

BUPT
TSEG

软件工程 模型与方法

Models & Methods of SE

软件工程概述



- 计算机软件
- 软件的发展和软件危机
- 软件工程
- 软件工程知识体系



- 本节内容
- 软件的定义
- 软件的特点
- 软件的分类



- Hello World
- 纸牌、挖地雷、2048...
- 文本编辑器
 - Notepad/vi...
 - Winword/WPS...
- 图书馆信息管理系统/办公自动化系统...
- 财务软件
 - 金蝶/用友...
- 电信运营支撑系统
 - 计费系统、帐务系统...
 - 呼叫处理/7号信令/网络管理系统...
- 操作系统
 - Dos/Windows/Unix/Linux...
- 数据库管理系统
 - Oracle, DB2, MySQL...



- IEEE定义：软件是计算机程序、规程以及运行计算机系统所需要的文档和数据。
- Wirth中指出：
 - 在结构化程序设计：程序 = 算法 + 数据结构
 - 在软件工程中：软件 = 程序 + 文档。
- 另一种对软件的公认解释是：软件是包括程序、数据及其相关文档的完整集合。
- 程序和数据是构造软件的基础，文档是软件质量的保证，也是保证软件更新及生命周期长短的必需品。



- 软件是一种逻辑实体，具有抽象性。
- 软件开发过程中没有明显的制造过程，应理解为“创作”。
- 不存在机械磨损和老化问题，但存在软件退化问题。
- 软件的开发和运行受到计算机系统的约束和限制。
- 软件开发至今未完全摆脱手工艺的开发方式；
- 软件是复杂的原因：
 - 实际需求（业务背景）的复杂性
 - 程序逻辑的复杂性
- 软件研制成本高，软件成本所占比例逐年增加。
- 软件投入运行时还涉及到许多社会因素。



- 根据软件服务对象的范围不同：
 - 通用软件：操作系统、数据库等；
 - 定制软件：企业ERP、办公自动化系统等；
- 根据软件完成功能所处的层次不同：
 - 应用软件、中间件软件、系统软件
- 系统软件：指能与硬件紧密配合在一起，使计算机系统各个部件、相关的软件和数据协调、高效地工作
 - 操作系统
 - 设备驱动程序
 - 数据库管理系统等



- 软件发展阶段
 - 程序设计阶段：20世纪50至60年代
 - 程序系统阶段：20世纪60至70年代
 - 软件工程阶段：20世纪70至90年代
 - 现代软件工程阶段：20世纪90年代至今
- 软件危机
 - 1960年后至1970之间的软件快速发展阶段
- 软件危机的解决途径
 - 1968年 软件工程大会第一次召开



软件发展的三个主要阶段

阶段 特点	程序设计	程序系统	(现代) 软件工程
软件所指	程序	程序及说明书	程序、文档和数据
程序设计语言	汇编及机器语言	高级语言	软件语言
软件工作范围	程序编写	包括设计和测试	软件生存期
需求者	程序设计本人	少数用户	市场用户
开发软件的组织	个人	开发小组	开发小组及大中型软件开发机构
软件规模	小型	中小型	大中小型
决定质量的因素	个人程序技术	小组技术水平	管理水平
开发技术和手段	子程序/程序库	结构化程序设计 	数据库、开发工具、开发环境、工程化开发方法、标准和规范、网络及分布式开发、面向对象技术、软件复用
维护责任者	程序设计者	开发小组	专职维护人员
硬件特征	价格高/存储容量小 工作可靠性差	降价、速度、容量及工作可靠性明显提高	向超高速、大容量、微型化及网络化发展
软件特征	完全不受重视	软件技术的发展不能满足需求，出现软件危机	开发技术有进步，但未获突破性进展，价高，未摆脱软件危机

- 20世纪60年代后，随着计算机软件应用领域增多，软件规模不断扩大，软件系统功能多，逻辑复杂，不断扩充，从而导致许多系统开发出现了不良的后果：
 - 系统存在大量错误，可用性和可靠性差；
 - 系统无法增加新功能，难于维护；
 - 系统无法按照计划时间完成；
 - ...等因素
 - 导致很多软件系统 的彻底失败。



- 所谓软件危机就是计算机软件在开发和维护过程中所遇到的一系列严重问题，导致了软件行业的信任危机，具体表现在：
 - 软件开发成本难以估算，无法制定合理的开发计划；
 - 用户的需求无法确切表达；
 - 软件质量存在问题；
 - 软件的可维护性差；
 - 缺乏文档资料；

Projects running over-budget

Projects running over-time

Software was very inefficient

Software was of low quality

Software often did not meet requirements

Projects were unmanageable and code difficult to maintain

Software was never delivered



- 产生软件危机的原因：
 - 软件系统本身的复杂性；
 - 软件开发的方法和技术不合理及不成熟；
- 软件工程方法
 - 1968年 Friedrich Ludwig (Fritz) Bauer 提出运用工程化原则和方法，组织软件开发解决软件危机，并提出“软件工程 Software Engineering”的概念。

