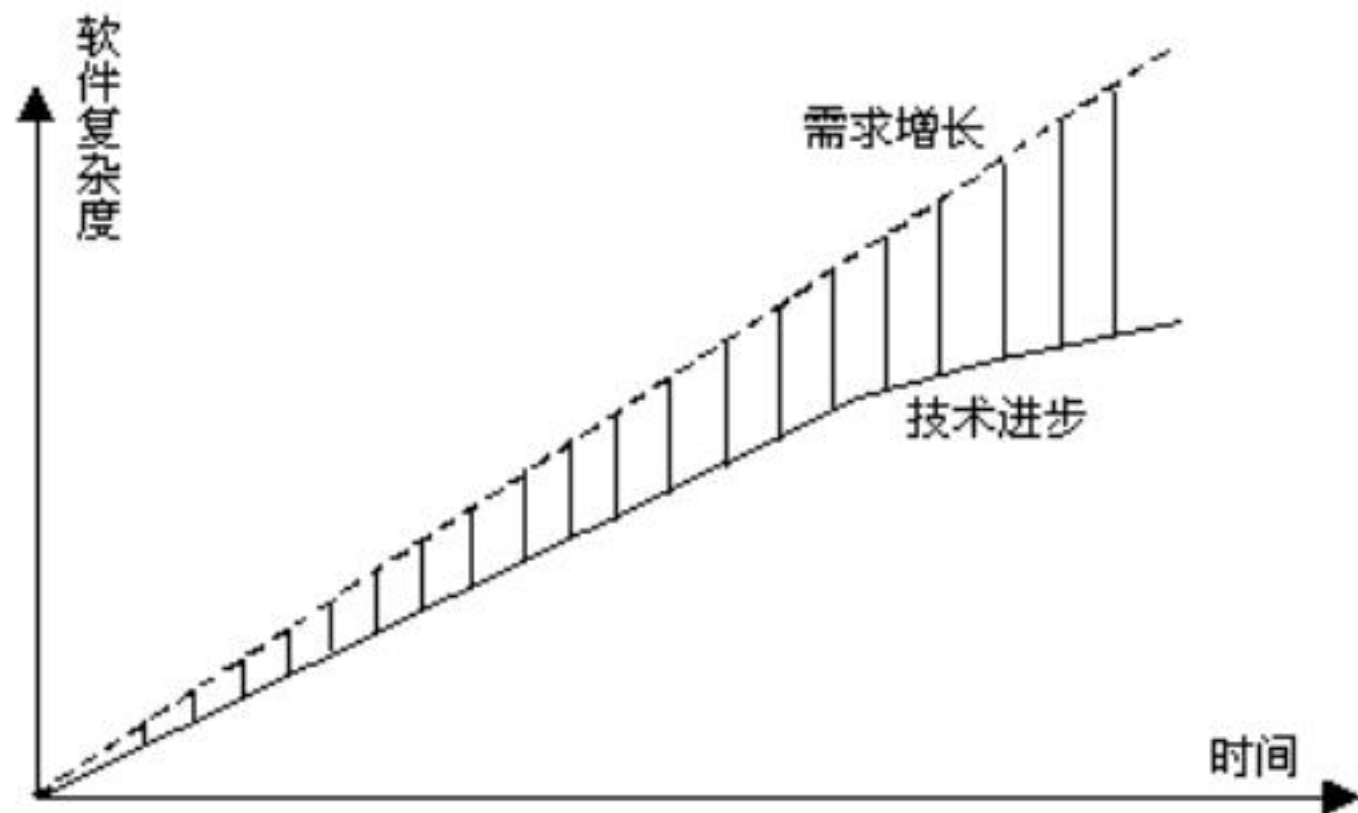
- 美国银行1982年进入信托商业领域，并规划发展信托软件系统。
- 项目原订预算2千万美元，开发时程9个月，预计于1984年12月31日以前完成，后来至1987年3月都未能完成该系统，期间已投入6千万美元。
- 美国银行最终因为此系统不稳定而不得不放弃，并将340亿美元的信托账户转移出去，并失去了6亿美元的信托生意商机。



软件成本日益增加



软件技术进步 < 需求增长



产生软件危机的原因

客观：软件本身特点

- 逻辑部件
- 规模庞大

主观：不正确的开发方法

- 忽视需求分析
- 错误认为：软件开发=程序编写
- 轻视软件维护

消除软件危机的途径：软件工程！

对计算机软件有一个
正确的认识：
软件≠程序



必须充分认识到软件开发不是某种个体劳动的神秘技巧，而应该是一种组织良好、管理严密、各类人员协同配合、共同完成的工程项目。



推广使用在实践中总结出来的开发软件的成功技术和方法。
开发和使用更好的软件工具。

