信息与软件工程学院
School of Information and Software Engineering

高级软件工程

第一章 软件工程的现实与理想

姓名 | 许毅

xuyi0421@uestc.edu.cn

2024/1/2

课程介绍

- 学时 40
- Part 1：软件工程理论与技术 学时约17
- Part 2： IT项目管理实践、案例与新技术 学时约20
- Part 3：软件工程课程答辩 学时3
- 项目要求（40分）（项目经理确定每人贡献度）：课程组老师提供3-4个（偏应用或技术），学生可以直接选择或自拟（自拟题目阶段：要求给出题目和调研情况）。
  - 围绕一个复杂软件工程项目，从软件工程过程进行需求分析、概要设计和详细设计，体现软件工程的体系结构风格和设计模式，体现复杂软件工程的过程管理模型，撰写符合软件过程规范的文档。
  - 5-7人一组，通过开源项目按照软件工程进行实践，题目自拟（考虑到研究兴趣），建议以新的开发理念或方法，团队协作完成。代码+文档（20页以上），说明项目开发过程中的主要原理和过程。
  - 最后一周按照实验报告过程进行答辩。（以小组为单位）
- 综述报告（20分）：
  - 调研前沿领域的软件工程实践现状，学习软件工程PMBOK相关文档，撰写一篇课程报告。（以个人提交，格式参考正式综述论文）
- 期末考试（40分）

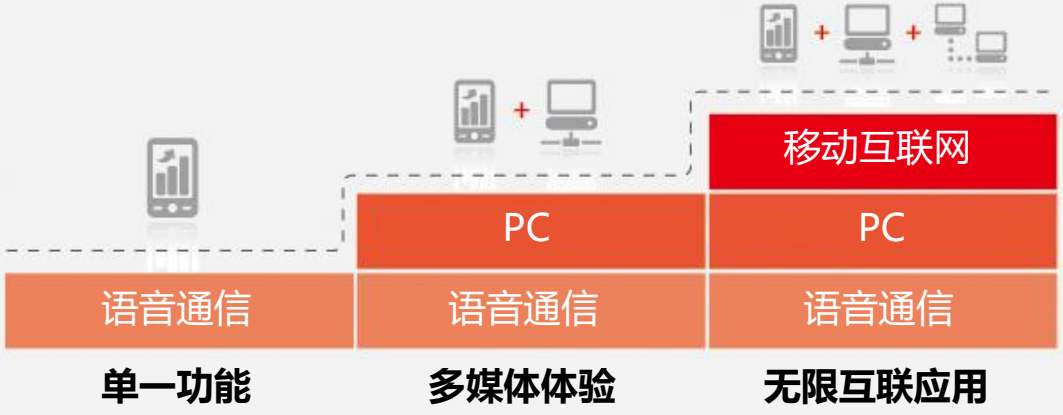
第一章目录

软件应用平台的发展趋势——效率、整合、性能

个人应用平台发展趋势

- 平台开放：丰富的互联网应用
- 性能卓越：听觉、视觉、触觉
- 体验完美：工作、娱乐、生活

从单一通讯终端 → 软硬件一体化应用平台



应用平台发展趋势

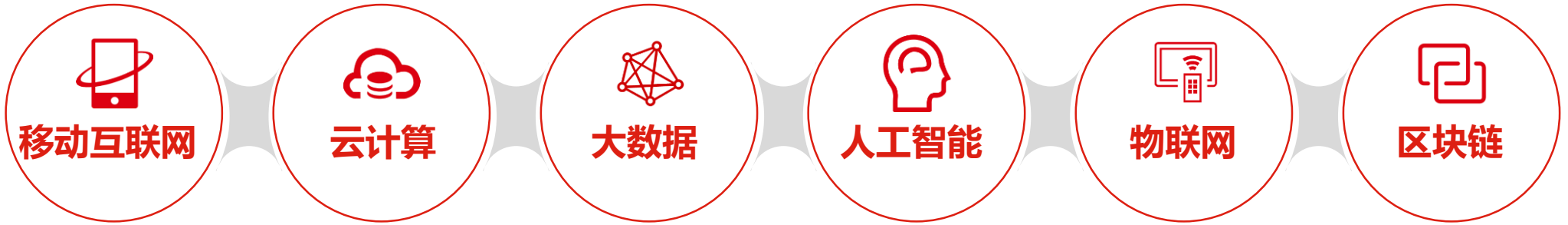
- 业务融合：应用整合、流程整合、信息聚合
- 性能卓越：高并发访问、大数据处理
- 平台开放：跨异构系统、多种开发语言、多应用模式

