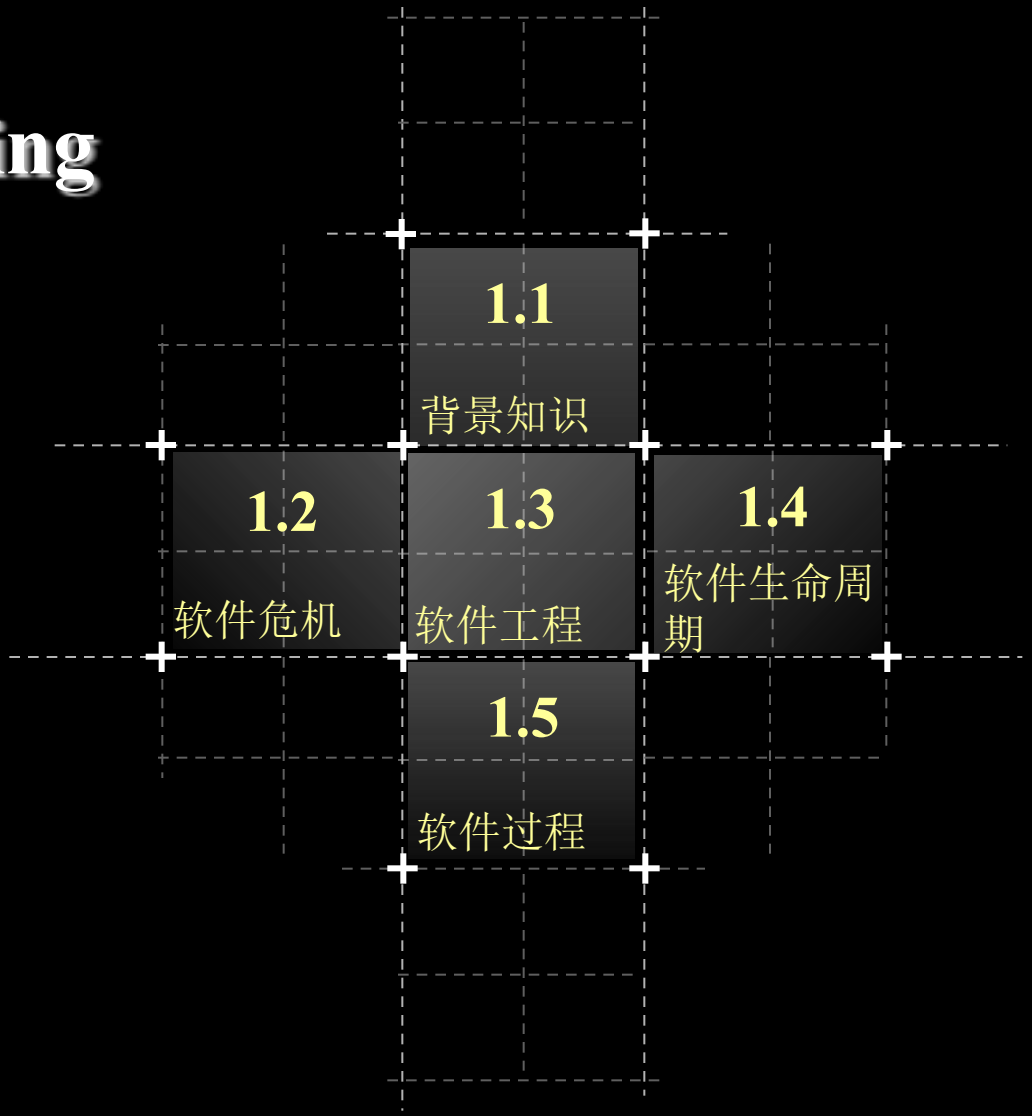
第1章　软件工程概述

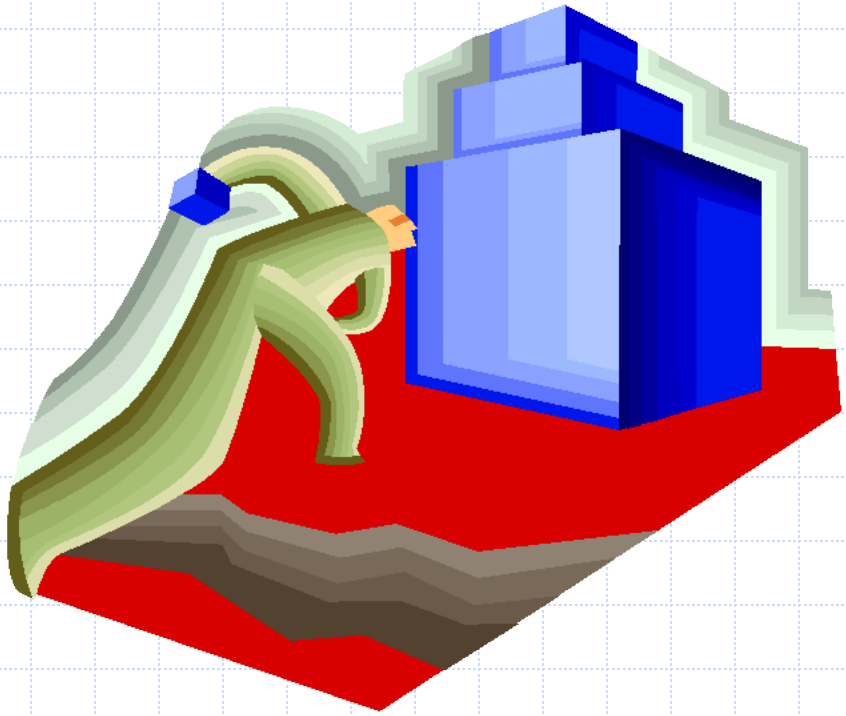
Software Engineering

