

◎ 计算机

“卓越工程师计划”应用型教材



Computer



UML面向对象 系统分析与设计教程

UML Object Oriented
System Analysis and Design

◎ 胡智喜 唐学忠 殷凯 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY <http://www.phei.com.cn>

计算机“卓越工程师计划”应用型教材

UML 面向对象系统分析与 设计教程

胡智喜 唐学忠 殷 凯 费贤举 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是由多年从事软件工程教学的教师和科研人员根据教学的特点精心组织和编写的。本书从面向对象技术的基本概念开始,由浅入深地介绍了 UML 语言建模的基本语法、建模的基本步骤、RUP 开发过程等,重点章节配套实例讲解,最后通过一个应用案例详细介绍了基于 UML 语言的面向对象建模开发过程。全书共 12 章,第 1~3 章介绍了面向对象分析与设计技术的基本概念;第 4~10 章详细介绍了基于 UML 语言进行面向对象建模的详细方法和步骤及注意事项;第 11 章介绍了 RUP 开发过程;第 12 章通过一个应用案例详细介绍了面向对象开发的过程。

本书既可作为相关专业的大专院校教材,也适合计算机软件开发人员使用。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

UML 面向对象系统分析与设计教程/胡智喜等编著. —北京:电子工业出版社,2014.7
计算机“卓越工程师计划”应用型教材

ISBN 978-7-121-23231-2

I. ①U… II. ①胡… III. ①面向对象语言—程序设计—高等学校—教材 IV. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 100925 号

责任编辑:刘海艳 (lhy@phei.com.cn)

印 刷:北京天宇星印刷厂

装 订:北京天宇星印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787×1 092 1/16 印张:13.75 字数:352 千字

版 次:2014 年 7 月第 1 版

印 次:2014 年 7 月第 1 次印刷

印 数:3 000 册 定价:39.00 元



凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

计算机“卓越工程师计划”应用型教材 编委会

主任委员：庄燕滨

名誉主任：杨献春 夏世雄

副主任委员：乐光学 汤克明 严云洋 吴克力 张永常 李存华
邵晓根 陈 荣 赵 梅 徐煜明 顾永根 常晋义

编委会秘书长：陶 骏

委员：王文琴 王 刚 刘红玲 何中胜 宋晓宁 张宗杰 张 勇
张笑非 李永忠 杨学明 胡局新 胡智喜 费贤举 徐 君
徐建民 郭小荟 高 尚 黄 旭

丛书序言

党的十八大提出要“努力办好人民满意的教育”，要“推动高等教育内涵式发展”，“全面实施素质教育，深化教育领域综合改革，着力提高教育质量，培养学生社会责任感、创新精神、实践能力。”这对高等教育提出了新的要求，明确了人才培养的目标和标准。

十八大明确指出“坚持走中国特色新型工业化、信息化、城镇化、农业现代化道路，推动信息化和工业化深度融合、工业化和城镇化良性互动、城镇化和农业现代化相互协调，促进工业化、信息化、城镇化、农业现代化同步发展”“推动信息化和工业化深度融合”对高等工程教育改革发展提出了迫切要求。

遵照《国家中长期教育改革和发展规划纲要（2010—2020年）》和《国家中长期人才发展规划纲要（2010—2020年）》，为贯彻落实教育部“卓越工程师教育培养计划”，促进我国由工程教育大国迈向工程教育强国，培养造就一批创新能力强、适应经济社会发展需要的高质量计算机工程技术人才，电子工业出版社决定组织相关实施和计划实施卓越计划以及江浙两省实施软件服务外包人才培养试点的地方高校的相关教师，在以往实践校企合作人才培养的基础上编写一套适合地方高校的计算机“卓越工程师教育培养计划”人才培养系列教材。

我们将秉承“行业指导、校企合作、分类实施、形式多样”的“卓越工程师教育培养计划”四原则，坚持“学科规范、本科平台、行业应用”，以“具备较为扎实的专业基础知识、拥有良好的职业道德素质、具有创新的计算机应用能力”为目标，探索“校企一体化”产学研结合人才培养模式改革，强化“岗位目标、职业培养”，努力实现计算机工程型技术人才（应用型）培养目标：

