

软件工程系系列规划教材

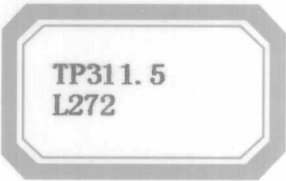
软件工程概论

李彤 王炜 郁湧 编著

INTRODUCTION
TO SOFTWARE
ENGINEERING



科学出版社



郑州大学 *040107783861*

软件工程系列规划教材

软件工程概论

李彤 王炜 郁湧 编著



科学出版社

北京

TP311.5
L272

内 容 简 介

本书以 CDIO 工程教育模式的四个阶段——构思、设计、实现和运作为主线,比较全面地介绍了软件工程的基本概念、基本理论、基本方法及其应用技术。全书共 14 章,分为 5 篇。导引篇包括第 1、2 章,介绍了软件工程的基础知识、CDIO 工程教育模式和软件过程相关模型;构思篇包括第 3~5 章,从传统分析方法和面向对象分析方法两个层次讨论了需求分析的相关内容,重点介绍了传统软件需求分析方法、面向对象的基本原理和面向对象需求分析方法;设计篇包括第 6~8 章,主要关注软件工程中的设计方法和技术,重点介绍了软件体系结构、传统软件设计方法和面向对象设计方法;实现篇包括第 9~11 章,其核心内容是软件工程中的实现方法和技术,讨论了如何选定程序设计语言,介绍了软件开发的一些新方法、软件测试的相关原理和技术;运作篇包括第 12~14 章,内容包括软件维护与演化、软件项目管理和软件过程改进。

本书可作为高等院校高年级本科生、研究生学习软件工程的教材,也可供从事软件开发和应用的有关人员参考。

图书在版编目(CIP)数据

软件工程概论/李彤,王炜,郁湧编著. —北京:科学出版社,2012

软件工程系列规划教材

ISBN 978-7-03-033493-0

I. ①软… II. ①李…②王…③郁… III. ①软件工程-高等学校-教材
IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2012)第 020366 号

责任编辑:潘斯斯 刘兰霞 / 责任校对:包志虹

责任印制:张克忠 / 封面设计:迷底书装

科学出版社 出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

北京市文林印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2012 年 2 月第 一 版 开本:787×1092 1/16

2012 年 2 月第一次印刷 印张:18 1/2

字数:470 000

定价:43.00 元

(如有印装质量问题,我社负责调换)

“软件工程系列规划教材”专家委员会

主任委员 陈国良 院士 中国科学技术大学

副主任委员 陈平 教授 西安电子科技大学

侯义斌 教授 北京工业大学

李彤 教授 云南大学

胡华强 编审 科学出版社

委 员 (按姓氏笔画排序)

丁刚毅 教授 北京理工大学

卢苇 教授 北京交通大学

朱敏 教授 四川大学

陈珉 教授 武汉大学

陈越 教授 浙江大学

肖来元 教授 华中科技大学

武波 教授 西安电子科技大学

柳青 教授 云南大学

周激流 教授 成都大学

赵一鸣 副教授 复旦大学

骆斌 教授 南京大学

秦志光 教授 电子科技大学

黄虎杰 教授 哈尔滨工业大学

傅育熙 教授 上海交通大学

秘 书 长 柳青 教授 云南大学

秘 书 张德海 副教授 云南大学

周维 副教授 云南大学

前言

当今,软件产业已经成为国民经济与社会发展的先导性和战略性产业,是信息产业与国民经济新的增长点和重要支柱。采用先进的工程化方法进行软件开发和生产是实现软件产业化的基本方法,而该方法的顺利实施则依赖于工程型人才的数量和质量。从这个意义上来说,软件产业的竞争越来越集中到工程型人才的竞争。然而,长期以来,我国工程型软件人才培养的现状远远不能满足软件产业发展的需要。因此,如何为业界培养工程型软件人才,特别是高层次工程型软件人才,已成为众多高校所面临的一个重大课题。

自软件工程概念诞生以来,围绕着实现软件优质高产这个目标,我们的前辈们从技术到管理开展了大量卓有成效的探索和实践,形成“软件工程”这一计算机领域的新兴学科,并从计算机科学与技术学科中独立出来,逐渐形成一门独立的一级学科。软件工程教育的地位变得越来越重要,软件工程课程也成为软件工程、计算机科学与技术等专业的必修课程。