从单一应用系统 → 企业计算平台

5G时代，加速技术更新，云计算与大数据助力数字中国建设

云计算技术 & 大数据：推动社会进步和发展的两个轮子

新技术让曾经的很多不可能变成了现实！



语音自然语言交互　智能试衣镜　行业知识图谱　财务机器人　刷脸支付　RFID

以图搜图　AVR互动营销/建筑模型　智能形象顾问　HR机器人　无人机快递　商品溯源　智能合约

会员画像　销量预测　自动定价　仓容规划　导购机器人　AVG自动引导运输车

计算模式	架构模式	交互模式	数据模式	流程模式	开发模式

信息化建设紧跟技术换代——智慧应用6大技术模式的变化

数据模式	结构化、非结构化、时序数据、图数据、实时数据、数据湖、区块链/分布式账本
流程模式	流程驱动->数据驱动、 规则驱动->智能驱动、单据驱动->场景&事件驱动
交互模式	极致体验、Web化、移动化(H5/Native/Hybrid)、多端化(语音/视频/传感器)
开发模式	瀑布开发、敏捷开发、DevOps(运维测试智能化)、研发管理->研发运营
架构模式	单体巨石架构、SOA、分布式、微服务、中台架构/混合云架构
计算模式	单一物理机、传统IDC、云计算(虚拟化、容器化(Serverless)、边缘计算/IOT

现代软件项目开发与管理是环环相扣的八大系统工程

1. 现状分析与变革诉求
2. 模式梳理与管理提升
3. 体系架构与基础设计
4. 蓝图设计与信息化愿景
5. 结构与应用功能
6. 平台与基础支撑
7. 策略与落地路径
8. 组织与资源保障