（1）尝试以“知识保障、能力渐进、素质为本，重视技术应用能力培养”为主线，坚持以“素质教育，能力培养”为导向，体现本科平台、能力定位、应用背景构建课程体系。

（2）尝试“以学生工程意识、创新精神和工程实践能力培养”为核心，坚持以“培养学生的工程化开发能力和职业素质”为原则，校企合作构建实践教学体系。

本系列教材基于“以德为先、能力为重、全面发展”的人才培养观念，在内容选择、知识点覆盖、课程体系安排、实践环节构建、企业强化训练上按照能力培养和满足职业需求为本进行了有益的、初步的探索。

然而，由于社会对计算机人才的需求广泛而多样，各领域的人才规格和标准既有共性又有特殊性，同时各相关高校在计算机相关专业设置以及人才培养的探索上各有特点，我们编写的本套系列教材目前只能部分满足计算机相关专业人才培养的需要。我们力争建立一个体系，以模块构建的增量方式实现教材编写的滚动、增加和淘汰，逐步建设可供地方高校计算机不同专业、针对不同领域培养计算机工程技术人才选择的教材库：①所有专业的公共基础课相对统一，不同专业的专业基础课按模块划分，各自专业的专业课按领域整合，拓展课紧跟技术和行业发展；②公共基础课、专业基础课以经典知识为主，专业课、拓展课与国际主流技术接轨；③实践环节或实践课程必须接纳企业文化、优选企业实际工程项目，体现校企合作，重视企业导师的参与。

“卓越工程师教育培养计划”的实施具有三个特点：一是行业企业深度参与培养过程；二是学校按通用标准和行业标准培养工程人才；三是强化培养学生的工程能力和创新能力。

本系列教材的编写得到了中软国际、苏州花桥国际商务区（及所属企业）、常州创意产业基地（及所属企业）等热心和关注计算机类人才培养的国家重点企业、园区的大力支持。我们曾以“目标明确、责任共担、实现共赢”为原则探索了多种人才培养合作途径：从师资培养到校企共建实训基地，到建立校内软件学院，再到学生进企业强化、顶岗实训……取得了一定的经验。在“卓越工程师教育培养计划”的实施中，企业和学校签订了全面合作协议，共同确定人才培养标准，制订人才培养方案，参与人才培养过程，提供企业学习课程和项目案例，确保学生在企业的学习时间。

同样，本系列教材的编写总结了参编高校和支撑企业在校企合作人才培养过程中共同取得的经验和教训，并涵盖了我们已经做的、想要做的实施卓越计划的理念和努力。这仅是初步的尝试，会存在许多不足和缺陷，但希望由此能起到抛砖引玉的作用。在卓越计划的实施探索中，我们衷心地希望能有更多的地方高校计算机院系、更多的行业企业加入团队，面对企业必须参与的国际化产业竞争，为培养优秀的、具有应用创新精神的计算机工程技术（包括软件）人才，企业和学校能深度合作、各尽其责；每一位教育工作者都能贡献自己的聪明才智，尽一份绵薄之力。

对给予本套丛书编审大力支持的江苏计算机学会、中国矿业大学计算机学院以及参与编写教材的高校、单位表示由衷的感谢！

计算机“卓越工程师计划”应用型教材编委会

前 言

面向对象分析和设计技术从诞生以来，一直受到了广大软件开发人员的喜爱。随着软件规模的不断扩大和软件复杂度的不断增加，软件开发人员之间迫切需要一种新的技术用于在软件设计之间进行沟通和交流，UML 建模技术正是在这样的背景下产生的。

最早的面向对象分析和设计方法主要有 Booch、Jacobson、Rumbaugh、Youdon 等，它们各有特色，又有不足之处，而且使用的术语不统一，缺乏共同的标准，常常给软件开发人员带来困惑。

UML 语言综合了目前主流的面向对象分析和设计技术，它为面向对象建模提出了一个统一的标准。1997 年，UML 被美国标准化组织 OMG 接受，经过不断使用、修改、补充、完善，UML 日趋成熟，得到众多计算机厂家如 IBM、HP、SUN、Microsoft 等的支持。

UML 适合系统开发过程中从用户需求开始到系统完成的各个阶段，通过静态结构建模和动态行为建模来抽象系统的模型，同时又可以将模型转化为面向对象语言实现的代码，为广大软件开发人员带来了极大的方便。

