

全国计算机技术与软件专业技术资格（水平）考试参考用书

系统分析师技术指南

希赛IT教育研发中心 组编 张友生 主编
全国计算机技术与软件专业技术资格（水平）考试办公室推荐

清华大学出版社



(2007版)

全国计算机技术与软件专业技术资格（水平）考试参考用书

系统分析师技术指南

（2007版）

希赛IT教育研发中心 组编 张友生 主编
全国计算机技术与软件专业技术资格（水平）考试办公室推荐

清华大学出版社
北京

内 容 简 介

本书对前沿而又成熟的系统分析技术和方法进行了讨论,包括 CMM 与过程改进、J2EE 与 .NET 平台、中间件及相关技术、应用服务器、Web 服务、数据仓库与数据挖掘、操作数据存储、异构数据库的集成、企业应用集成、XML、软件构架、设计模式、UML、RUP、SOAP、ERP/SCM/CRM、工作流、软件产品线、XP 与敏捷编程、实时系统和嵌入式系统、网格计算与普适计算等。这些技术和方法是合格的系统分析师必须具备的知识,也是系统分析师考试必考的知识点。阅读本书,犹如进入 IT 新技术和新方法的殿堂。

本书是计算机技术与软件专业技术资格(水平)考试参考用书,同时也可作为系统分析师日常工作的参考手册;既是软件设计师、数据库系统工程师、网络工程师进一步深造和发展的必读书籍,也是计算机专业教师的教学和工作参考书。

本书扉页为防伪页,封面贴有清华大学出版社防伪标签,无上述标识者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

系统分析师技术指南:2007 版/张友生主编. —北京:清华大学出版社,2007.1

ISBN 978-7-302-14379-6

I. 系… II. 张… III. 软件工程—系统分析—工程技术人员—资格考核—自学参考资料 IV. TP311.5

中国版本图书馆 CIP 数据核字(2006)第 162156 号

责任编辑:冯志强

责任校对:张 剑

责任印制:李红英

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:清华大学印刷厂

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185×230 印 张:29.5 防伪页:1 字 数:645 千字

版 次:2007 年 1 月第 1 版 印 次:2007 年 1 月第 1 次印刷

印 数:1~5000

定 价:45.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:023793-01

前 言

系统分析是 IT 业组织开发优秀的应用系统的重要工作,需要拥有扎实的理论知识和丰富的实际经验的人员来完成。随着应用系统规模越来越大,复杂程度越来越高,系统分析师在系统开发的过程中,发挥着越来越重要的作用。

全国计算机技术与软件专业技术资格(水平)考试是培养和选拔计算机技术和软件专业人才的一个考试,其难度较大。主要原因是考试范围十分广泛,涉及计算机专业的每门课程,还要有数学、外语、系统工程、信息化和知识产权等知识,且注重考查新技术和新方法的应用。考试不但注重广度,而且还有一定的深度。特别是高级资格考试(系统分析师),不但要求考生具有扎实的理论知识,还要具有丰富的实践经验。

众所周知,IT 技术日新月异,作为一名合格的系统分析师,必须善于学习,及时了解和掌握新技术和新方法。本书对当前比较前沿而又成熟的系统分析技术和方法进行了讨论,这些技术和方法是任何一位合格的系统分析师必须具备的知识,也是系统分析师考试必考的知识点。阅读本书,犹如进入 IT 新技术和新方法的殿堂。

本书作为全国计算机技术与软件专业技术资格(水平)考试参考用书,可作为系统分析师日常工作的参考手册,也可作为软件设计师、数据库系统工程师、网络工程师进一步深造和发展的参考用书,计算机专业教师的教学和工作的参考书。