软件的本质特性

构造性

软件 是 典型的知识产品

是 客观世界中 问题空间与解空间的 具体描述

⟹ 软件 是有结构的

⟹ 构造性 是软件的本质特性

传统的软件开发 是 个体作坊式的

主要解决功能问题

较少考虑结构问题

造成 软件 复杂度 高

维护难度 大

软件的本质特性（续）

演化性

软件 是 客观事物的一种反映
是 知识的提炼
知识的体现
知识的固化

客观世界 不断发展
不断发生变化

⟹ 软件系统 不可能一成不变
新需求、新技术不断出现
软件系统要不断升级
不断演化

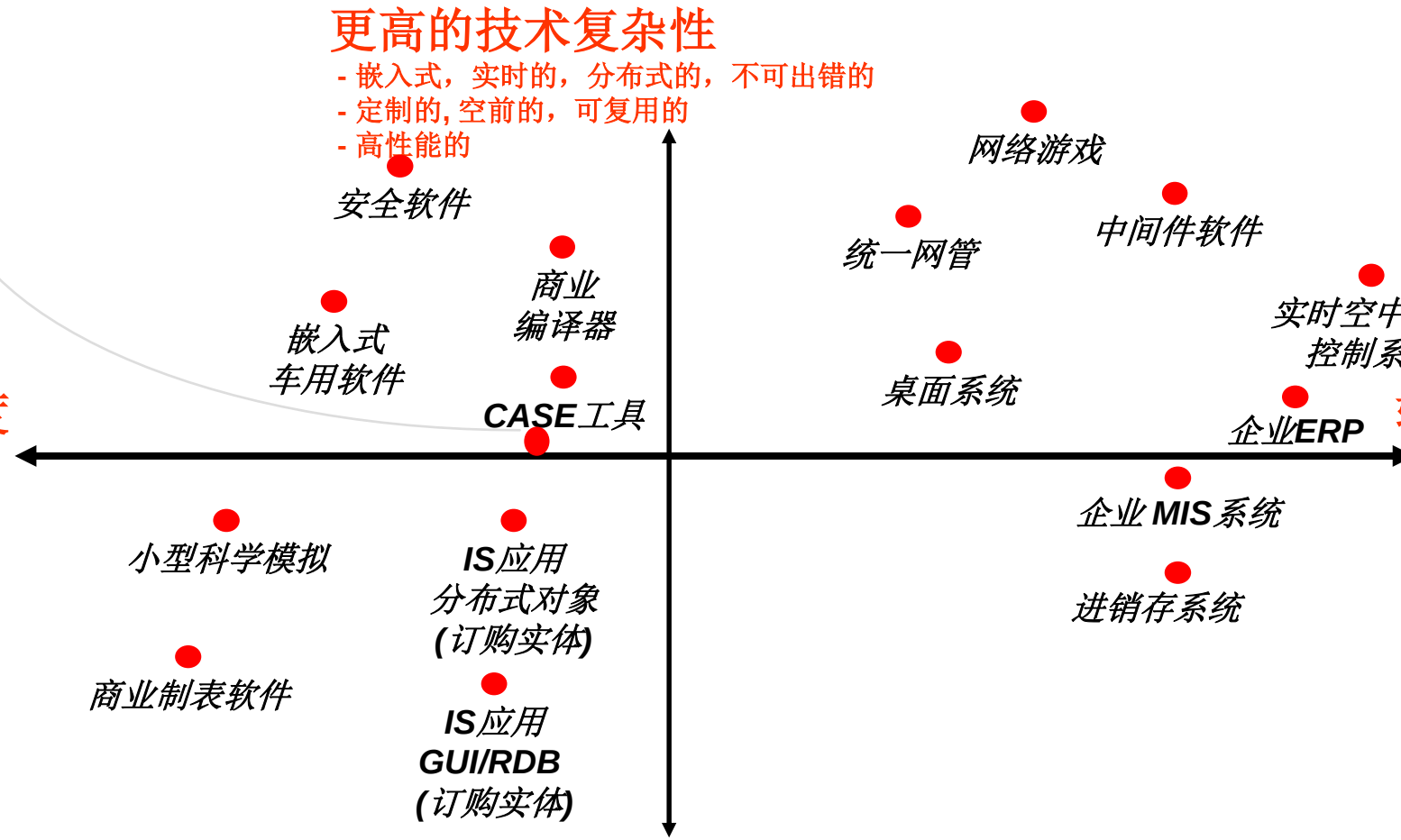
软件技术的总体发展趋势

平台 网络化
方法 对象化
系统 构件化
产品 家族化
开发 工程化
过程 规范化
生产 规模化
竞争 国际化

其中：
系统软件 是 核心
支撑软件 是 手段
应用 是 目标
标准规范 是 基础
安全 是 保障

软件管理复杂度与技术复杂度关系

一个比较中等的项目
- 5-10 人
- 10-15 个月的开发周期
- 3-5 个外部界面
- 一些不可知的事情 & 风险



信息系统项目常见类型分类

- 1 流程型制造（钢铁、化工、医药……）
- 2 离散型制造（汽车、电子电器、家电……）
- 3 项目型制造（建筑、地产、电力…..）
- 4 密集资产运行EAM（轨道交通、高铁、核电…..）
- 5 科研型项目制造（军工、科研院所 …….）
- 6 MIS型信息管理

软件项目管理的关键概念点

项目、产品与平台判断

721原则　8020原则

软件项目合理利润

30%?

软件项目管理体系分类

CMM XP IPD……

软件公司费用构成

人力成本50%　平台管理成本50%

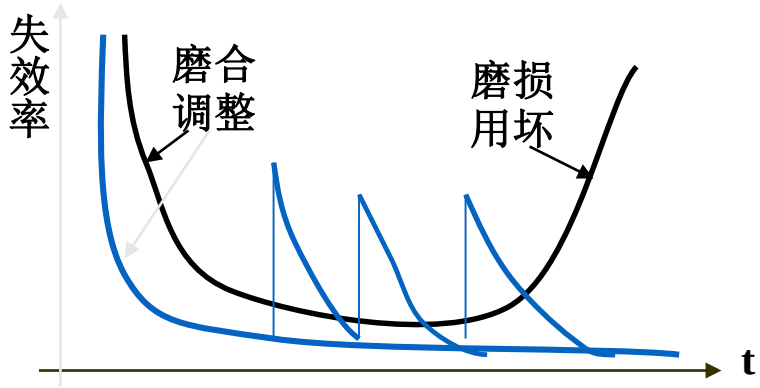
软件项目开发工期预测

人月与人日

软件危机和软件工程

软件危机的表现

- 开发成本难以控制，进度不可预计；
- 软件系统的质量和可靠性很差，难以满意；
- 软件文档相当缺乏，软件系统不可维护；
- 软件开发生产率很低，软件产品供不应求。
- 软件产品成本十分昂贵。