本教材是由多年从事软件工程教学的教师和科研人员根据教学的特点精心组织和编写的。在教材编写过程中，根据第 1 版《UML 面向对象分析与建模》在教学过程中的实际情况做了很大的修改，从软件工程的基本概念开始，由浅入深地介绍了面向对象技术和 UML 的基本语法、建模的基本步骤、RUP 开发过程等，最后通过一个应用案例详细介绍 UML 开发的过程。

本书共 12 章：第 1 章介绍了软件工程的基本概念；第 2 章介绍了面向对象开发技术的基本概念，第 3 章介绍了 UML 语言的基本语法、概念和符号；第 4~10 章详细介绍了 UML 静态建模和动态建模的详细方法步骤和注意事项；第 11 章介绍了 RUP 开发过程；第 12 章通过一个应用案例详细介绍了 UML 开发的过程。

本书理论联系实际，既可作为大专院校相关专业的教材，又可作为软件开发人员的技术参考书，尤其适合使用面向对象技术分析和建模的用户。在内容安排上，本着从入门到精通的原则，内容合理、语言通俗易懂。

本书由常州工学院胡智喜、唐学忠、殷凯、费贤举编著，参与编写的还有常州工学院王文、王文琴、王树锋。在本书的编写过程中我们得到了徐州工程学院程红林老师和中软国际的软件培训部的王晓华老师的大力支持，他们为本书提供了相当多的案例，另外，王文琴和王文两位老师也帮助编制了本书的部分图表，同时，本书编写过程中得到了计算机“卓越工程师计划”应用型教材编委会庄燕滨教授的鼓励和支持，特在此表示感谢。

编著者
2014.4

目 录

第 1 章 软件工程和建模	(1)
1.1 软件工程的发展历史	(1)
1.2 软件工程的基本原理	(2)
1.3 软件生命周期	(3)
1.4 软件开发模型	(5)
1.5 软件开发方法	(10)
1.5.1 面向过程的开发方法	(10)
1.5.2 面向数据结构的开发方法	(10)
1.5.3 面向对象的开发方法	(11)
1.6 软件开发方法的评价与选择	(12)
1.7 软件建模	(14)
本章小结	(16)
习题 1	(16)
第 2 章 面向对象技术	(17)
2.1 什么是面向对象	(17)
2.2 面向对象的基本概念	(18)
2.3 面向对象的分析	(22)
2.4 面向对象的设计	(23)
2.4.1 面向对象的设计准则	(23)
2.4.2 面向对象的设计过程	(25)
2.5 面向对象方法比较	(28)
2.5.1 Booch 面向对象方法	(29)
2.5.2 Jacobson 的面向对象方法	(31)
2.5.3 Coad-Yourdon 面向对象方法	(32)
2.5.4 OMT 面向对象方法	(34)
2.6 面向对象开发语言和建模工具	(36)
本章小结	(36)
习题 2	(36)
第 3 章 UML 语言基础	(37)
3.1 UML 发展历史	(37)
3.2 UML 的主要内容	(38)
3.2.1 UML 语义	(39)
3.2.2 UML 表示法	(39)
3.2.3 对象约束语言	(41)
3.3 UML 中的符号和图形	(41)

3.3.1	模型的概念	(41)
3.3.2	模型元素	(41)
3.3.3	模型元素之间的关系	(42)
3.3.4	模型组织	(43)
3.4	公共机制	(46)
3.4.1	修饰	(46)
3.4.2	说明	(46)
3.4.3	公共划分	(46)
3.5	扩展机制	(47)
3.5.1	构造型	(47)
3.5.2	标记值	(48)
3.5.3	约束	(48)
3.6	UML 的特点和应用领域	(49)
3.6.1	UML 的主要特点	(49)
3.6.2	UML 的应用领域	(50)
	本章小结	(52)
	习题 3	(52)
第 4 章	用例图	(53)
4.1	概述	(53)
4.2	系统	(54)
4.3	活动者	(55)
4.3.1	定义	(55)
4.3.2	确定活动者	(55)
4.3.3	活动者之间的关系	(56)
4.4	用例	(57)
4.4.1	定义	(57)
4.4.2	用例表示法	(57)
4.4.3	用例描述	(57)
4.5	用例之间的联系	(59)
4.5.1	泛化联系	(59)
4.5.2	使用联系	(59)
4.5.3	包含联系	(60)
4.5.4	扩展联系	(60)
4.6	用例建模	(61)
4.6.1	建立用例图	(61)
4.6.2	用例建模中应注意的问题	(61)
4.7	实例讲解	(62)
4.7.1	班级信息管理系统	(62)
4.7.2	图书借阅管理系统	(63)
	本章小结	(65)
	习题 4	(65)