本书由希赛 IT 教育研发中心组稿,张友生主编,徐锋、田俊国、施游、李雄参加了审稿和修改工作。

全书共分 22 章。

第 1 章由张友生编写。主要介绍 CMM/CMML,以及软件企业如何实施 CMM,CMM 与 TSP、PSP 的联系及三者之间的结合。

第 2 章由戎橄编写。介绍了 J2EE 与 .NET 平台,并从多个方面对这两个平台进行了比较分析。

第 3 章由徐雷明编写。详细讨论了中间件及相关技术,中间件在电子商务中的应用,以及中间件的发展趋势和相关的标准和协议。

第 4 章由戎橄编写。介绍应用服务器的基本概念,并对当前流行的几种应用服务器产品进行了比较分析,以及应用服务器的选择方法。

第 5 章由戎橄编写。对 Web 服务及其应用进行了介绍,讨论了 Web 服务与 XML 的结合。

第 6 章由高艳明编写。对数据仓库的概念和体系结构、涉及的技术、数据仓库的设计与

开发进行了较深入的讨论,介绍了数据仓库的典型应用领域。

第 7 章由高艳明编写。首先介绍数据挖掘的概念、功能,然后讨论数据挖掘的常用技术、工具和流程,Web 数据挖掘与 XML 的关系,最后介绍数据挖掘的典型应用。

第 8 章由周峻松编写。介绍了操作数据存储、讨论了操作数据存储和数据仓库的区别,操作数据存储的应用,重点讨论操作数据存储系统的设计。

第 9 章由周峻松编写。讨论了异构数据库的集成,包括异构数据库系统的背景、体系结构和异构数据库的互联问题。

第 10 章由黄以宽编写。从表示集成、数据集成和功能集成等几个方面介绍企业应用集成,讨论了企业应用集成的标准,对不同技术提供商的企业应用集成解决方案进行了比较分析,最后,讲述企业实施应用集成的步骤。

第 11 章由黄以宽编写。介绍可扩展标记语言的背景和特点及发展展望,较详细地分析了可扩展标记语言的一些功能和方法,以及可扩展标记语言的编程接口。

第 12 章由张友生编写。对软件构架的概念、建模、风格和特定领域软件构架进行了讨论,详细介绍了基于构架的软件开发模型和软件构架评估的方法。

第 13 章由张友生和施游编写。介绍了设计模式的概念、组成和分类,讨论了模式和软件构架的关系,MVC 构架的设计与实现。

第 14 章由杨飞编写。详细讨论了统一建模语言,重点介绍了统一建模语言的 10 种图形及其应用领域。

第 15 章由张友生编写。文中对统一开发过程进行了详细介绍。

第 16 章由黄以宽编写。介绍了简单对象访问协议及其与 Web 服务的关系,详细讨论了简单对象访问协议的使用及其安全性。

第 17 章由马映冰和张剑鲲编写。对 ERP/SCM/CRM 所涉及的概念和理论进行了介绍,重点讨论了 ERP 系统的应用。

第 18 章由田俊国编写。介绍了工作流的定义、特征,工作流自动化的发展与应用现状,以及工作流与 ERP 系统的应用。

第 19 章由张友生编写。内容包括软件产品线的概念、过程模型、组织结构、建立方式框架和应用框架技术、软件产品线的基本活动、软件产品线构架的设计以及软件产品线构架的演化。