硬、软件产品失效率曲线

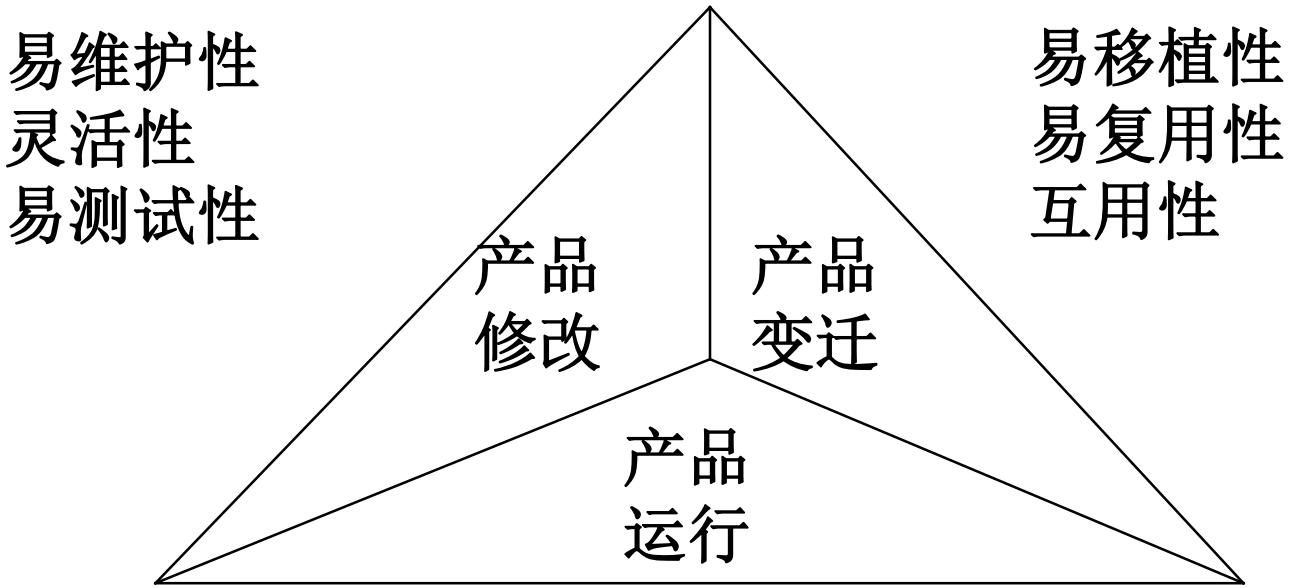
软件开发与项目管理体系的核心

- 软件开发与管理的关注：质量与效率
- 软件基本特点：构造与演化

软件质量定义

明确声明的功能和性能需求、明确文档化的开发标准、以及专业人员开发的软件所应具有的所有隐含特征都得到满足。

McCall模型中的软件质量要素



软件质量保证体系的研究和主要技术

目前国际上软件过程质量管理最主要的三个典型代表：

- CMM
- ISO9000系列
- ISO/IEC15504

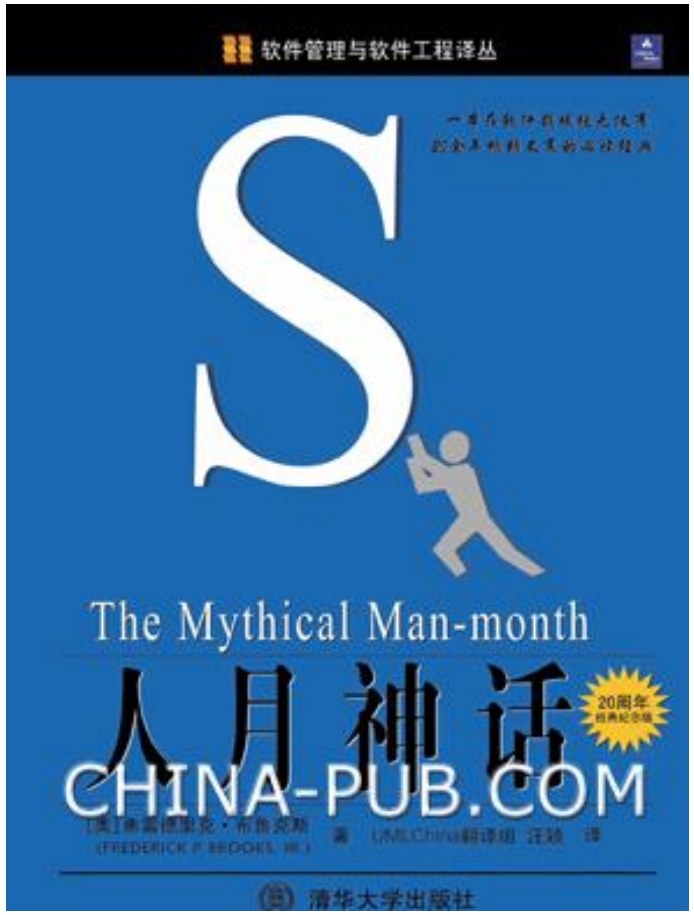
第一章目录

1.2.1

人月神话

相关信息

- **书籍简介**
  - 1975年首次发行
  - 软件工程的经典之作
  - 收录了作者多年的技术文章
  - 为大型的软件开发提供启示



相关信息

- **作者简介**
  - Frederick Brooks
  (弗雷德里克.布鲁克斯)
  1931年4月19日—2022年11月17日
  - 美国工程院院士
  - 曾主持开发IBM/360
  - 1999年的图灵奖获得者

Photo credit: © Jerry Markatos

软件开发中的焦油坑

图片注解：中生代时期拉布雷亚(La Brea Tar Pits)焦油坑复原图

软件开发中的焦油坑

“史前史中，没有别的场景比巨兽在焦油坑中垂死挣扎的场面更令人震撼。

上帝见证着恐龙、剑齿虎在焦油中的挣扎。它们挣扎的越是猛烈，焦油越是缠的越紧，没有任何猛兽足够强壮或具有足够的技巧，能够挣脱束缚，它们最后都沉到了坑底。” ——*Brooks*

- **大型的系统开发类经常遇到焦油坑！**

