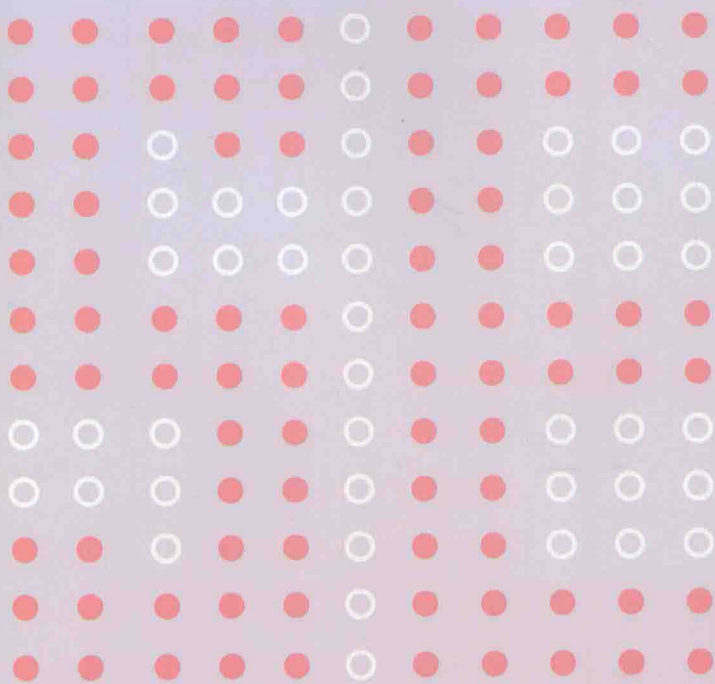


软件工程系列教材

王振武 编著

软件工程 理论与实践



<http://www.tup.com.cn>

Ruan Jian Gong Cheng Cheng

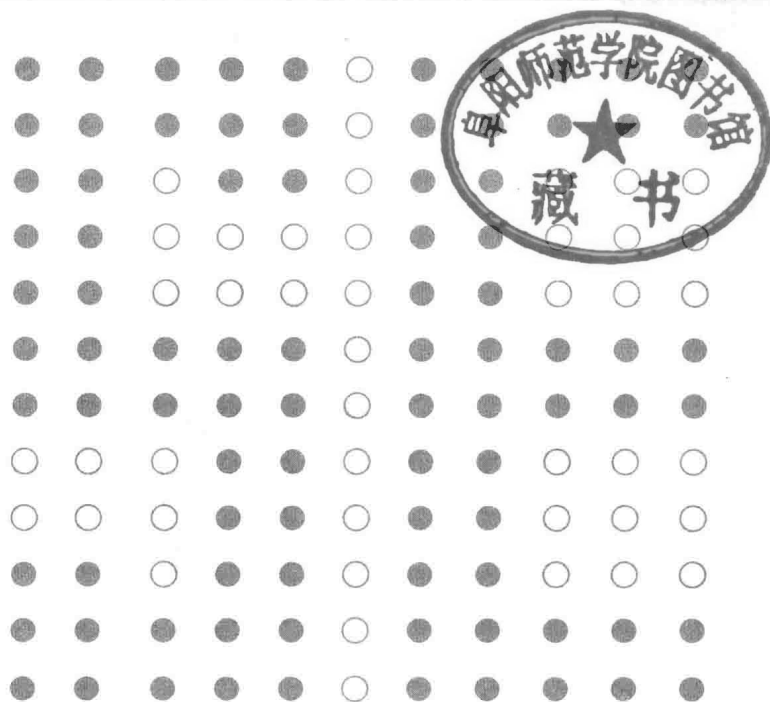
清华大学出版社



软件工程系列教材

软件工程 理论与实践

王振武 编著



清华大学出版社
北京

内 容 简 介

本书对软件工程的基本理论进行了系统的介绍,并且用一个具体的实例贯穿全书,对具体知识点也配有丰富的例题,这种理论与实践相结合的方式极大地方便了读者对抽象、枯燥的软件工程理论的理解和掌握。