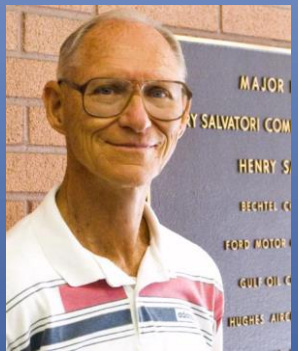


- 软件工程定义
- 软件工程要素
- 软件工程的目标和原则
- 软件工程基本原理





- 1968年10月，Fritz Bauer首次提出了“软件工程”的概念：软件工程是为了经济地获得能够在实际机器上高效运行的可靠软件而建立和使用的一系列好的工程化原则。



- Barry Boehm的软件工程定义：运用现代科学技术知识来设计并构造计算机程序及为开发、运行和维护这些程序所必需的相关文件资料。





- Richard E. Fairley 认为：软件工程学是为在**成本限额以内****按时**完成开发和修改软件产品所需的系统生产和维护的技术和管理的学科。
- IEEE计算机学会将“软件工程”定义为：(1)应用系统化的、**规范化的、定量的方法**来开发、运行和维护软件，即：**将工程应用到软件**；(2)对(1)中各种方法的研究。



- 软件工程三要素：方法、工具和过程。
 - 方法：提供了“如何做”的技术
 - 工具：提供了自动或半自动的软件支撑环境
 - 过程：将软件工程的方法和工具综合起来以达到合理、及时地进行计算机软件开发的目的



- 软件工程的目标可概括为：生产具有正确性、可用性以及开销适宜的软件产品。
- 软件工程的最终目的是摆脱手工生产软件的状况，逐步实现软件研制和维护的自动化。

填空题 | 判断题



- 软件 engineering 研究的内容:

- 软件开发技术

- 软件开发方法学
 - 开发过程模型
 - 开发工具
 - 软件 engineering 环境

- 软件 engineering 管理

- 软件管理学
 - 软件 engineering 经济学
 - 软件心理学等内容

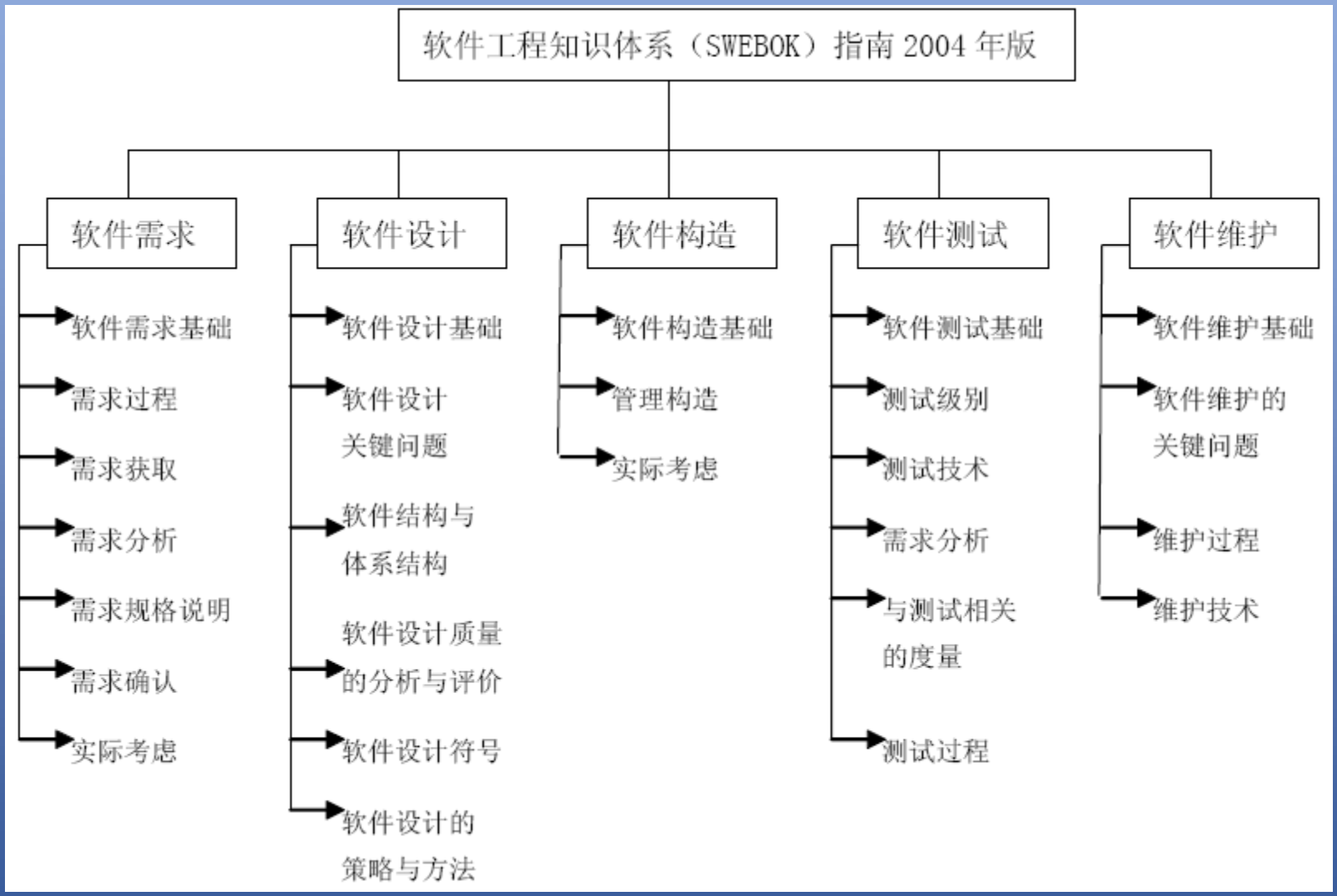


- 软件工程知识体系指南简介
- 软件工程知识体系知识域

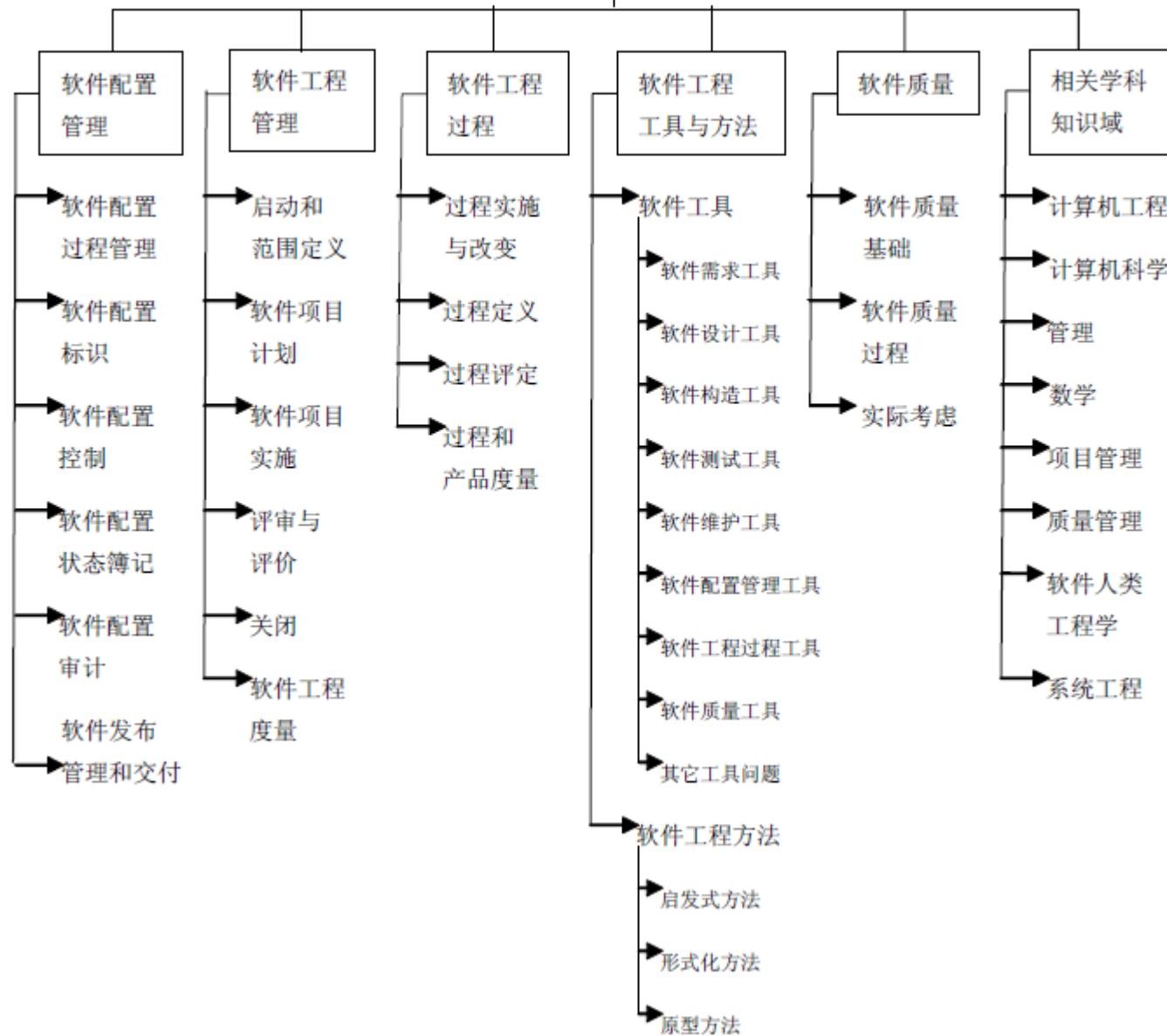


- SWEBOK是IEEE计算机学会的职业实践委员会 (Professional Practices Committee) 主持的一个项目，目的：
 - 促进世界范围内对软件工程的一致观点；
 - 阐明软件工程相对其它学科（如计算机科学、项目管理、计算机工程和数学等）的位置，并确立它们的分界；
 - 刻画软件工程学科的内容；
 - 提供使用知识体系的主题；
 - 为开发课程表和个人认证与许可材料提供一个基础；





软件工程知识体系 (SWEBOOK) 指南 2004 年版



软件工程知识体系知识域