教学单位

教师介绍



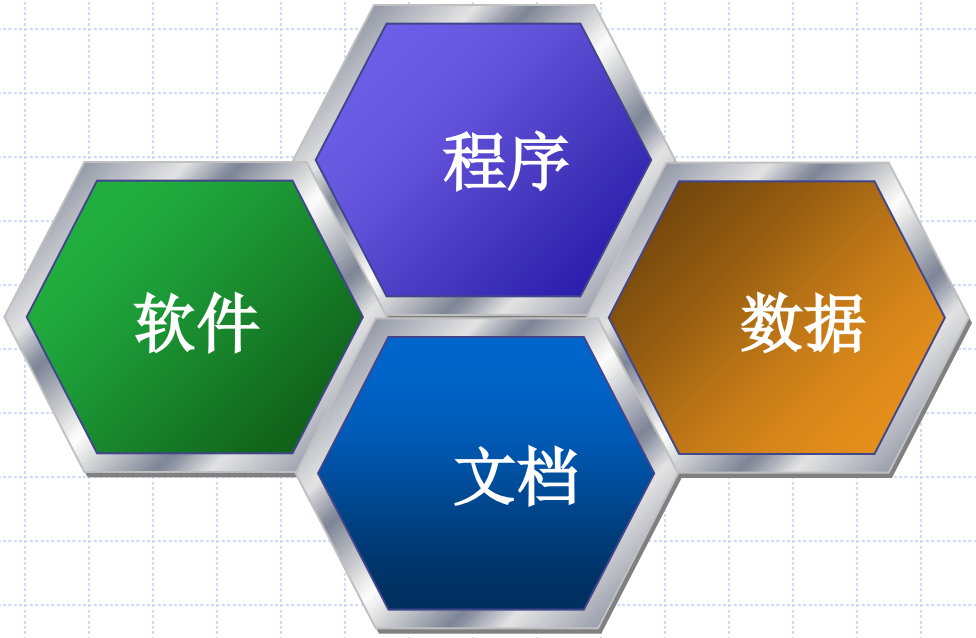
1.1 背景知识



1.1.1 什么是软件

按事先设计的功能和性能要求执行的指令序列(instructions)