本书共分 11 章,按照软件生命周期的流程组织各章内容,同时覆盖了结构化开发方法和面向对象开发方法,具体章节包括软件工程概述、可行性研究、需求分析、概要设计、详细设计、编码、测试、维护、软件项目管理、面向对象方法与 UML 建模以及面向对象分析与设计等内容。

本书可以作为高等院校软件工程课程的教材,也可以作为从事软件开发与测试工作以及其他相关工程技术工作的人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

软件工程理论与实践/王振武编著.--北京:清华大学出版社,2014

软件工程系列教材

ISBN 978-7-302-37378-0

I. ①软… II. ①王… 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(2014)第 163215 号

责任编辑:白立军 王冰飞

封面设计:何凤霞

责任校对:焦丽丽

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:21.25

字 数:493 千字

版 次:2014 年 11 月第 1 版

印 次:2014 年 11 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:059379-01

PREFACE

软
件
工
程
系
列
教
材

前 言

自 1968 年北大西洋公约组织提出“软件工程”概念以来, 软件工程的相关技术和方法得到了迅速的发展, 对软件设计、开发以及维护工作起到了重要的指导作用。为适应我国软件工程的的教学工作, 编者在多年软件开发以及软件工程教学实践的基础上, 参阅多种国内外最新版本的教材, 编写了本书。本书可以作为高等院校本科生的教材, 也可以为相关行业的工程技术人员提供有益的参考。

本书内容在安排上循序渐进, 对软件工程理论进行了通俗易懂的讲解。本书最大的特点是理论与实践相结合, 全书通过一个实例贯穿始终, 把软件工程的基本理论和方法系统、全面地讲解清楚, 这种方法克服了过去重理论轻实践的内容组织方式, 大大方便了读者的理解。具体而言, 本书 11 章内容之间的关系如下页图所示。

本书有配套的教学课件, 读者可从清华大学出版社网站 (www.tup.com.cn) 上下载。由于编者水平有限, 本书难免存在不少缺点和不足之处, 恳请专家和读者批评指正。



编者
2014年5月

CONTENTS

软
件
工
程
系
列
教
材

目 录

第 1 章 软件工程概述	1
1.1 软件	1
1.1.1 软件的定义	1
1.1.2 软件的特点	2
1.1.3 软件分类	2
1.2 软件危机与软件工程	5
1.2.1 软件危机	5
1.2.2 软件工程	6
1.2.3 软件工程的基本内容	7
1.2.4 软件工程的基本原则	7
1.3 软件生命周期与软件开发模型	8
1.3.1 软件生命周期	8
1.3.2 软件开发模型	9
1.4 软件开发方法	17
1.4.1 结构化开发方法	17
1.4.2 面向对象开发方法	20
1.5 CASE 工具与环境	22
1.5.1 Sybase PowerDesigner	22
1.5.2 Rational Rose	24
1.5.3 Microsoft Visio	24
1.6 软件工程的演变与发展	24
1.6.1 软件发展的新阶段和新问题	24
1.6.2 软件工程的发展	25
小结	25
习题	26

第2章 可行性研究	27
2.1 可行性研究的含义	27
2.2 可行性研究的任务和步骤	28
2.2.1 可行性研究的任务	28
2.2.2 可行性研究的步骤	29
2.3 可行性研究的要素	31
2.3.1 经济可行性	31
2.3.2 技术可行性	31
2.3.3 社会环境可行性	32
2.4 成本/效益分析	32
2.4.1 成本估计	32
2.4.2 费用估计	33
2.4.3 度量效益的方法	34
2.5 方案选择与决策	35
2.6 可行性分析实例——《学生教材购销系统》可行性研究报告	36
小结	45
习题	45
第3章 需求分析	47
3.1 需求分析的基本概念	47
3.1.1 软件需求的定义和特点	47
3.1.2 需求分析的目标和任务	48
3.1.3 需求分析的原则	50
3.2 需求获取的方法	50
3.2.1 用户访谈	50
3.2.2 问卷调查	50
3.2.3 专题讨论会	51
3.2.4 快速建立软件原型	52
3.3 需求建模方法	53
3.3.1 数据流图	53
3.3.2 数据字典	57
3.3.3 层次方框图	60
3.3.4 Warnier图	61
3.3.5 E-R图	61
3.3.6 状态转换图	64
3.4 需求验证	65
3.4.1 需求验证的内容	66