然而,软件工程课程是工程性较强的课程,传统的软件工程教学脱离了工程环境,学生很难理解相关的理论知识,因此,无法满足培养工程型软件人才的需要。CDIO是由MIT等世界一流工科大学提出的先进工程教育模式,涵盖了从理念到大纲、课程、教学直至评估的一整套工程教育模式。本书以CDIO大纲为指导,运用CDIO先进工程教育思想,试图将软件工程的基本概念、原理、方法、技术、工具和管理与实际产品从研发到运行的生命周期形成映射,让学生以主动的、实践的、课程之间有机联系的方式经历工程实践的四个重要阶段:构思(C)、设计(D)、实现(I)和运作(O)。

全书共14章,分为5篇,各篇内容简介如下。

(1) 导引篇(第1、2章)。对软件工程的基本概念、理论、方法及CDIO思想进行简要的阐述,试图让读者掌握基于CDIO思想的软件工程的概貌。

(2) 构思篇(第3~5章)。CDIO教育思想认为“构思”即按照客户的需求,考虑技术、企业战略等因素,设立系统目标和要求;定义功能、概念和结构,为项目的设计、实现和运作奠定基础。本篇从传统分析方法和面向对象分析方法两个层次分别讨论了需求分析过程。

(3) 设计篇(第6~8章)。本篇在构思阶段成果的基础上,将关注点放在软件系统“如何实现”上。不仅要说明为实现构思阶段所产生的需求模型必须引入哪些计算模块及模块之间的关系,还必须从提高软件设计质量和效率方面考虑如何改进软件结构。

(4) 实现篇(第9~11章)。本篇不仅讨论了如何运用选定的程序设计语言,把设计阶段所产生的软件设计模型转换为该语言所书写的源代码,同时还讨论了软件系统的测试和开发方法等问题。

(5) 运作篇(第12~14章)。运作阶段的关注点在于如何对实现阶段产生的产品、系统或过程进行运行和优化;培训和操作;改进和演变;弃置和运行管理。

本书为践行CDIO教学理念还设计了一个实践项目,依托该项目以5~6名学生为一组,要求学生先后完成构思性实践、设计性实践、实现性实践和运作性实践所要求的相关任务。本书力图做到结构合理,内容丰富,讲解由浅入深,既体现知识点的连贯性、完整性,又体现其在工程中的应用。

在本书编写过程中,美国 Gordon 大学的 Russel C. Bjork 教授给予了大力支持和帮助,为作者提供了自动提款机案例。此外,本书的编写还得到了云南大学软件学院柳青教授、张德海副教授的关心和帮助。在本书付梓之际,谨向他们表示衷心的感谢!

本书的编写得到了国家教学质量工程项目“基于 CDIO 的国际化、工程化软件人才培养创新实验区”、“国家精品课程——软件工程”和云南省教学质量工程项目“CDIO 软件人才培养教学团队”的支持,在此一并致谢。

由于作者的水平和学识所限,书中不足之处在所难免,真诚希望读者、专家和同行不吝赐教。

作者
2011 年秋于云南大学

目 录

前言

导 引 篇

| | |
|-----------------|----|
| 第 1 章 软件工程概述 | 1 |
| 1.1 软件的概念和分类 | 1 |
| 1.2 软件危机与软件工程 | 3 |
| 1.3 CDIO 工程教育模式 | 7 |
| 1.4 CDIO 大作业 | 9 |
| 习题 | 11 |
| 第 2 章 软件过程 | 12 |
| 2.1 绪论 | 12 |
| 2.2 软件开发的主要活动 | 15 |
| 2.3 软件过程模型 | 20 |
| 2.4 软件过程中的并行工程 | 27 |
| 习题 | 32 |

构 思 篇

| | |
|----------------------|----|
| 第 3 章 传统软件需求分析 | 33 |
| 3.1 需求分析与需求工程 | 33 |
| 3.2 软件需求工程过程 | 37 |
| 3.3 软件需求获取 | 40 |
| 3.4 结构化分析 | 45 |
| 3.5 案例分析 | 51 |
| 习题 | 55 |
| 第 4 章 面向对象方法概论 | 56 |
| 4.1 什么是面向对象 | 57 |
| 4.2 面向对象方法在软件工程学中的作用 | 58 |
| 4.3 面向对象方法的主要概念 | 62 |
| 4.4 面向对象方法与 UML | 69 |
| 习题 | 82 |
| 第 5 章 面向对象需求分析 | 83 |
| 5.1 面向对象需求分析概述 | 83 |
| 5.2 案例 | 86 |
| 5.3 建立用例模型 | 87 |
| 5.4 发现对象和类 | 92 |

| | |
|------------------|-----|
| 5.5 建立行为模型 | 101 |
| 5.6 需求验证 | 106 |
| 习题 | 109 |