计算机系统中与硬件相互依存的另一部分，它是包括程序，数据及其相关文档的完整集合。



使程序能正常操纵信息的数据结构

与程序开发，维护和使用有关的图文材料

1.1.2 软件的分类

按软件的规模进行划分

类别	参加人员数	研制期限	源程序行数
微型	1	1～4周	0.5k
小型	1	1～6月	1k～2k
中型	2～5	1～2年	5k～50k
大型	5～20	2～3年	50k～100k
甚大型	100～1000	4～5年	1M(=1000k)
极大型	2000～5000	5～10年	1M～10M

1.1.2 软件的分类

按软件的功能进行划分

1

系统软件

- 操作系统
- 数据库管理系统
- 设备驱动程序
- 通信处理程序

2

支撑软件

- 文本编辑程序
- 文件格式化程序
- 磁盘向磁带进行数据传输的程序
- 程序库系统

3

应用软件

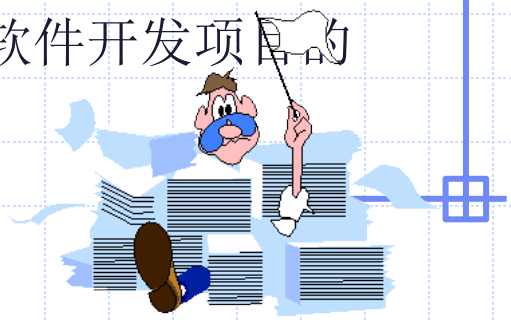
- 商业数据处理软件
- 系统仿真软件
- 计算机辅助设计
- 系统仿真软件等

1.2 软件危机

1 Case：美国IBM公司在1963年至1966年开发的IBM360机的操作系统。这一项目花了5000人一年的工作量，最多时有1000人投入开发工作，写出了近100万行源程序.....据统计，这个操作系统每次发行的新版本都是从前一版本中找出1000个程序错误而修正的结果.....

- 这个项目的负责人F. D. Brooks事后总结了他在组织开发过程中的沉痛教训时说："……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.....”。IBM360操作系统的历史教训成为软件开发项目的典型事例为人们所记取。

Software Crisis !



1.2.1 软件危机的概念

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。

软件危机

如何开发软件，以满足对软件日益增长的需求。

如何维护数量不断膨胀的已有软件。

1.2.1 软件危机的概念

软件危机主要有以下典型表现：

- 对软件开发成本和进度的估计常常很不准确。
- 软件产品的质量往往靠不住。
- 软件常常是不可维护的。
- 软件通常没有适当的文档资料。
- 软件成本在计算机系统总成本中所占的比例逐年上升。

1.2.2 产生软件危机的原因

- 在软件开发和维护的过程中存在这么多严重问题：
  - 与软件本身的特点有关。
  - 与软件开发与维护的方法不正确有关。

软件的特点

- 表现形式
  - 软件是一种逻辑实体，而不是具体的物理实体。因而它具有抽象性
- 生产方式
  - 软件的生产与硬件不同，在它的开发过程中没有明显的制造过程，大多数软件仍是定制的。
- 维护
  - 在软件的运行和使用期间，没有硬件那样的机械磨损，老化问题
- 要求
  - 软件产品不允许误差

软件的特点

- 软件的开发和运行常受到计算机系统的限制，对计算机系统有着不同程度的依赖性
- 软件的开发至今尚未完全摆脱手工艺的开发方式
- 软件本身是复杂的
  - 实际问题的复杂性
  - 程序逻辑结构的复杂性
- 软件成本相当昂贵
- 相当多的软件工作涉及到社会因素