“表面上看起来好像没有任何一个单独的问题会导致困难，每个都能被解决，但是当它们相互纠缠和积累在一起的时候，团队的行动就会变得越来越慢。对问题的麻烦程度，每个人似乎都会感到惊讶，并且很难看清问题的本质。” ——*Brooks*

软件开发中的焦油坑

- **Windows NT 5.0(即windows 2000)**
- **时间：**
  - 计划开发时间：3年
  - 实际开发时间：5年
- **公布数据：**
  - 程序员人数：5,000人
  - 代码行数：35,000,000 行代码
  - 开发时间：5年
- **每位程序员每年生产多少行代码？**

软件开发中的焦油坑

- **每位程序员每年生产多少行代码?**

  以最不幸的情况来估计，每行代码都需要自己编写，得到结果：35,000,000 行/(5000人*5年) = 1400行/人.年。

  这个效率远远低于一名正常程序员的产出量。

- **两种可能**:
  - (1)微软雇佣了5000名不合格的程序员去开发windows NT 5.0
  - (2)开发一个大规模的程序系统产品远难于堆砌出单一的程序。

焦油坑之一：进度滞后

- **进度滞后的原因**
  - 乐观主义的盛行（软件开发是纯思维性的活动）
  - 人月神话
  - 各项任务的时间安排不当（特别是测试时间）
  - 迫于用户压力制定了不合理的进度计划
  - Brooks法则

焦油坑之一：进度滞后

“使用人月为单位来衡量一份工作的规模是一个危险和具有欺骗性的神话。它暗示着人员数量和时间是可以相互替换的。”——brooks

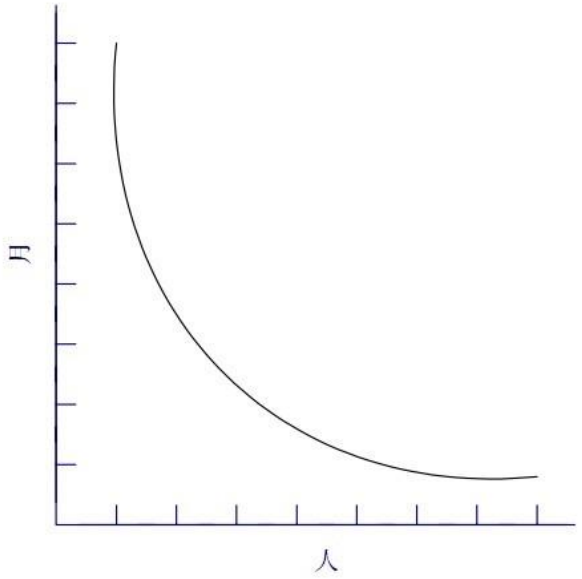
- **人月神话**
  - 人月：参与开发的程序员数目 * 项目持续的月数
  - 为什么说人月是神话?