设计篇

| | |
|------------------------------|-----|
| 第6章 软件体系结构 | 110 |
| 6.1 概述 | 110 |
| 6.2 软件体系结构的类别及重要性 | 111 |
| 6.3 软件体系结构的构成 | 111 |
| 6.4 软件体系结构的描述语言 | 114 |
| 6.5 体系结构模式和风格 | 116 |
| 6.6 体系结构设计原理 | 122 |
| 6.7 分布式软件体系结构 | 124 |
| 习题 | 128 |
| 第7章 传统软件设计 | 129 |
| 7.1 软件设计的概念与原则 | 129 |
| 7.2 结构化设计方法 | 134 |
| 7.3 变换型设计与事务型设计 | 135 |
| 7.4 数据库设计 | 138 |
| 7.5 结构化程序设计 | 142 |
| 7.6 案例分析:系统设计 | 144 |
| 7.7 案例分析:办公室管理子系统分析与设计 | 150 |
| 习题 | 159 |
| 第8章 面向对象设计 | 160 |
| 8.1 面向对象设计概述 | 160 |
| 8.2 问题域设计 | 164 |
| 8.3 驱动控制设计 | 168 |
| 8.4 用户界面设计 | 170 |
| 8.5 数据库系统设计 | 172 |
| 8.6 类设计 | 173 |
| 习题 | 175 |

实现篇

| | |
|---------------------------|-----|
| 第9章 面向对象程序设计 | 176 |
| 9.1 面向对象程序设计的目的 | 176 |
| 9.2 选择程序设计语言 | 177 |
| 9.3 面向对象程序设计风格 | 178 |
| 9.4 面向对象程序设计准则 | 182 |
| 习题 | 183 |

| | |
|------------------------------|-----|
| 第 10 章 新兴软件开发方法 | 184 |
| 10.1 敏捷软件开发方法 | 184 |
| 10.2 软件复用 | 191 |
| 10.3 基于构件的软件工程 | 196 |
| 习题 | 200 |
| 第 11 章 软件测试 | 201 |
| 11.1 软件测试的基本概念 | 201 |
| 11.2 白盒测试 | 204 |
| 11.3 黑盒测试 | 208 |
| 11.4 单元测试 | 210 |
| 11.5 集成测试 | 213 |
| 11.6 确认测试 | 215 |
| 11.7 面向对象的软件测试 | 218 |
| 习题 | 224 |

运 作 篇

| | |
|-----------------------------|-----|
| 第 12 章 软件维护与演化 | 225 |
| 12.1 概述 | 225 |
| 12.2 软件维护 | 227 |
| 12.3 软件演化 | 233 |
| 习题 | 237 |
| 第 13 章 软件项目管理 | 238 |
| 13.1 软件项目管理概述 | 238 |
| 13.2 软件项目需求管理 | 240 |
| 13.3 软件项目成本管理 | 247 |
| 13.4 软件项目进度管理 | 258 |
| 13.5 软件项目风险管理 | 266 |
| 13.6 软件项目配置管理 | 272 |
| 习题 | 276 |
| 第 14 章 软件过程改进 | 277 |
| 14.1 过程与产品质量的关系 | 277 |
| 14.2 软件过程的建立与分析 | 278 |
| 14.3 过程度量和监控 | 280 |
| 14.4 过程变更和改进 | 281 |
| 14.5 过程改进框架 | 282 |
| 习题 | 284 |
| 参考文献 | 285 |

导 引 篇

工程教育的目的是将学生培养成为“来之能战”的工程师,即在其从事职业前就具备较好的工程能力和深厚的技术基础知识。为了达到这个目标,工程教育界引入了 CDIO 先进教育理念,试图不断地改进本科工程教育的质量和内涵。软件工程作为一门研究如何按工程化的原则和方法组织软件开发的学科,应当借鉴 CDIO 原理,丰富软件工程教学的内涵与外延。本篇对软件工程的基本概念、理论、方法以及 CDIO 思想做了简要的阐述,试图让读者掌握基于 CDIO 思想的软件工程的概貌。

第 1 章 软件工程概述

软件工程学科自 1968 年创立以来,已历经 40 多年。软件工程含义有二:其一,工程的含义,即软件分析、设计、实现、维护、管理等活动之集合;其二,科学的含义,即软件工程学科中所涉及的理论、原理、方法和技术的集合。软件工程学科的建立开启了软件开发从“艺术”、“技巧”和“个人行为”向“工程”和“群体协作”转化的历程。该历程同样见证了软件工程学科从幼稚到成熟的历史轨迹。