软件开发与维护的方法

相当多的软件专业人员在实践过程中或多或少地采用了错误的方法和技术，这可能是使软件问题发展成软件危机的主要原因。

忽视软件需求分析的重要性。作好软件定义时期的工作，是降低软件成本提高软件质量的关键。

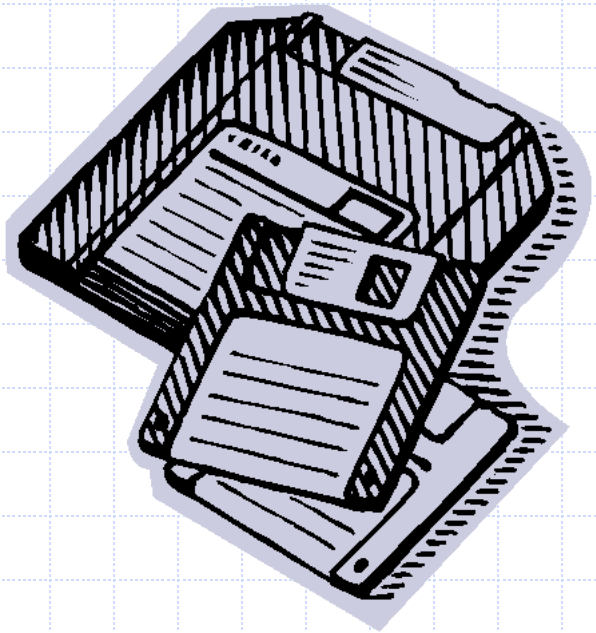
对用户要求没有完整准确的认识就匆忙着手编写程序是许多软件开发工程失败的主要原因之一。必须认识到程序只是完整的软件产品的一个组成部分。必须清除只重视程序而忽视软件配置其余成分的观念。

轻视软件维护等。轻视维护是一个最大的错误。

1.2.3 消除软件危机的途径

- 彻底消除“软件就是程序”的错误观念。
- 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的技术和方法，并且研究探索更好更有效的技术和方法。
- 开发和使用更好的软件工具。

1.3 软件工程



1.3.1 软件工程定义

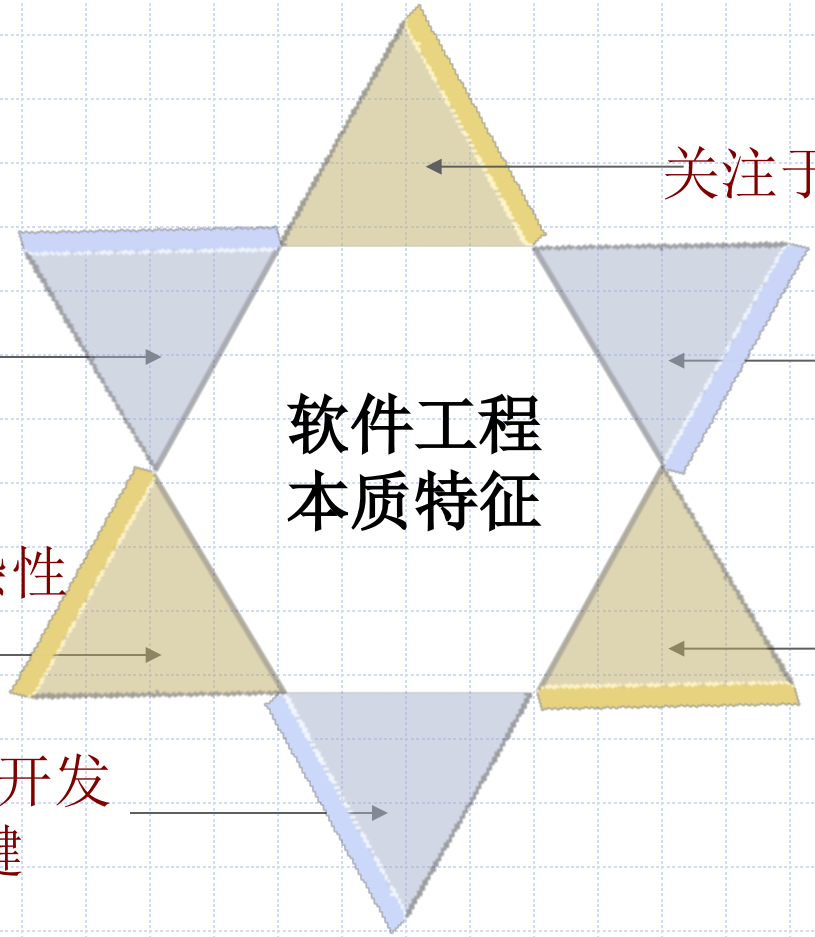
- 软件工程是一类求解软件的工程，它应用计算机科学、数学（用于构造模型和算法）和管理科学（用于计划、资源、质量和成本等的管理）等原理，借鉴传统工程（用于制定规范、设计范型、评估成本、权衡结果）的原则和方法，创建软件以达到提高质量、降低成本的目的。