    （1）许多任务是无法拆解的

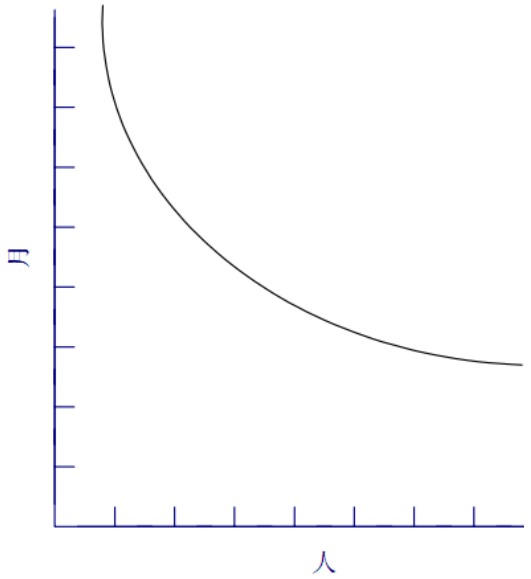
    （2）即使任务可以拆解，人员之间的沟通交流时间随着人手的增加以(n-1)*n/2的规模递增

    20人* 5个月 > 50人* 2个月

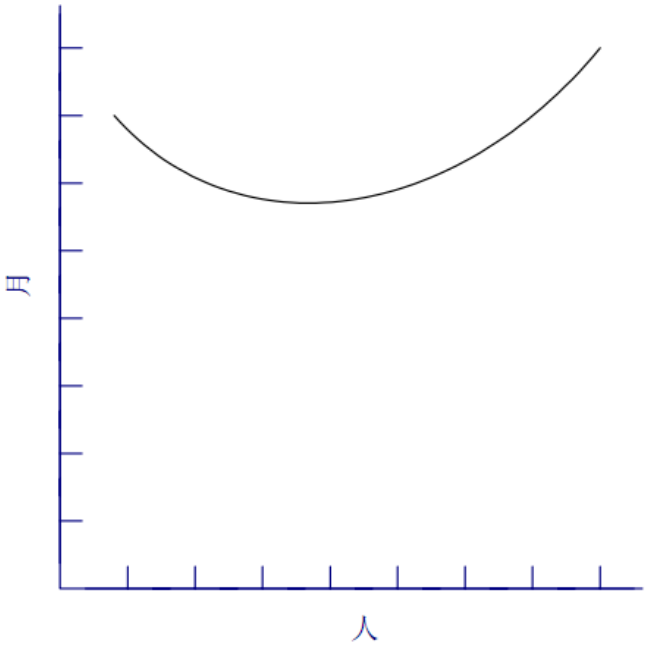
焦油坑之一：进度滞后



图一:
可以完全分解的任务
（在软件开发中不存在)



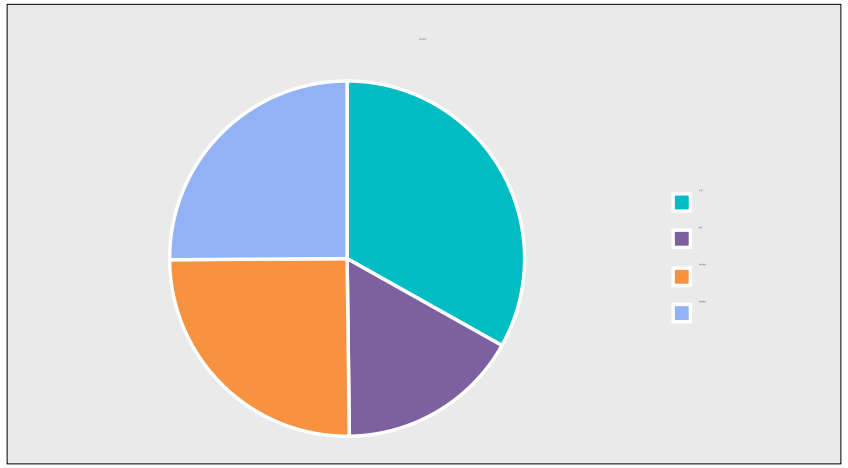
图二:
可以分解但需要沟通
交流的任务
(一般情况)



图三:
关系错综复杂的任务
（极端情况)