本章将对软件工程的相关概念、原理、技术、方法、知识体系以及 CDIO 工程教学模式进行简要介绍,使读者对软件工程的总体框架和 CDIO 工程教育模式获得初步的了解。

1.1 软件的概念和分类

1.1.1 软件的概念

50 多年前,人们对软件的概念知之甚少,甚至从未听说过这个词。如今,软件已经渗透到人类生活的方方面面,但人们对于软件的理解并不准确。“软件就是程序,软件开发就是编程”这种错误的观点仍然存在。随着软件工程新方法、新技术和新理论的出现,软件的外延和内涵较传统软件已经发生了巨大的改变。因此正确地把握软件的概念对于正确认识软件工程具有非常重要的意义。

关于软件的一种公认的定义是:软件是计算机系统中与硬件相互依存的另一部分。软件是程序、数据及其相关文档的完整集合。其中,程序是按事先设计的功能和性能要求执行的指令序列;数据是使程序能够正确地处理信息的数据结构;文档是与程序开发、维护和使用有关的图文资料。简单地表述,可以认为:软件=程序+文档。

不同的时代赋予程序不同的含义。在结构化程序设计时代,程序的最小单位是函数及子程序,程序与数据是分离的。在面向对象程序设计时代,程序的最小单位是类,在类中封装了相关的数据及对数据的操作。如果摒弃这些表现形式上的差异,可以归纳出软件具有如下属性。

1. 形态特性

软件是逻辑实体。因此是无形的、不可见的,当然无法用几何尺寸、物理性质和化学成分等属性对其进行度量。即便如此,也不能因而否定它的存在和低估它的价值。

2. 质量特性

控制软件产品的质量是困难的,表现在以下方面:

(1) 目标软件的需求在软件开发之初常常是不确切的,也不容易确切地给出。同时,需求的变更在软件开发过程中也是经常发生的,这就使软件质量控制变得异常困难。

(2) 由于软件产品是人类思维的外化产物,而人类的思维方式是因人而异、因时而异、因事而异的。这样很难如同规范劳动密集型制造业一样约束软件开发的过程,所以软件产品的质量是难以控制的。

(3) 软件测试技术是保证软件质量的重要手段之一,然而该技术存在先天不足。软件测试只能证明软件是存在错误的,永远不能证明软件是没有错误的。因为,任何测试都无法穷尽软件运行过程中的所有场景,无法枚举软件运行过程中的庞大的应用实例数据。仅仅是在极大数量的应用场景和数据中,选取极为有限的一部分用以测试,因此,导致无法得到完全没有缺陷的软件产品。

3. 智能特性

软件是人类思维的外化产物,它的开发基于人们大量的脑力劳动,是知识、实践经验和人类智慧的沉淀。它可以帮助解决复杂的计算、分析、判断和决策等问题,因此具有了部分的类似人类的智能。

4. 开发特性

软件开发的自动化(简称软件自动化)从某种意义上来说是软件工程学科发展的终极目标,但到目前为止,距离该目标还有很长的路要走。原因在于软件开发仍然包含了相当分量的个体劳动,使得这一大规模知识密集型工作充满了个人因素和个人行为。软件的开发不同于传统制造业。传统制造业的工艺都已经相当成熟,大规模地采用了自动化生产技术,摆脱了手工作坊式的生产;而大多数的软件产品是根据用户的需求进行定制开发的个性化产品,虽然出现了一些具有现代工业生产特征的软件开发技术和方法(如构件技术等),但现实情况是,这些技术只能从某一方面提高软件开发的自动化程度,改变或改善软件产品的生产效率。

5. 环境特性

软件的开发和运行依赖于不同的计算机系统环境(包括支持它的开发和运行的相关硬件和软件)。软件对于计算机系统的环境有着不可摆脱的依赖性。

6. 生产特性

与硬件或传统的制造业产品的生产过程不同,软件一旦设计开发出来,可以同时提供给多个用户使用。软件的复制是十分简单的,其成本也是极为有限的。正因为如此,设计开发成本占据了大部分软件产品的生产成本。

7. 演化特性

软件投入使用以后,随着用户需求的变更、新生态环境的变化、新错误的发现,使得原有软件逐渐沦为遗产软件。但完全抛弃这些遗产软件,重新开发适合当前环境和技术要求的新系统往往又是代价高昂的,有些时候甚至是不可行的。因此软件需要具备演化能力,针对相应的变化动态地进行调整。