第 5 章 对象图和类图	(67)
5.1 概述	(67)
5.2 对象类定义	(67)
5.2.1 类的名称	(68)
5.2.2 定义属性	(68)
5.2.3 定义操作	(69)
5.3 对象类的关联	(70)
5.3.1 关联的定义	(70)
5.3.2 关联类型	(71)
5.3.3 聚合和组合	(72)
5.3.4 泛化	(72)
5.3.5 依赖	(73)
5.4 对象图	(73)
5.5 接口	(74)
5.6 高级对象类	(75)
5.6.1 抽象类	(75)
5.6.2 模板对象类	(76)
5.7 对象类图建模	(76)
5.7.1 建立对象类图	(76)
5.7.2 类和对象建模中应注意的问题	(77)
5.8 实例讲解	(79)
5.8.1 图书借阅管理系统类图	(79)
5.8.2 旅游宾馆客房预订管理	(79)
本章小结	(80)
习题 5	(80)
第 6 章 交互图	(82)
6.1 概述	(82)
6.2 序列图	(82)
6.2.1 序列图的组成	(83)
6.2.2 同步消息和异步消息	(85)
6.2.3 分支和条件	(85)
6.2.4 循环	(86)
6.2.5 时间约束	(86)
6.2.6 对象创建和销毁	(87)
6.2.7 自调用和回调	(87)
6.3 协作图	(88)
6.3.1 协作图的组成	(88)
6.3.2 对象的创建与销毁	(89)
6.3.3 同步消息和异步消息	(90)
6.3.4 自调用和回调	(90)
6.4 交互图建模	(91)

6.4.1	建立交互图	(91)
6.4.2	交互图建模的基本步骤	(92)
6.4.3	交互图建模中应注意的问题	(92)
6.5	实例讲解——图书借阅管理系统交互图	(93)
6.5.1	图书借阅管理系统序列图	(93)
6.5.2	图书管理系统协作图	(95)
	本章小结	(96)
	习题 6	(96)
第 7 章	状态图	(97)
7.1	概述	(97)
7.2	状态及其分类	(97)
7.2.1	状态	(97)
7.2.2	状态的种类	(99)
7.3	状态机和状态图	(103)
7.3.1	状态机	(103)
7.3.2	状态图	(103)
7.4	消息、事件和状态迁移	(105)
7.4.1	消息	(105)
7.4.2	事件	(106)
7.4.3	守卫条件	(108)
7.4.4	动作表达式	(109)
7.4.5	状态迁移的种类	(109)
7.5	并发状态图	(112)
7.5.1	并发子状态	(112)
7.5.2	同步	(113)
7.6	状态图建模	(114)
7.6.1	建立状态图	(114)
7.6.2	状态图建模中应注意的问题	(115)
7.7	实例讲解	(116)
	本章小结	(116)
	习题 7	(117)
第 8 章	活动图	(118)
8.1	活动图概述	(118)
8.2	活动图的组成	(118)
8.2.1	组成要素	(119)
8.2.2	动作流	(120)
8.2.3	泳道	(121)
8.2.4	对象流	(122)
8.3	活动分解	(123)
8.4	活动图的并发与同步	(124)
8.4.1	并发与同步	(124)