焦油坑之一：进度滞后

- 各项任务的时间安排不合理



测试时间不足的恶果：直到项目的发布时期，才有人发现进度的问题。

焦油坑之一：进度滞后

- **Brooks法则**
  - 定义：“*Adding manpower to a late software project makes it later.*” *0--Brooks*
  - 出现原因：

    新增的程序员需要进行培训，同时增加了沟通的成本，使得开发团队增加了更多的开发时间，这个时间超过了新增程序员所做的贡献。
  - 结论过于武断？应该附加的前提：

    (1)项目已经进行了相当长时间的开发

    (2)系统比较复杂，新人的学习成本较高

焦油坑之一：进度滞后

- **对于进度滞后项目的解决方案**
  - (1)在新的进度安排中分配充分的时间，确保工作能彻底、认真地完成
  - (2)当项目延期所导致的后续成本很高时，往往削减系统的功能是唯一的解决方法

对于许多小型的开发团队，加派人手是通常的解决方法?

焦油坑之二--缺乏沟通

- **巴别塔为什么会失败？**

巴别塔：圣经中继“诺亚方舟”后人类第二个大工程，以失败告终

焦油坑之二--缺乏沟通

- **巴别塔为什么会失败?**

……现在整个大地都使用同一种语言。在一次迁徙的过程中，人们发现了苏美尔地区，并且在那里定居下来。他们说：“来，让我们建造一座带有高塔的城市，这个塔将高耸云霄，也让我们声名远扬。”于是上帝决定下来看看人们建造的城市和高塔，看了以后，他说：“他们只是一个种族，使用一种语言，如果他们一开始就能造出城市和高塔，那以后就没有什么能难得倒他们了。来，让我们在人类的语言里制造些混淆，让他们相互之间不能听懂。”上帝于是改变并区别开了人类的语言，巴比伦塔不得不停工了。 --旧约 第11章

巴别：在希伯来语中是"变乱"的意思

焦油坑之二--缺乏沟通

如果大型编程项目中交流不足，很可能面临和巴别塔一样的结局。

- **大型编程项目中的交流方法**
  - (1)成员之间非正式途径的经常性讨论
  - (2)定期召开的项目会议
  - (3)项目工作手册

焦油坑之二--缺乏沟通

- **在树状组织架构中进行交流**

  传统的沟通方式是网状的，n个人之间的交流需要(n-1)*n/2个接口。
  然而团队的组织架构总是树状的，可以利用这种树状的组织结构减
  少沟通成本。需要为每棵子树定义一些基本要素：

  - (1)每棵子树需要完成的任务
  - (2)子树的产品负责人（对外沟通）
  - (3)子树的技术主管（对内技术指导）
  - (4)进度
  - (5)人员的划分
  - (6)子树对外接口的定义

焦油坑之三--文档问题

- **撰写关键的文档**
  - 书面的决策更加精确
  - 文档可作为沟通交流的手段
  - 文档可为系统将来的优化和扩展提供指导
  - 项目经理的文档可作为项目的数据基础和检查表
- **不提倡过度文档**
  - 给用户使用的最终产品并非是文档

焦油坑之三--文档问题

- **需要提供哪些关键的文档?**
  - 做什么：开发目标
  - 怎么做：产品技术说明。以建议书开始，以内部文档和用户手册结束
  - 时间：进度表
  - 资金：预算
  - 地点：工作空间分配
  - 人员：组织图

焦油坑之三--文档问题

- **自文档化的程序**
  - 为什么提倡自文档化程序?

    数据处理的基本原理告诉我们，试图把信息放在不同文件中，并试图维护它们的同步是件费力不讨好的事情。在软件开发里，我们却试图维护一份机器可读的程序，以及一系列包含记述性文字和流程图的文档。
  - 如何产生自文档化的程序?

    (1)在程序源代码里附加尽可能多的信息，例如变量名，函数名等

    (2)尽可能使用空格和一致性的格式来提高程序的可读性，表现从属和嵌套关系

    (3)以段落注释的格式，向程序插入必要的记述性文字

1.2.2

没有银弹

是否存在终极利器？--银弹之争

- **人狼、银弹与软件项目**
  - 人狼：满月时会由人形变成狼形的怪兽。
  - 银弹：唯一可以杀死人狼的武器。
  - 软件项目：类似于人狼，常常看似简单明了的东西，却有可能变成一个落后进度、超出预算、存在大量缺陷的怪物。