3.4.2	需求验证的方法	67
3.5	需求分析实例——《学生教材购销系统》需求规格说明书	67
3.5.1	《学生教材购销系统》非功能性需求	68
3.5.2	《学生教材购销系统》功能性需求	77
小结	86
习题	86
第4章	概要设计	87
4.1	概要设计概述	87
4.2	概要设计的主要内容	88
4.2.1	概要设计的任务和过程	88
4.2.2	软件体系结构设计	89
4.2.3	数据库设计	94
4.3	概要设计的原则	98
4.3.1	模块化	98
4.3.2	耦合	99
4.3.3	内聚	101
4.3.4	启发规则	103
4.3.5	面向对象设计模式	103
4.4	概要设计的工具	104
4.4.1	层次方框图	104
4.4.2	IPO图	104
4.4.3	HIPO图	105
4.5	概要设计实例——《学生教材购销系统》概要设计说明书	105
小结	113
习题	114
第5章	详细设计	116
5.1	详细设计的内容与原则	116
5.1.1	详细设计的内容	116
5.1.2	详细设计的原则	117
5.2	数据代码设计的工具	117
5.2.1	程序流程图	118
5.2.2	N-S图	119
5.2.3	PAD图	120
5.2.4	伪代码和PDL语言	122
5.2.5	判定表和判定树	123
5.2.6	详细设计工具的选择	125

5.3	人-机界面设计	125
5.4	程序结构复杂性的定量度量	127
5.4.1	McCabe 方法	127
5.4.2	Halstead 方法	130
5.5	详细设计实例——《学生教材购销系统》详细设计说明书	130
	小结	140
	习题	140
第 6 章	编码	143
6.1	程序设计语言	143
6.1.1	程序设计语言的分类	143
6.1.2	程序设计语言的特点	146
6.1.3	程序设计语言的选择	148
6.2	编程风格	150
6.2.1	良好编程风格的意义	150
6.2.2	编程风格的内容	151
6.3	程序效率	156
6.3.1	程序效率准则	156
6.3.2	算法对效率的影响	157
6.4	编程实例分析——《学生教材购销系统》编码规范说明	157
	小结	170
	习题	170
第 7 章	测试	172
7.1	软件测试概述	172
7.1.1	软件测试的定义	172
7.1.2	软件测试的对象	173
7.1.3	软件测试的准则	173
7.1.4	软件测试的方法	174
7.2	单元测试	175
7.3	集成测试	177
7.4	确认测试	179
7.5	白盒测试	180
7.6	黑盒测试	184
7.7	调试过程、技术与原则	187
7.7.1	调试过程	187
7.7.2	调试技术	188
7.7.3	调试原则	189

7.8 测试实例——《学生教材购销系统》测试报告	190
小结	195
习题	195
第8章 维护	197
8.1 软件维护概述	197
8.1.1 软件维护的分类	197
8.1.2 软件维护的特点	198
8.2 维护任务的实施	199
8.2.1 维护组织	200
8.2.2 维护报告	200
8.2.3 维护过程	201
8.3 软件的可维护性	201
8.3.1 软件可维护性的定义	201
8.3.2 影响软件可维护性的因素	201
8.3.3 提高软件可维护性的方法	202
8.4 软件维护的副作用	204
8.4.1 修改代码的副作用	204
8.4.2 修改数据的副作用	205
8.4.3 修改文档的副作用	205
8.5 软件再工程	205
8.6 维护报告实例——《学生教材购销系统》软件维护报告	206
小结	211
习题	211
第9章 软件项目管理	212
9.1 软件项目管理概述	212
9.2 软件进度计划管理	214
9.2.1 进度计划管理概述	214
9.2.2 进度计划编制方法	216
9.2.3 进度计划控制	218
9.3 软件质量管理	219
9.3.1 软件质量	219
9.3.2 软件质量保证措施	222
9.3.3 能力成熟度模型	223
9.4 软件成本管理	227
9.4.1 软件成本分析	227
9.4.2 软件成本估算	228