8.4.2	同步状态	(125)
8.4.3	动态并发	(125)
8.5	活动图建模	(126)
8.5.1	建立活动图	(126)
8.5.2	活动图建模中应注意的问题	(127)
8.6	实例讲解	(127)
	本章小结	(128)
	习题 8	(128)
第 9 章	包图	(129)
9.1	概述	(129)
9.2	包的基本含义	(129)
9.2.1	包的语义和表示	(129)
9.2.2	包的嵌套	(131)
9.2.3	标准构造型	(131)
9.3	包的联系	(132)
9.3.1	依赖	(132)
9.3.2	泛化	(134)
9.4	包图	(135)
9.5	包图建模	(135)
9.5.1	建立包图	(135)
9.5.2	包图建模中应注意的问题	(137)
	本章小结	(138)
	习题 9	(138)
第 10 章	物理图	(139)
10.1	概述	(139)
10.2	构件图	(139)
10.2.1	构件	(139)
10.2.2	构件的种类	(142)
10.2.3	构件的联系	(143)
10.2.4	建立构件图	(144)
10.3	配置图	(148)
10.3.1	节点	(148)
10.3.2	节点的联系	(149)
10.3.3	建立配置图	(150)
10.4	物理图建模	(153)
	本章小结	(154)
	习题 10	(154)
第 11 章	RUP 统一建模过程	(155)
11.1	软件过程概述	(155)
11.1.1	软件过程介绍	(155)
11.1.2	当前流行的软件开发过程	(156)

11.2 RUP 简介	(159)
11.2.1 RUP 过程	(159)
11.2.2 RUP 过程的特点	(167)
11.3 RUP 的核心 workflow	(168)
11.3.1 业务建模	(168)
11.3.2 需求	(169)
11.3.3 分析和设计	(171)
11.3.4 实现	(172)
11.3.5 测试	(173)
11.3.6 部署	(175)
11.3.7 核心支持 workflow	(176)
本章小结	(178)
习题 11	(179)
第 12 章 应用系统案例	(180)
12.1 系统概述	(180)
12.2 系统需求	(181)
12.2.1 系统总体功能需求	(181)
12.2.2 各关键模块需求	(181)
12.3 系统用例模型	(183)
12.3.1 确定系统范围和系统边界	(183)
12.3.2 确定活动者	(183)
12.3.3 定义用例	(184)
12.3.4 建立用例图	(184)
12.3.5 用例描述	(186)
12.4 设计实体类模型	(193)
12.4.1 识别对象类	(193)
12.4.2 类图	(194)
12.4.3 类属性	(195)
12.5 设计接口和控制类模型	(196)
12.5.1 识别接口类	(197)
12.5.2 识别控制类	(198)
12.5.3 系统类模型	(199)
12.6 设计动态模型	(199)
12.6.1 数据访问类	(199)
12.6.2 建立序列图	(199)
12.7 系统部署	(205)
12.7.1 建立构件图	(205)
12.7.2 建立配置图	(206)
本章小结	(206)
习题 12	(206)

第 1 章

软件工程和建模

本章学习目标

- (1) 了解目前常用的软件开发方法；
- (2) 熟悉软件生命周期过程，理解常见的软件开发模型并会比较它们之间的区别；
- (3) 比较不同软件开发方法的适应场合；
- (4) 理解软件建模的意义和必要性。

1.1 软件工程的发展历史

自 1946 年，世界上第一台电子计算机研制成功以来，计算机已经从最初单纯的科学计算演变为解决各类问题不可替代的工具。层出不穷的软件开发技术、软件开发方法不断推动计算机技术向前发展。

在计算机系统发展的初期，软件功能简单，软件开发没有可以遵循的系统方法，大多数软件都是使用者为解决某个特定的科学计算而编写的，通用性十分有限。

从 20 世纪 60 年代中期到 70 年代中期，随着软件需求的扩大，软件产品越来越被广泛使用，各类专业的软件公司纷纷成立，软件行业进入了一个蓬勃发展的时期。但是，随着软件规模的急剧膨胀，软件需求也日益复杂，软件维护的难度也越来越大，开发成本呈指数级增长，失败的软件项目比比皆是。这就是我们所熟知的“软件危机”。

“软件危机”的出现使得人们开始对软件开发的方法进行重新审视。一个优秀的程序除了功能正确、性能优良之外，还应该易读、易懂、易用、易维护。

1968 年，北大西洋公约组织召集了当时全世界一流的程序员、计算机科学家以及工业界人士在德国召开了一次以摆脱“软件危机”为主题的国际学术会议。在该会议上，Fritz Bauer 首先提出了“软件工程”的概念，试图建立并使用正确的工程方法开发出低成本、高效、可靠的软件，从而解决或缓解软件危机。