8. 管理特性

对软件开发,特别是大型软件开发过程的管理,实际上是对大规模知识型工作者的智力劳动管理,包括必要的培训、指导、激励、制度化规程的推行、过程的量化分析与监督以及沟通、协调,甚至过程文化的建立和实施。因此,很难借鉴管理产业工人的方法进行管理。

9. 应用特性

软件的应用极为广泛,如今它已渗透到社会生活的各个领域,软件产业现已成为关乎国计民生的重要产业。

1.1.2 软件的分类

按照软件的作用,一般可以将软件分为以下两种。

1. 系统软件

系统软件是指控制和协调计算机及外部设备,支持应用软件开发和运行的各种程序的集合。其主要功能是调度、监控和维护计算机系统,管理计算机系统中各种独立的软硬件,使得它们可以协调工作。操作系统、数据库管理系统、设备驱动程序以及通信和网络处理程序等均属于系统软件。系统软件是计算机系统必不可少的一个组成部分。

2. 应用软件

应用软件是在系统软件的支持下,面向特定领域内开发,为特定目的服务的一类软件。例如,办公软件、即时通信软件、财务软件等都是应用软件。

1.2 软件危机与软件工程

1.2.1 软件危机

20世纪60年代,计算机的应用在很多行业得到广泛普及,并且待解决问题的规模及难度也得到了前所未有的增加。但是软件的发展速度却远远滞后于硬件的发展速度,不能满足社会日益增长的需求。这个时期软件产业的困难局面称为软件危机。具体来说,周期长、成本高、质量差、维护困难是软件危机的基本特征。

这些特征表现在具体的软件开发项目上,最具有代表性的实例是美国IBM公司在1963~1966年为IBM 360机开发的操作系统。这个项目的负责人F. D. Brooks在总结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。”

软件危机发生的原因主要有以下几个方面：

(1) 随着软件规模的增大,其复杂性往往呈指数级升高。为了在预定时间内开发出规模庞大的软件,必须以多人分工,协同工作的模式进行软件开发。然而,保证将多人完成的程序集成在一起,形成一个能够满足用户需求的软件系统是一件极其困难的工作。其原因在于,该过程涉及软件开发的分析方法、设计方法、编程方法、维护方法和科学管理方法。这些方法的知识在当时还是欠缺的。

(2) 软件开发人员与用户的交流存在障碍。除了知识背景的差异,缺少合适的交流方法及需求描述工具也是一个重要原因,这使得获取的需求不充分或存在错误,存在的问题在开发的初期难以发现,往往在开发的后期才暴露出来。使得开发周期延长,成本增加。

(3) 缺乏软件开发的经验和有关软件开发数据的积累,使得开发工作的计划很难制订。通常的情况是,开发人员主观盲目地制订开发计划,而该计划与实际情况相差甚远,致使预算经常被突破,工期一拖再拖。

(4) 缺少有效的软件评测手段,提交用户的软件质量差。

(5) 软件开发过程不规范,缺少方法论和规范的指导,开发人员各自为战,缺少整体的规划和配合,不重视文字资料工作,软件难以维护。

1.2.2 软件工程

针对软件危机,北大西洋公约组织在1968年10月召开的计算机科学会议上,Fritz Bauer首次提出“软件工程”的概念,试图将工程化方法应用于软件开发。科学家经过多年实践,最后得出一个结论:按工程化的原则和方法组织软件开发工作是有用的,是摆脱软件危机的一条主要出路。

历史上人们曾经给软件工程下过许多定义,例如,Fritz Bauer认为软件工程是“为了经济地获得可靠的和能在实际机器上高效运行的软件,而建立和使用的健全的工程原则”。Boehm认为软件工程是“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”。1983年IEEE给出的定义是“软件工程是开发、运行、维护和修复软件的系统方法”。其中“软件”的定义为:计算机程序、方法、规则、相关的文档资料及在计算机上运行时所必需的数据。

2010年教育部高等学校软件工程专业教学指导分委员会在2010年制定的《高等学校软件工程本科专业规范》中对软件工程的定义为“应用计算机科学理论和技术以及工程管理原则和方法,按预算和进度,实现满足用户要求的软件产品的定义、开发、发布和维护的工程或进行研究的学科”。

概括地说,软件工程是指导软件开发和维护的工程性学科,它以计算机科学理论和其他相关学科的理论为指导,采用工程化的概念、原理、技术和方法进行软件的开发和维护,把经过时间考验而证明是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,以较少的代价获得高质量的软件并维护它。