9.4.3	软件成本控制	229
9.5	配置管理	232
9.5.1	配置管理概述	232
9.5.2	配置管理的组织	232
9.5.3	配置管理的主要活动	235
9.6	人力资源管理	238
9.6.1	软件项目人力资源的特征	238
9.6.2	人力资源管理的主要内容	238
9.6.3	人员的组织与分工	239
9.7	项目管理实例——《学生教材购销系统》项目管理方案	242
	小结	251
	习题	252
第 10 章	面向对象方法与 UML 建模	254
10.1	面向对象方法概述	254
10.1.1	面向对象方法的含义	254
10.1.2	面向对象的软件工程	256
10.1.3	面向对象的基本概念和特征	260
10.2	统一建模语言	267
10.2.1	模型的建立	267
10.2.2	UML 的特点与应用	272
10.2.3	UML 提供的常用图	274
10.3	用例图	275
10.4	状态图	277
10.5	活动图	278
10.6	类图	280
10.7	对象图	286
10.8	顺序图	286
10.9	协作图	287
10.10	组件图	288
10.11	部署图	288
10.12	使用 PowerDesigner 进行建模	289
10.12.1	PowerDesigner 的安装	289
10.12.2	PowerDesigner 的功能	290
10.12.3	PowerDesigner 的运行界面	292
	小结	293
	习题	294

第 11 章 面向对象分析与设计	296
11.1 面向对象分析	296
11.1.1 面向对象分析的过程与原则	296
11.1.2 确定对象与类	301
11.1.3 确定属性	303
11.1.4 定义服务	304
11.1.5 对象间的通信	308
11.2 面向对象设计概述	314
11.2.1 面向对象设计的概念	314
11.2.2 面向对象设计的准则及启发规则	315
11.2.3 面向对象设计的方法	317
11.3 使用 PowerDesigner 画学生教材购销系统的 9 种图	319
11.3.1 学生教材购销系统的用例图	319
11.3.2 学生教材购销系统的类图	320
11.3.3 学生教材购销系统的对象图	321
11.3.4 学生教材购销系统的状态图	321
11.3.5 学生教材购销系统的活动图	323
11.3.6 学生教材购销系统的顺序图	323
11.3.7 学生教材购销系统的协作图	325
11.3.8 学生教材购销系统的组件图	325
11.3.9 学生教材购销系统的部署图	325
小结	326
习题	326
参考文献	328

软件工程概述

通过本章的学习,读者可以掌握软件的定义、特点以及分类,了解软件危机产生的原因以及人们为了最大限度地避免软件危机所采用的方法;了解 CASE(Computer Aided (或 Assisted)Software Engineering)工具,即计算机辅助工具的应用环境,提高编程速度和效率;掌握软件生命周期和软件开发方法,了解软件工程的演变和发展,为后续学习打下坚实的基础。

学习目标:

- 掌握软件的定义、特点以及分类
- 了解软件危机产生的原因
- 掌握软件工程的定义和基本内容
- 了解软件工程的基本原则
- 掌握软件生命周期和软件开发方法
- 了解 CASE 工具及其开发环境
- 了解软件工程的演变和发展

1.1 软件

1.1.1 软件的定义

尽管在不同的开发阶段,人们对软件的认识不尽相同,但目前对软件的定义已达成共识。软件的定义可以表述为:

- (1) 在运行中能提供所希望的功能和性能的指令集(即程序);
- (2) 使程序能够正确运行的数据结构;
- (3) 描述程序研制过程、方法所用的文档。

软件由两大部分组成,一是机器可执行的程序及有关数据;二是机器不可执行的与软件开发、运行、维护和使用相关的文档。

1.1.2 软件的特点

在计算机系统中,软件是一个逻辑部件,硬件是一个物理部件,因此在开发、维护方面,软件与硬件产品相比有着明显的特性。

(1) 软件是设计开发的,而不是传统意义上生产制造的。软件和硬件均可通过设计获得好的产品品质,但硬件制造阶段的质量问题是易于控制和纠正的,软件生产阶段的质量控制要困难得多。相对于硬件的生产,软件开发过度依赖于开发人员的素质、能力及协作,因此不能像管理硬件制造项目那样管理软件开发项目。

(2) 软件不会“磨损”。图 1-1 描述了以时间为变量的硬件的失效率。这个被称为“浴缸曲线”的关系图显示硬件在早期具有较高的失效率。纠正缺陷后,失效率降低,并在一段时间内保持平稳。然而,随着时间的推移,因为灰尘、振动、不当使用、温度等因素的影响,硬件的失效率会增大,硬件开始“磨损”,但软件不会受到引起硬件磨损的环境问题的影响。

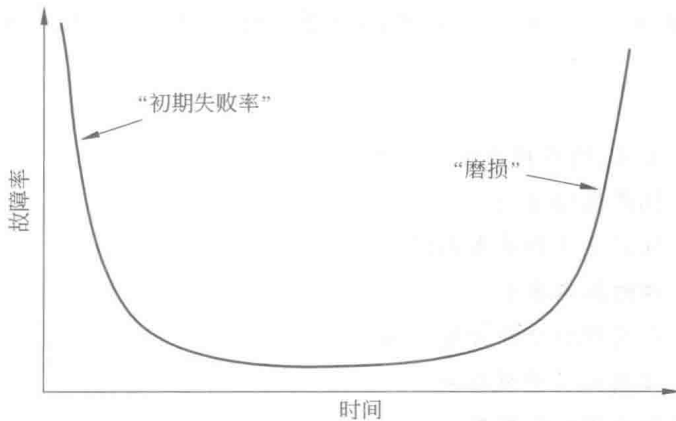


图 1-1 硬件的故障率随时间变化的曲线

从理论上讲,软件在交付初期失效率较高,随着缺陷的纠正,曲线将趋于平缓,即软件不会磨损,如图 1-2 所示。但是在实际使用过程中,未知的缺陷总是存在,而软件的每次变更都可能引入新的错误,使得失效率如图 1-2 所示的“实际曲线”那样陡然上升。在曲线回到最初的稳定状态前,新的变更会引起曲线的又一次上升。就这样,最小的失效率水平在逐渐上升,可以说,不断的变更是软件退化的根本原因。

(3) 大多数软件是根据客户的要求定制的。硬件通常是是根据一定的规范标准制定的,软件通常是是根据客户的要求定制的。虽然目前商业化软件的组件技术发展很快,但完全使用现有的组件实现软件系统仍然不现实,基于组件的软件开发模式仍然需要根据软件需求来开发系统。

1.1.3 软件的分类型

大家在工作和学习中会经常接触到各种各样的软件,至于这些数量众多的软件究竟

归为哪种类型,则需要考虑对计算机软件进行分类的依据。事实上,由于人们与软件的关系各不相同且所关心软件的侧重点也不相同,要给出计算机软件一个科学的、统一的严格分类标准是不现实的。但对软件的类型进行必要的划分对于根据不同类型的工程对象采用不同的开发和维护方法是很有价值的,因此有必要从不同角度对计算机软件做适当的分类。