所谓软件工程，就是采用工程化的概念、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，以经济地开发出高质量的软件并有效地维护它。概括地说，就是用工程、科学和数学的原则与方法研制、维护计算机软件的有关技术及管理方法。

软件工程是指导计算机软件开发和维护的工程学科。软件工程方法是完成软件工程项目的手段。软件工程使用的各种自动的或半自动的软件开发支持环境等软件工具，是软件开发活动中智力和体力的扩展与延伸。软件管理人员对软件开发的质量、进度、成本进行评估、管理和控制，包括人员组织、计划跟踪与控制、成本估算、质量保证、配置管理等。软件工程的方法、工具和过程构成了软件工程的三要素。

1.2 软件工程的基本原理

1983 年，Boehm 总结了多年来软件开发的经验，在一篇论文中提出了软件工程的 7 条基本原理。

(1) 用分阶段的生存周期计划进行严格的管理

Boehm 认为，在软件的整个生命周期中应该制定并严格执行 6 类计划：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。不同层次的管理人员都必须严格按照计划各尽其职地管理软件开发与维护工作，绝不能受客户或上级人员的影响而擅自背离预定计划。

(2) 坚持进行阶段评审

软件的质量保证工作不能等到编码阶段结束之后再进行。第一，大部分错误是在编码之前造成的，例如，根据 Boehm 等人的统计，设计错误占软件错误的 63%，编码错误仅占软件错误的 37%；第二，错误发现与改正得越晚，所需付出的代价也越高。因此，在每个阶段都要进行严格的评审，以便尽早发现在软件开发过程中所犯的错误，是一条必须遵循的重要原则。

(3) 实行严格的产品控制

在软件开发过程中不应随意改变需求，因为改变一项需求往往需要付出较高的代价。但是，在软件开发过程中改变需求又是难免的，由于外部环境的变化，相应地改变用户需求是一种客观需要，显然不能硬性禁止客户提出改变需求的要求，而只能依靠科学的产品控制技术来顺应这种要求。也就是说，当改变需求时，为了保持软件各个配置成分的一致性，必须实行严格的产品控制，其中主要是实行基准配置管理。所谓基准配置又称基线配置，它们是经过阶段评审后的软件配置成分（各个阶段产生的文档或程序代码）。基准配置管理也称为变动控制：一切有关修改软件的建议，特别是涉及对基准配置的修改建议，都必须按照严格的规程进行评审，获得批准以后才能实施修改。绝对不能谁想修改软件（包括尚在开发过程中的软件），就随意进行修改。

(4) 采用现代程序设计技术

从提出软件工程的观念开始，人们一直把主要精力用于研究各种新的程序设计技术，如 20 世纪 60 年代末提出的结构程序设计技术，以及后来提出的面向对象程序设计技术等。实践表明，采用先进的技术既可提高软件开发的效率，又可提高软件维护的效率。

(5) 软件工程结果应能清楚地审查

软件产品不同于一般的物理产品，它是看不见摸不着的逻辑产品。软件开发人员的工作进展情况可见性差，难以准确度量，从而使得软件产品的开发过程比一般产品的开发过程更难于评价和管理。为了提高软件开发过程的可见性，更好地进行管理，应该根据软件开发项目的总目标及完成期限，规定开发组织的责任和产品质量标准，从而使得所得到的结果能够清楚地审查。

(6) 开发小组的人员应该少而精

软件开发小组的组成人员的素质应该高，而人数则不宜过多。开发小组人员的素质和数量是影响软件产品质量和开发效率的重要因素。素质高的人员的开发效率比素质低的人员的开发效率可能高几倍至几十倍，而且素质高的人员所开发的软件中的错误明显少于素质低的人员所开发的软件中的错误。此外，随着开发小组人员数目的增加，因为交流情况讨论问题而造成的通信开销也急剧增加。因此，少而精的开发小组是软件工程的一条基本原理。