第 20 章由徐锋编写。主要讨论 XP 与敏捷编程,首先简单地介绍敏捷方法论,然后从极限编程的价值观、原则和最佳实践进行了较深入的讨论。

第 21 章由陈建忠编写。主要介绍实时系统和嵌入式系统的基本知识,实时系统的研究现状和发展,嵌入式实时操作系统特征。

第 22 章由施游编写。对网格计算与普适计算的概念、特点、组成及应用领域进行了详细的介绍,讨论了网格计算研究现状和普适计算的关键问题。

本书在编写的过程中参考了许多相关的资料和书籍,在此恕不一一列举(详见各章的参

参考文献列表),编者在此对这些参考文献的作者表示真诚的感谢。同时感谢清华大学出版社在本书的出版过程中给予的支持和帮助。

由于编者水平有限,且本书涉及的知识点多、新,书中难免有不妥和错误之处,编者诚恳地期望各位专家和读者不吝赐教和帮助,对此,我们将深为感谢。

有关本书的意见反馈和咨询,可在希赛网社区(<http://bbs.csai.cn>)“书评在线版块”中的“希赛 IT 教育研发中心”栏目上与作者进行交流。

编 者

2006 年 11 月

目 录

第 1 章 CMM 与软件过程改进	1	主要参考文献	55
1.1 CMM 综述	1	第 3 章 中间件技术	56
1.1.1 CMM 基本概念	2	3.1 中间件概述	56
1.1.2 CMM 的基本框架	4	3.2 中间件的定义	56
1.1.3 软件过程评估和软件能力评价	7	3.3 中间件的功能和分类	57
1.2 软件企业如何实施 CMM	8	3.3.1 中间件的基本分类	58
1.3 CMM 存在的问题	11	3.3.2 中间件的详细分类	58
1.4 ISO 9001 与 CMM 的比较	12	3.4 中间件的优点及其应用	60
1.5 CMMI 综述	14	3.4.1 中间件应用层次	61
1.6 CMM 与 CMMI 的比较	17	3.4.2 中间件技术在集成中的应用	61
1.7 个体软件过程	18	3.5 J2EE 中间件实现	62
1.8 群组软件过程	20	J2EE 应用服务器中间件的体系结构	63
1.9 CMM/TSP/PSP 三者的结合	21	3.6 中间件与电子商务	64
主要参考文献	23	3.6.1 电子商务中间件构架	64
第 2 章 J2EE 与 .NET 平台	24	3.6.2 电子商务应用服务器	65
2.1 J2EE 平台简介	24	3.6.3 通信平台	66
2.1.1 分布式的多层应用程序	24	3.6.4 安全平台	66
2.1.2 J2EE 组件	25	3.7 构件技术与中间件	67
2.1.3 J2EE 容器	28	3.8 中间件与 Web 服务	70
2.1.4 J2EE 的部署	29	从中间件到 Web 服务	71
2.1.5 J2EE 标准	30	3.9 中间件的发展趋势	74
2.1.6 Java EE	32	3.10 相关技术标准与协议	75
2.2 .NET 平台简介	34	主要参考文献	76
2.2.1 .NET 平台概述	34	第 4 章 应用服务器	77
2.2.2 .NET Framework	35	4.1 什么是应用服务器	77
2.2.3 通用语言运行时 CLR	36	4.1.1 两层计算模式与多层计算模式的区别	77
2.3 J2EE 和 .NET 平台的异同	45	4.1.2 应用服务器的作用	78
2.3.1 JVM 与 CLR	46	4.1.3 应用服务器的用途	79
2.3.2 开发语言的选择	46	4.1.4 应用服务器的类型	81
2.3.3 垃圾收集	47	4.1.5 应用服务器的成本	83
2.3.4 对企业计算的支持	50		
2.3.5 安全性	53		
2.3.6 小结	54		

4.1.6 应用服务器发展过程 与方向	83	6.2.3 数据仓库的体系结构	126
4.2 应用服务器产品比较	84	6.3 数据仓库的设计与开发	130
4.2.1 BEA WebLogic	85	6.3.1 数据仓库的设计	130
4.2.2 IBM WebSphere Application Server	85	6.3.2 数据仓库的开发、实施	130
4.2.3 Microsoft Transaction Server	86	6.4 数据仓库技术	132
4.2.4 Oracle Application Server ...	86	6.4.1 数据管理技术	132
4.2.5 SilverStream	87	6.4.2 数据存储技术	133
4.2.6 Sybase Enterprise Application Server	87	6.4.3 数据仓库接口技术	134
4.2.7 应用服务器的选择	88	6.5 数据仓库厂家产品	134
主要参考文献	89	6.6 一些工具软件	138
第 5 章 Web 服务及应用	90	6.7 数据仓库的典型应用	140
5.1 Web 服务概述	90	主要参考文献	141
5.1.1 什么是 Web 服务	90	第 7 章 数据挖掘技术	142
5.1.2 Web 服务模型	91	7.1 数据挖掘技术概述	142
5.1.3 Web 服务使用流程	92	7.1.1 数据挖掘的定义	142
5.1.4 Web 服务协议堆栈	93	7.1.2 数据挖掘的发展	143
5.1.5 XML 在 Web 服务中 的应用	97	7.1.3 数据挖掘的功能	145
5.1.6 Web 服务与分布式计算	97	7.1.4 数据挖掘常用技术	146
5.2 Web 服务中的常用协议	98	7.2 数据挖掘工具	148
5.2.1 WSDL	99	7.2.1 数据挖掘工具简介	148
5.2.2 UDDI	109	7.2.2 数据挖掘工具的 评价标准	148
5.3 构造一个简单的 Web 服务	115	7.3 数据挖掘的流程	150
主要参考文献	117	7.4 Web 数据挖掘与 XML	151
第 6 章 数据仓库技术	118	7.4.1 Web 数据挖掘的难点	151
6.1 数据仓库概论	118	7.4.2 XML 与 Web 数据挖掘	152
6.1.1 数据仓库的产生	118	7.5 数据挖掘热点	153
6.1.2 数据仓库的概念	119	7.6 数据挖掘应用	154
6.1.3 与数据仓库技术相关的 其他概念	121	7.6.1 数据挖掘在市场营销 的应用	155
6.1.4 数据仓库与传统数据库 的区别	121	7.6.2 成功案例	156
6.2 数据仓库的结构	122	主要参考文献	158
6.2.1 数据仓库的概念结构	122	第 8 章 操作数据存储	159
6.2.2 数据仓库的参考框架	123	8.1 操作数据存储介绍	159
		8.2 商业问题	159
		8.3 什么是操作数据存储	161
		8.3.1 ODS 的定义	161
		8.3.2 ODS 的特点	161
		8.3.3 ODS 的分类	162