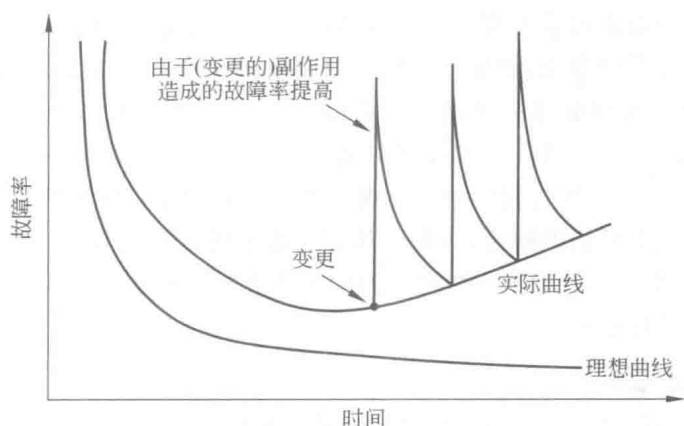


图 1-2 软件的失效曲线图

1. 基于软件的功能划分

(1) 系统软件。系统软件是与计算机硬件紧密结合以使计算机的各个部件与相关软件及数据协调、高效工作的软件,例如操作系统。系统软件在工作时频繁地与硬件交互,以便为用户服务,共享系统资源,在这中间伴随着复杂的进程管理和复杂的数据结构的处理。系统软件是计算机系统必不可少的重要组成部分。

(2) 支撑软件。它是协助用户开发软件的工具性软件,包括帮助程序人员开发软件产品的工具和帮助管理人员控制开发进程的工具,可划分为以下几种。

- 一般类型:包括文本编辑程序、文件格式化程序、程序库系统等。
- 支持需求分析:包括 PSL/PSA 问题描述语言、问题描述分析器、一致性检验程序等。
- 支持设计:包括图形软件包、结构化流程图绘图程序、设计分析程序、程序结构图编辑程序等。
- 支持实现:包括编译程序、交叉编译程序、预编译程序、连接编译程序等。
- 支持测试:包括静态分析程序、符号执行程序、模拟程序、测试覆盖检验程序等。
- 支持管理:包括 PERT 进度计划评审方法和绘图程序、标准检验程序和库管理程序。

(3) 应用软件。应用软件是在特定领域内开发的,为特定目的服务的一类软件。现在几乎所有的国民经济领域都使用了计算机,为这些计算机应用领域服务的应用软件种类繁多。其中,商业数据处理软件是占比例最大的一类,工程与科学计算软件大多属于数值计算问题。应用软件还包括计算机辅助设计/计算机辅助制造、系统仿真、智能产品嵌

入软件(如汽车油耗控制、仪表盘数字显示、刹车系统),以及人工智能软件(如专家系统、模式识别)等。此外,在事务管理、办公自动化、中文信息处理、计算机辅助教学等方面的软件也得到了迅速发展,产生了惊人的生产效率和巨大的经济效益。

2. 基于软件工作方式划分

(1) 实时处理软件。实时处理软件指在事件或数据产生时立即处理并及时反馈信号的软件,这类软件一般来讲需要监测和控制运行过程,主要包括数据采集、分析、输出三个部分,其处理时间应严格限定,如果在任何时间超出了这一限定,都将造成事故。

(2) 分时软件。分时软件指允许多个联机用户同时使用计算机,系统把处理机时间轮流分配给各联机用户,使各用户都感到只是自己在使用计算机的软件。

(3) 交互式软件。交互式软件指能实现人机通信的软件。这类软件接收用户给出的信息,但在时间上没有严格的限定,这种工作方式给予用户很大的灵活性。

(4) 批处理软件。批处理软件指把一组输入作业或一批数据以成批处理的方式一次运行,按顺序逐个处理的软件。

3. 基于软件规模划分

根据开发软件所需的人力、时间以及完成的源程序行数,可划分为下列 6 种不同规模的软件。

(1) 微型软件。微型软件指一个人在几天之内完成的、程序不超过 500 行语句且仅供个人专用的软件。通常,这类软件没有必要做严格的分析,也不必要有完整的设计、测试资料。

(2) 小型软件。小型软件指一个人半年之内完成的 2000 行以内的程序。这种程序通常没有与其他程序的接口,但需要按一定的标准化技术、正规的资料书写以及进行定期的系统审查,只是没有大项目那样严格。