综上所述,软件工程是一门综合性学科,在40多年的发展过程中,已经形成了特有的方法、理论和技术。这些方法、理论和技术在某一些方面相似于传统工程学科,如建筑、桥梁、制造等,但也存在着巨大的差别。

(1) 软件是逻辑产品而不是实物产品,因此软件不是有形产品。这就意味着软件开发的费

用主要集中在设计、开发上而不在生产上。另外,由于软件是逻辑产品,因此不存在磨损、老化的情况,其可靠性是由逻辑性所确定。

(2) 软件开发在许多方面涉及数学证明,特别是软件开发的理论。但软件产品的评价的主要指标是软件在问题求解中是否有用,而不是决定于抽象的正确性判定的数学标准。即评价软件产品时主要使用的是工程标准,而不是数学标准。

(3) 实现软件产品的功能依赖于硬件和软件的运行环境以及人们对它的操作。没有计算机相关硬件的支持,软件难有实用价值。同样,没有软件支持的计算机硬件,也只是毫无使用价值的机器,软件与硬件的密切相关的程度是一般工程所没有的。

(4) 软件产品较一般的工程性产品具有较高的复杂性。软件的复杂性主要包括三方面:其一,功能的多样性。软件提供的功能较一般工程产品更为复杂多样。其二,实现的多样性。由于软件开发依赖于开发人员。不同的开发人员对软件有不同的认识,也就会有不同的实现方法。所以实现上的差异会带来使用上的差异。其三,形态的不可观察性。由于软件是逻辑产品,因此对软件进行观察是非常困难的,这又为软件开发管理增加了难度。总之,前两方面属于技术复杂性,后一方面属于管理复杂性,为了控制软件复杂性,人们进行了各种研究,这在一般工程中是前所未有的。

(5) 任何一种工程,在其早期总是人工密集的,而到成熟时期则成为资金密集的。但是软件工程也有它自己的特点,尽管今后可能有许多活动可以自动化,但软件工程始终是智力密集型产业。

1.2.3 软件工程学科的范畴

40 多年来,围绕着实现软件优质高产这个目标,前辈们从技术到管理开展了大量的实践,逐渐形成了“软件工程”这一计算机新兴学科,图 1-1 列举了软件工程学科所包含的主要内容。

1. 方法学

早期的程序设计带有非常多的个人因素。因此,软件开发被视为一种艺术,程序员各行其是,并无统一的方法可循。20 世纪 60 年代后期兴起的结构化程序设计方法,使人们认识到采用结构化的方法来编写程序,不仅可以改善程序结构,而且也能提高软件的可靠性与生产率。

随后,人们又认识到编写程序仅是软件开发过程中的一个环节。一个有效的软件开发过程应该包括“需求分析”、“软件设计”、“程序设计”等多个阶段。把结构化的思想扩展到分析阶段和设计阶段,于是形成了“结构化分析”与“结构化设计”等传统的软件开发技术。除结构化方法以外,还有其他一些方法也体现了结构化程序设计的思想,如 Jackson 方法、LCP 方法等。尽管这些方法的具体内容各有不同,但它们都遵循结构化程序设计的原则,都对软件开发步骤和文档格式提出了规范的要求。自此,软件生产已经摆脱了过去随心所欲的“个人化”的状态,进入有章可循的、向结构化和标准化迈进的“工程化”阶段。

20 世纪 80 年代出现的 Smalltalk、C++ 等语言,促进了面向对象程序设计的广泛流行。但是,仅仅使用面向对象程序设计并不能产生最好的效果,只有在软件开发的早期甚至全过程都采

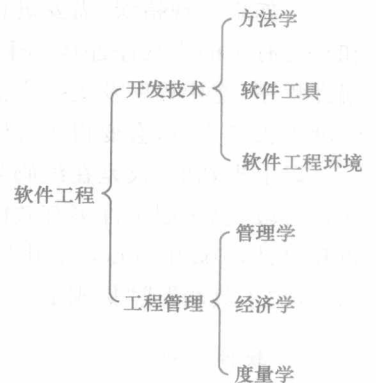


图 1-1 软件工程包含的主要内容

用面向对象技术,才能更好地发挥该技术的固有优势。于是,包括“面向对象需求分析(Object Oriented Analysis, OOA)、面向对象设计(Object Oriented Design, OOD)、面向对象程序设计(Object Oriented Programming, OOP)”在内的一系列面向对象软件工程方法学相继问世,并逐步取代传统软件开发方法,成为许多软件工程师的首选方法。同时,面向对象技术还促进了“软件复用”技术的发展。软件复用技术提高了软件开发的效率,而软件开发又为软件复用提供了更多的选择,并最终使软件复用技术成为当前主流的软件开发方法之一。

