

软件工程

第一部分 软件工程概述

- 1.1 软件危机
- 1.2 软件工程概念与原则
- 1.3 软件生命周期
- 1.4 软件过程

软件工程概述

1

教学要求

- 了解软件工程学科发展的源起，即软件危机特点、出现原因与解决途径
- 掌握软件工程的定义与基本原理
- 理解软件生命周期概念及其各个阶段的任务
- 掌握不同的软件过程模型及其比较

软件工程概述

2

计算机软件

- ✓ 计算机系统由软件系统和硬件系统组成。
- ✓ 计算机软件包括程序，数据及其相关文档的完整集合。

软件工程概述

3

1.1 软件危机

计算机软件开发与维护中遇到一系列的严重问题。主要特征：

- ✓ 软件开发成本严重超标；
- ✓ 软件成本在计算机系统总成本所占比例逐年上升；
- ✓ 软件开发周期大大超过规定日期；
- ✓ 软件系统达不到用户要求；
- ✓ 软件质量难于保证；
- ✓ 软件修改、维护困难；
- ✓ 缺少适当的文档资料；
- ✓ 软件开发生产率无法满足需求。

软件工程概述

4

软件危机的产生原因

- 软件本身的特点
 - 计算机系统逻辑部件，缺乏“可见性”。质量难评价，难管理和控制、难维护。
 - 规模庞大，而且程序复杂性将随着程序规模的增加而呈指数上升。
- 软件开发与维护的方法
 - 对软件开发与维护的糊涂观念，如忽视软件需求分析的重要性、轻视软件配置、轻视软件维护等。
 - 在实践过程中或多或少地采用了错误的方法和技术。

软件工程概述

5

软件危机的消除途径

从管理（组织措施）和技术（方法与工具）两方面研究如何更好地开发和维护计算机软件。

- 对计算机软件概念的正确认识（软件由计算机程序、数据及文档组成；软件配置管理）；
- 认识软件开发中管理重要性（是“工程活动”）；
- 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（经验软件工程）；
- 开发和更好的软件工具（CASE）。

软件工程概述

6

1.2 软件工程

- 软件工程概念的出现源自软件危机。
- 概括地讲，软件工程从**管理和技术**两方面研究如何更好地开发和维护计算机软件的一门**工程学科**。
- 采用**工程的概念、原理、技术和方法**来开发与**维护软件**，把经过时间考验而证明正确的管理技术和当前能够得到的最好的技术方法结合起来，以**经济地（按进度）**开发出**高质量的软件**并有效地维护它。

软件工程概述

7

软件工程定义

- 1968年德国人 Bauer 在北大西洋公约组织会议上的定义：“建立并使用完善的工程化原则，以较经济的手段获得能在实际机器上有效运行的可靠软件的一系列方法”。
- 1983年 IEEE 的软件工程定义：“软件工程是开发、运行、维护和修复软件的系统方法”。
- 1993年 IEEE 的一个更加综合的定义：“将系统化的、规范的、可度量的方法应用于软件的开发、运行和维护的过程，即将工程化应用于软件中”。

软件工程概述

8

软件工程的本质特性

软件工程一些共性的特点：

- 关注于大型软件的构造
- 控制复杂性
- 有效应对变化
- 提高软件开发效率
- 组织和管理软件开发
- 有效地支持用户需求
- 有效地与不同领域的专家合作

软件工程概述

9

软件工程的基本原理

著名的软件工程专家 Barry Boehm 综合了软件工程的若干准则，提出了七条基本原理：

- ◆ **按软件生存期分阶段制定计划并认真实施**
把整个软件开发过程视为一项工程，把工程划分为若干阶段，分别制定每个阶段的计划，逐个实施。
- ◆ **坚持进行阶段评审** 前一阶段的结果将成为下一阶段的依据。坚持阶段的评审才能保证错误不传播到下一阶段。

软件工程概述

10

- 3) **坚持严格的产品控制** 将影响软件质量的因素在整个过程中置于严格控制之下（采用基线配置技术）
- 4) **使用现代程序设计技术** 先进的程序设计技术带来的是生产率和质量的提高。使用合适的开发模式和工具可以有效地建立功能强大的系统。
- 5) **明确责任，使得工作结果能够得到清楚的审查** 开发组织严格划分责任并制定产品的标准，使得每个成员的工作有据可依，确保产品的质量。

软件工程概述

11

- 6) **用人少而精** 开发组织不在人多，在于每个人的技能符合要求。同时用人少而精，可减少沟通路径，提高生产率。
- 7) **不断改进开发过程** 在开发的过程中不断总结经验，改进开发的组织和过程，有效地通过过程质量的改进，提高软件产品的质量。

软件工程概述

12

软件工程方法学

- 软件工程包括**技术**和**管理**两方面的内容，是技术与**管理**紧密结合所形成的工程学科。
- 管理就是通过**计划、组织和控制**等一系列活动，合理地配置和使用各种资源，以达到既定目标的过程。
- 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**，也称为**范型**。
- **软件工程方法学**包含3个要素：**方法**、**工具**和**过程**。

软件工程概述

13

- **软件工程方法学**包含3个要素：

- 方法** 完成软件开发的各项任务的技术方法，回答“怎样做”的问题。
- 工具** 为运用方法而提供的自动的或半自动的软件工程支撑环境。
- 过程** 为了获得高质量的软件所需要完成的一系列任务的框架，规定了完成各项任务的工作步骤。

软件工程概述

14

▪ 传统方法学：结构化方法学（生命周期方法学）

- 采用结构化技术(结构化分析、结构化设计与结构化编程)及其工具完成软件开发的各项任务
- 将软件开发过程划分成若干个阶段，阶段之间有严格的技术审查和管理复审。从技术和管理两方面对各个阶段的开发成果进行检查。
- 使用得十分广泛、也是非常成熟的软件工程方法学。在开发中小规模的软件时比较有效。

软件工程概述

15

▪ 面向对象方法学

- 把对象（数据与操作的融合）作为**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。类是对具有相同数据和相同操作的一组相似对象的定义。
- 采用继承机制将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。

软件工程概述

16

▪ 面向对象方法学与结构化方法学的比较

- **结构化方法学**强调自顶向下顺序地完成软件开发的各阶段任务。
- **面向对象方法学**的出发点和基本原则，尽量模拟人类习惯的思维方式，使开发软件的方法与过程尽可能接近人类认识世界解决问题的方法与过程，从而使描述问题的**问题空间**与实现解法的**解空间**在结构上尽可能一致。具有：概念和表示方法上的一致性；多次反复迭代的演化过程；促进了软件重用等优点。

软件工程概述

17