(3) 中型软件。中型软件指 5 个人以内在一年多时间里完成的 5000~50 000 行的程序。中型软件开始出现了软件人员之间、软件人员与用户之间的联系、协调配合关系的问题,因而计划、资料书写以及技术审查需要比较严格地进行。在开发中使用系统的软件工程方法是完全必要的,这对提高软件产品质量和程序人员的工作效率起着重要的作用。

(4) 大型软件。大型软件指 5~10 个人在两年多的时间里完成的 50 000~100 000 行的程序。参加工作的软件人员需要按级管理,在任务完成过程中,人员调整往往不可避免,因此会出现对新手的培训和逐步熟悉工作的问题。对于这样规模的软件,采用统一的标准,实行严格的审查是绝对必要的。由于软件的规模庞大以及问题的复杂性,大型软件往往在开发的过程中会出现一些事先难以做出估计的不测事件。

(5) 甚大型软件。甚大型软件指 100~1000 人参加,用 4~5 年时间完成的具有 100 万行程序的软件。这种甚大型项目可能会划分成若干个子项目,每一个子项目都是一个大型软件,子项目之间具有复杂的接口。例如,实时处理系统、远程通信系统、多任务系统、大型操作系统、大型数据库管理系统通常有这样的规模。很显然,如果这类问题没有软件工程方法的支持,它的开发工作是不可想象的。

(6) 极大型软件。极大型软件指 2000~5000 人参加,10 年内完成的 1000 万行以内

的程序。这类软件很少见,往往是军事指挥、弹道导弹防御系统等。

可以看出,对于规模大、时间长、很多人参加的软件项目,其开发工作必须要有软件工程的知识做指导,而规模小、时间短、参加人员少的软件项目也要用到软件工程概念,并遵循一定的开发规范,其基本原则是一样的。

4. 基于软件失效的影响进行划分

工作在不同领域的软件,在运行中对可靠性有不同的要求。事实上,随着计算机进入国民经济等各个重要领域,其软件的可靠性越来越显得重要。人们一般称这类软件为关键软件,其特点在于:

- (1) 可靠性质量要求高;
- (2) 常与完成重要功能的大系统的处理部件相关联;
- (3) 含有的程序可能对人员、公众、设备或设施的安全造成影响,还可能影响到环境的质量和关系到国家的安全与机密。

5. 基于软件服务对象的范围划分

软件工程项目完成后可以有两种情况提供给用户:

(1) 定制软件。定制软件是受某个特定客户(或少数客户)的委托,由一个或多个软件开发机构在合同的约束下开发出来的软件。

(2) 产品软件。产品软件是由软件开发机构开发出来直接提供给市场,或是为千百个用户服务的软件。

1.2 软件危机与软件工程

1.2.1 软件危机

在20世纪60年代末“软件工程”概念提出之前,由于计算机硬件资源的限制(如运算速度、内存容量),程序员不得不利用个人的编程技巧来提高代码效率,使得程序的可理解性较差。规模小的程序尚能应付测试和运行维护,随着计算机硬件和软件技术的发展,软件的规模也越来越大,一个软件项目可能有几万、几十万甚至几百万行代码,软件的开发和维护就出现了很严重的问题,在软件和软件开发技术不能很好地满足软件产品开发要求时,软件危机(Software Crisis)就爆发了。软件危机指的是软件开发和维护过程中遇到的一系列严重问题。主要表现如下:

(1) 产品不符合用户的实际需要。因为软件开发人员对用户需求没有深入、准确的了解,甚至对所要解决的问题还没有正确认识就着手编写程序,而且软件开发人员和用户之间的信息交流往往很不充分,导致用户对软件产品不满意的现象发生。

(2) 软件开发生产率提高的速度远远不能满足客观需要,软件的生产率远远低于硬件生产率和计算机应用的增长速度,使人们不能充分利用现代计算机硬件提供的巨大潜力。

(3) 软件产品的质量差。软件可靠性和质量保证的定量概念出现不久,软件质量保