是否存在终极利器？--银弹之争

- **没有银弹（“No Silver Bullet”）**
  - *没有任何技术或管理上的进展，能够独立地许诺十年内使生产率、可靠性或简洁性获得数量级上的进步。*--Brooks,1986
  - 原因：由软件工程的内在特性所决定的：复杂度，一致性，可变性和不可见性。
- **存在银弹（"There Is a Silver Bullet"）**
  - *在信息化社会里，市场对信息的巨大需求将成为经济诱因，促使银弹的出现。*--Cox,1990

没有银弹！ --Brooks

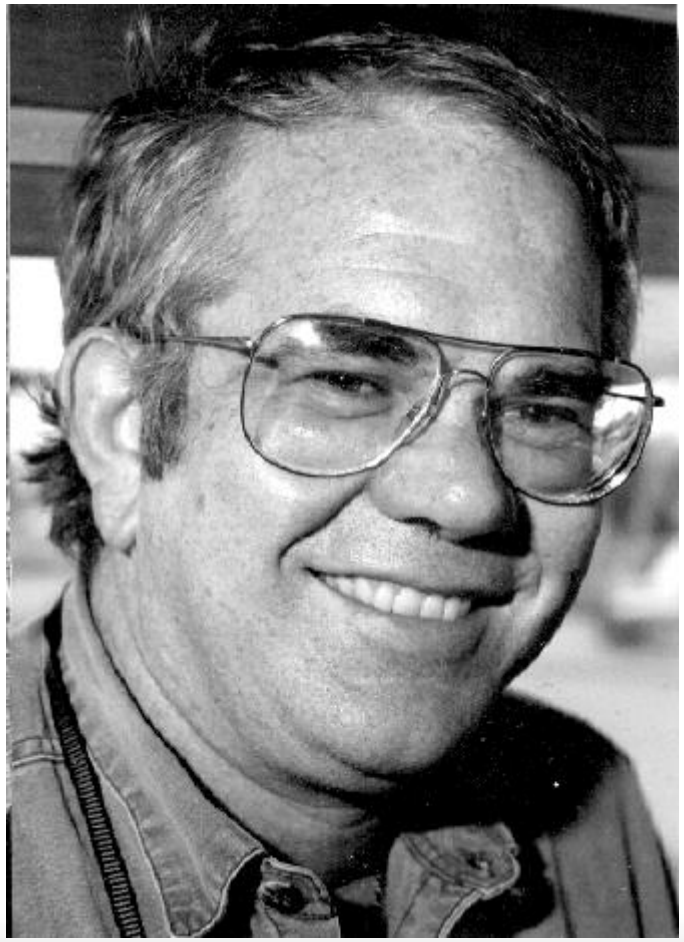
- **软件工程的内在特性**
  - **复杂度**：不同于建筑、汽车等产品，软件实体可能比任何由人类创造的其它实体都要复杂，因为没有任何两个软件部分是相同的（至少是在语句的级别上）。
  - **无规则性**：不同于数学、物理等学科，软件工程所控制的很多复杂度是随心所欲、毫无规则可言的，来自于若干必须遵循的人为惯例和系统。
  - **可变性**：由于软件是纯粹的思维产物，易于修改，用户经常会提出改进要求。
  - **不可见性**：软件是无法可视化的，不仅限制了个人的设计过程，也阻碍了设计人员之间的交流。
- **由于软件工程的内在特性限制了银弹的出现！**

没有银弹！ --Brooks

- 能够提高效率但并非银弹的技术：
  - 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
  - 工作站
  - 购买而非自行开发
  - 增量开发——增长，而非搭建系统
  - 卓越的设计人员

银弹就在这里！ --Cox

- **Cox简介**
  - “以人为本”
  - 计算机学和社会学大师
  - 开发了Objective-C
    （类似于C++，已经消亡)



银弹就在这里！ --Cox

- **Cox眼中的银弹：类似硬件晶片般的软件组件**
  - 通过使用结构化的方法，将软件组件内的复杂结构包装得完美，使得组件简单易用，由这些组件整合而成大型软件，自然简单易用，软件危机于是被化解了。

- **为何现在没有出现银弹？**
  - 一般的工业产品每卖出一件就消耗一份组件，然而软件无论卖出多少件都只需要消耗一份组件。这样使软件组件提供商没有动力去生产出完美的组件。

思考与讨论

人月神话　50岁了

你对焦油坑和银弹如何理解？

你认为会有银弹吗？