近年来,面向服务的体系结构(Service Oriented Architecture, SOA)成为一种新兴的软件工程方法学。SOA是一个构件模型,该模型将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来,形成一个较大规模的计算实体。其中,接口采用中立的方式进行定义,它独立于实现服务的硬件平台、操作系统和编程语言。这使得架构在各类硬件平台、操作系统和编程语言之上的服务,可以以一种统一和通用的方式进行交互。这种具有中立的接口定义(没有强制绑定到特定的实现上)的特征称为服务之间的松耦合。松耦合系统通常是基于消息的系统,此时客户端和远程服务并不知道对方是如何实现的。客户端和服务之间的通信由消息的架构支配。只要消息符合协商的架构,客户端或服务的实现就可以根据需要进行更改,而不必担心会破坏对方。松耦合通讯机制提供了紧耦合机制所没有的许多优点,并且有助于降低客户端和远程服务之间的依赖性。而与此相对,紧耦合意味着应用程序的不同组件之间的接口与其功能和结构是紧密相连的,当需要对部分或整个应用程序进行某种形式的更改时,它们就显得非常脆弱。

2. 软件工具

软件工具是指帮助软件开发的软件。它们对提高软件生产率,促进软件生产自动化具有重要作用。“工欲善其事,必先利其器”。设想在计算机上用C语言实现一个应用软件的过程。首先,要在编辑程序支持下把源程序输入计算机。然后对C语言进行编译,把源程序翻译成目标程序。如果发现错误,需要进行修改。编译通过后,再调用连接程序把所有通过编译的目标程序和与之有关的库程序连接起来,构成一个能在计算机上运行的可执行软件。在这里,编辑程序、编译程序、连接程序及支持它们的计算机操作系统,都属于软件工具。离开了这些工具,软件开发就失去了支持,会变得十分困难和低效,甚至不能进行。

以上提到的,仅是在编码阶段常用的一些软件工具。在其余阶段,如分析阶段、设计阶段和测试阶段,也研制了许多有效的工具。例如,Rational Rose和Power Designer可用于软件的分析 and 设计,LoadRunner可用于系统测试,Sourcesafe可用于版本控制。众多的工具组合起来,可以组成“工具箱”或“集成工具”,供开发人员在不同的阶段按需选用。

3. 软件工程环境

工具和方法,是软件开发技术的两大支柱,它们密切相关。当一种方法提出并证明有效后,往往随之研制出相应的工具,来帮助实现和推行这种方法。若将方法与工具结合起来,再加上配套的软、硬件支持就形成软件工程环境。

4. 管理学

在工业生产中,即使有先进的设备与技术,管理不善的企业也不能获得良好的经济效益。不少软件在生产中不能按质按时完成计划,管理混乱往往是其中的重要原因。可惜的是,软件管理尚未获得普遍的重视。

软件工程管理的目的是为了按照进度及预算完成软件开发计划,实现预期的经济和社会效益。它包括成本估算、进度安排、人员组织和质量保证等多方面的内容。

5. 经济学

软件工程经济学是软件工程与经济的交叉学科,是研究软件工程实践活动经济效益的学科。即以软件工程项目为主体,以技术/经济系统为核心,研究如何有效利用资源,提高经济效益的学科。软件工程经济学研究各种软件工程技术方案的经济效益,研究各种技术在使用过程中如何以最小的投入获得预期产出或者说如何以等量的投入获得最大产出;如何用最低的寿命周期成本实现软件产品的必要功能。

6. 度量

软件度量是对软件开发项目、过程及其产品进行数据定义、收集以及分析的持续性量化过程,目的在于对软件项目开发、产品等因素加以理解、预测、评估、控制和改善。没有软件度量,就不能实现对软件开发效果的定量分析。通过软件度量可以改进软件开发过程,促进项目成功,有助于开发高质量的软件产品。

软件的度量包含丰富的内涵,如顾客满意度度量、软件质量度量、项目进度度量、品牌资产度量及知识产权价值度量等。度量过程需要依靠事实、数据、原理、法则;其方法是测试、审核、调查;其工具是统计、图表、数字、模型;其标准是量化的指标。