1.3.1 软件工程定义

- **软件工程是一门交叉学科**
  - 软件开发技术：软件开发方法学
    软件开发过程
    软件工具和软件工程环境
  - 软件工程管理：软件管理学
    软件经济学
    软件心理学

- **软件工程所包含的内容不是一成不变的，随着人们对软件系统的研制开发和生产的理解。应用发展的眼光看待它。**

1.3.1 软件工程定义


关注于大型程序的构造
软件经常变化
软件必须有效地支持它的用户
软件工程本质特征
中心课题是控制复杂性
开发软件的效率非常重要
和谐地合作是开发软件的关键

1.3.2 软件工程原理

(1) 用分阶段的生命周期计划严格管理

♣项目概要计划 ♣里程碑计划 ♣项目控制计划 ♣产品控制计划 ♣验证计划 ♣运行维护计划

(2) 坚持进行阶段评审

(3) 实行严格的产品控制——基准配置管理 (Baseline configuration management)

(4) 采用现代程序设计技术

(5) 结果应能清楚地审查— set standards

(6) 开发小组的成员应该少而精

1+1 < 2

(7) 承认不断改进软件工程实践的必要性

1.3.3 软件工程方法学

- 通常把在软件生命周期全过程中使用的一整套技术方法的集合称为方法学。方法学包含3个要素：方法、工具和过程。目前使用得最广泛的软件工程方法学，分别是**传统方法学**和**面向对象方法学**。

1.3.3 软件工程方法学

1. 传统方法学

- 传统方法学也称为生命周期方法学。它采用**结构化技术**来完成软件开发的各项任务，并使用适当的软件工具来支持结构化技术的运用。这种方法学把软件生命周期的全过程依次划分为若干个阶段，然后顺序地完成每个阶段的任务。开发软件的时候，从对问题的抽象逻辑分析开始，一个阶段一个阶段地进行开发。

1.3.3 软件工程方法学

2. 面向对象方法学

面向对象方法学具有下述4个要点:

- 把对象作为融合了数据及在数据上的操作行为的统一的软件构件。用对象分解取代了传统方法的功能分解。
- 把所有对象都划分成类。每个类都定义了一组数据和一组操作。数据用于表示对象的静态属性，是对象的状态信息，而施加于数据之上的操作用于实现对象的动态行为。

1.3.3 软件工程方法学

- 按照父类与子类的关系，把若干个相关类组成一个层次结构的系统。
- 对象彼此间仅能通过发送消息互相联系。对象的所有私有信息都被封装在该对象内，不能从外界直接访问。

1.4 软件生命周期

- 由**软件定义**、**软件开发**和**运行维护**(也称为软件维护)3个时期组成。
- 软件定义(**系统分析**)时期的任务是：
  - **确定软件开发工程必须完成的总目标；**
  - **确定工程的可行性；**
  - **导出实现工程目标应该采用的策略及系统必须完成的功能；**
  - **估计完成该项工程需要的资源和成本，并且制定工程进度表。**
- 软件定义时期通常划分成3个阶段，即问题定义、可行性研究和需求分析。

1.4 软件生命周期

- 软件开发时期通常由4个阶段组成：总体设计，详细设计，编码和单元测试，综合测试。其中前两个阶段又称为**系统设计**，后两个阶段又称为**系统实现**。
- 维护时期的主要任务是使软件持久地满足用户的需要。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/947045061136006146