(7) 承认不断改进软件工程实践的必要性

遵循上述六条基本原理，就能够按照当代软件工程基本原理实现软件的工程化生产，但是，仅有上述六条原理并不能保证软件开发与维护的过程能赶上时代前进的步伐，不能跟上技术的不断进步。Boehm 提出应把承认不断改进软件工程实践的必要性作为软件工程的第 7 条基本原理。按照这条原理，不仅要积极主动地采纳新的软件技术，而且要注意不断总结经验，例如，收集进度和资源耗费数据，收集出错类型和问题报告数据，等等。这些数据不仅可以用来评价新的软件技术的效果，而且可以用来指明必须着重开发的软件工具和应该优先研究的技术。

1.3 软件生命周期

软件生命周期是指软件从立项、功能确定、设计，到开发成功投入使用，并在使用中不断地修改、增补和完善，直至被新的需要所替代而停止该软件的使用的全过程。

软件生命周期经历软件定义、软件设计、软件使用与维护三阶段，又可以具体分成以下几个子阶段，即可行性研究、需求分析和定义、总体设计、详细设计、编码（实现）、软件测试、运行/维护。

1. 软件定义时期

软件定义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。这个时期的工作通常又称为系统分析，由系统分析员负责完成。软件定义时期通常进一步划分成三个阶段，即问题定义、可行性研究和需求分析。

(1) 问题定义

问题定义必须明了要解决的问题是什么，通过问题定义阶段的工作，系统分析员应该提出关于问题性质、工程目标和规模的书面报告。通过对系统的实际用户和使用部门负责人的访问调查，分析员扼要地写出对问题的理解，征求用户意见之后，统一对问题的理解，最后得出一份双方都满意的文档。

(2) 可行性研究

可行性研究阶段讨论问题涉及的范围，探索这个问题是否值得去解，是否有可行的解决办法。可行性研究应该比较简短，这个阶段的任务不是具体解决问题。可行性研究的结果是使用部门负责人做出是否继续进行这项工程的决定的重要依据。

在问题定义阶段提出的对工程目标和规模的报告通常比较含糊。可行性研究阶段应该导出系统的高层逻辑模型，通常用数据流图表示，并且在此基础上更准确、更具体地确定工程规模和目标。然后分析员更准确地估计系统的成本和效益，对建议的系统进行成本/效益分析。

(3) 需求分析

需求分析是准确地确定：为了达到用户要求和系统的需求，系统必须做什么，系统必须具备哪些功能。在需求分析阶段确定的系统逻辑模型是以后设计和实现目标系统的基础，因此必须准确完整地体现用户的要求。

用户了解所面对的问题，知道必须做什么，但是通常不能完整准确地表达出要求，更不知道怎样利用计算机解决问题；软件开发人员知道怎样用软件实现人们的要求，但是对特定用户的具体要求并不完全清楚。在需求分析阶段必须和用户密切配合，充分交流信息，以得出经过用户确认的系统逻辑模型。

需求分析阶段应有用户参加，必须提出完整准确的系统逻辑模型，并经过用户确认。

2. 软件开发时期

软件开发时期具体设计和实现在前一个时期定义的软件，它通常由下述四个阶段组成：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。

(1) 总体设计

这个阶段必须要完成的任务是如何解决问题。从开发规模和开发成本上考虑，有三种可能的方案。

① 低成本的解决方案，系统只能完成最必要的工作。

② 中等成本的解决方案，系统不仅能够很好地完成预定的任务，使用起来很方便，而且可能还具有用户没有具体指定的某些功能和特点。虽然用户没有提出这些具体要求，但是系统分析员根据自己的知识和经验断定，这些附加是应该有的，或是不能少的。

③ 高成本解决方案，系统具有用户可能希望有的所有功能和特点。

系统分析员应该使用系统流程图或其他工具描述每种可能的解决方案，估算每种方案的成本和效益，还应该在充分权衡各种方案利弊的基础上，推荐一个较好的系统方案，并且制定实现所推荐的系统的详细计划。

通常用层次图或结构图描绘软件的结构，用结构设计的方法设计软件的结构，程序应该模块化，也就是说，一个大程序应该由许多规模适中的模块按合理的层次结构组织而成，确定程序由哪些模块组成以及模块间的关系。

(2) 详细设计

详细设计阶段的任务就是给出问题求解的每一步骤，给出怎样具体地实现这个系统的描述。这个阶段的任务还不是编写程序，而是设计出程序的详细规格说明。这种规格说明的作