1.3 CDIO 工程教育模式

CDIO 工程教育模式是近年来国际工程教育改革的最新成果。从 2000 年起,麻省理工学院和瑞典皇家工学院等四所大学组成的跨国研究获得 Knut and Alice Wallenberg 基金会近 2000 万美元巨额资助,经过 4 年的探索研究,创立了 CDIO 工程教育理念,并成立了以 CDIO 命名的国际合作组织。CDIO 代表构思(Conceive)、设计(Design)、实现(Implement)和运作(Operate)。它以产品研发到产品运行的生命周期为载体,让学生以主动的、实践的、课程之间有机联系的方式学习工程。CDIO 培养大纲将工程型毕业生的能力分为工程基础知识、个人能力、人际团队能力和工程系统能力四个层面。大纲要求以综合的培养方式使学生在这四个层面达到预定目标。

CDIO 的理念不仅继承和发展了欧美 20 多年来工程教育的理念,更重要的是系统地提出了具有可操作性的能力培养、全面实施以及检验测评的 12 条标准。2005 年,瑞典国家高教署(Swedish National Agency for Higher Education)采用这 12 条标准对本国 100 个工程学位计划进行评估,结果表明,新标准比原标准适应面更宽,更利于提高质量,尤为重要的是,新标准为工程教育的系统化发展提供了基础。迄今为止,已有几十所世界著名大学加入了 CDIO 组织,按 CDIO 模式培养的学生深受社会与企业欢迎。

1.3.1 CDIO 12 条标准

1) 标准 1——以 CDIO 为基本环境

讨论学校使命和专业目标在什么程度上反映了 CDIO 的理念,即把产品、过程或系统的构思、设计、实施和运行作为工程教育的环境。技术知识和能力的教学实践在多大程度上以产品、过程或系统的生产周期作为工程教育的框架或环境。

2) 标准 2——学习目标

从具体学习成果看基本个人能力、人际能力和对产品、过程和系统的构建能力在多大程度上满足专业目标,并经过专业利益相关者的检验。专业利益相关者是怎样参与学生必须达到的各种能力和水平标准的制定的。

3) 标准 3——一体化教学计划

个人能力、人际能力和对产品、过程和系统的构建能力是如何反映在培养计划中的。培养计划的设计在什么程度上做到了各学科之间相互支撑,并明确地将基本个人能力、人际能力和对产品、过程和系统构建能力的培养融于其中。

4) 标准 4——工程导论

个人能力、人际能力和对产品、过程和系统的构建能力是如何反映在培养计划中的。工程导论在多大的程度上激发了学生在相应核心工程领域的应用方面的兴趣和动力。

5) 标准 5——设计-实现经验

培养计划是否包含至少两个设计-实现经历(其中一个为基本水平,一个为高级水平)。在课内外活动中学生有多少机会参与产品、过程和系统的构思、设计、实施和运行。

6) 标准 6——工程实践场所

实践场所和其他学习环境怎样支持学生动手和直接经验的学习。学生有多大机会在现代工程软件和实验室内发展其从事产品、过程和系统建构的知识、能力和态度。实践场所是否以学生为中心、是否方便学生进入并易于交流。

7) 标准 7——综合性学习经验

综合性的学习经验能否帮助学生取得学科知识以及基本个人能力、人际能力和产品、过程和系统构建能力。综合性学习经验如何将学科学习和工程职业训练融合在一起。

8) 标准 8——主动学习

主动学习和经验学习方法怎样在 CDIO 环境下促进专业目标的达成。教和学的方法中在多大程度上基于学生自己的思考和解决问题的活动。

9) 标准 9——教师能力的提升

用于提升教师基本个人能力和人际能力以及产品、过程和系统构建能力的举措能得到怎样的支持和鼓励。

10) 标准 10——教师教学能力的提高

有哪些措施用来提高教师在一体化学习经验、运用主动和经验学习方法以及学生考核等方面的能力。

11) 标准 11——学生考核

学生的基本个人能力和人际能力,产品、过程和系统构建能力以及学科知识如何融入专业考核之中。这些考核如何度量和记录。学生在何种程度上达到专业目标。

12) 标准 12——专业评估

有无针对 CDIO12 条标准的系统化评估过程。评估结果在多大程度上反馈给学生、教师及其他利益相关者,以促进持续改进。专业教育有哪些效果和影响。

1.3.2 CDIO 在中国

我国高等工科教育的迫切任务是尽快培养与国际接轨的中国工程师。然而我国工科的教育实践还存在不少问题,如重理论轻实践、强调个人学术能力而忽视团队协作精神、重视知识学习