8.3.4 ODS 和 DW 的联系 与区别	162	10.4.1 IBM	198
8.3.5 从 DB 向 ODS 转化的 实现机制	165	10.4.2 BEA	199
8.4 操作数据存储的应用	166	10.4.3 Microsoft	199
8.5 操作数据存储系统的设计	167	10.4.4 Sybase	200
8.5.1 ODS 数据转换层	167	10.5 实施 EAI	200
8.5.2 ODS 平台特性	168	10.5.1 EAI 实施要点	200
8.5.3 ODS 系统中间件	169	10.5.2 企业实施 EAI 的 4 个 阶段	201
8.5.4 ODS 系统数据建模	169	主要参考文献	202
8.5.5 ODS 系统设计步骤	171		
主要参考文献	172		
第 9 章 异构数据库的集成	174	第 11 章 可扩展标记语言	203
9.1 异构数据库系统的背景	174	11.1 什么是 XML	203
9.2 异构数据库体系结构	175	11.1.1 XML 产生的背景	203
9.3 异构数据库互联	178	11.1.2 XML 的特点	204
主要参考文献	186	11.1.3 XML 带来的好处	206
		11.1.4 XML 应用分类	208
		11.1.5 XML 的意义	209
		11.1.6 XML 发展展望	210
第 10 章 企业应用集成	188	11.2 解析 XML	212
10.1 企业应用集成 EAI	188	11.2.1 自描述数据	212
10.1.1 谁需要 EAI	189	11.2.2 XML 与 HTML 的区别	213
10.1.2 EAI 的内容	189	11.2.3 组织良好的 XML	214
10.1.3 EAI 的技术基础	190	11.2.4 有效的 XML 与 DTD 和 XML 模式	215
10.1.4 EAI 给企业带来 的好处	191	11.2.5 CSS 与 XSL	217
10.1.5 EAI 发展现状	192	11.3 XML 编程接口	220
10.2 EAI 集成模型	192	11.3.1 支持 XML 的开发工具	222
10.2.1 什么是集成模型	193	11.3.2 用脚本语言处理 DOM	223
10.2.2 表示集成	193	11.3.3 XML 的建模	224
10.2.3 数据集成	194	主要参考文献	225
10.2.4 功能集成	194		
10.3 EAI 与标准化	195	第 12 章 软件架构	226
10.3.1 标准的层次	195	12.1 软件架构概述	226
10.3.2 应用标准的重要性	196	12.2 软件架构建模	228
10.3.3 英国电子政务中的应用 标准介绍	197	12.2.1 逻辑视图	228
10.3.4 信息孤岛原因分析及 EAI 对策	197	12.2.2 开发视图	230
10.4 不同技术提供商的 EAI 解决方案	198	12.2.3 进程视图	231
		12.2.4 物理视图	232
		12.2.5 场景	234
		12.3 软件架构风格	234
		12.3.1 分层系统	235

12.3.2 C2 风格	236	第 14 章 统一建模语言	288
12.3.3 客户/服务器风格	237	14.1 UML 概述	288
12.3.4 三层 C/S 结构风格	238	14.1.1 UML 是什么	288
12.3.5 浏览器/服务器风格	241	14.1.2 UML 的发展历史	289
12.3.6 公共对象请求代理架构	242	14.1.3 UML 结构	290
12.3.7 异构结构风格	245	14.1.4 UML 的主要特点	291
12.4 特定领域软件架构	245	14.1.5 UML 的应用领域	292
12.4.1 DSSA 的活动	246	14.2 用例图	292
12.4.2 DSSA 的建立过程	247	14.3 类图和对象图	295
12.5 面向服务的架构	248	14.4 交互图	298
12.5.1 SOA 的概念	248	14.4.1 顺序图	299
12.5.2 SOA 的特征	250	14.4.2 协作图	300
12.5.3 SOA 的优点和缺点	253	14.5 状态图	300
12.5.4 SOA 的生命周期	255	14.6 活动图	301
12.5.5 SOA 与其他技术的关系	256	14.7 构件图	303
12.6 富互联网应用架构	258	14.8 部署图	304
12.6.1 RIA 的概念	258	主要参考文献	305
12.6.2 RIA 模型	260	第 15 章 统一开发过程	306
12.6.3 RIA 的技术	262	15.1 RUP 概述	306
12.7 基于架构的软件开发模型	264	15.2 RUP 生命周期	307
12.7.1 架构需求	264	15.2.1 初始阶段	308
12.7.2 架构设计	265	15.2.2 细化阶段	309
12.7.3 架构文档化	266	15.2.3 构建阶段	311
12.7.4 架构复审	267	15.2.4 交付阶段	311
12.7.5 架构实现	267	15.2.5 技术评审	312
12.7.6 架构演化	267	15.3 统一开发过程项目管理	313
12.8 软件架构评估	269	主要参考文献	316
主要参考文献	271	第 16 章 简单对象访问协议	317
第 13 章 设计模式	273	16.1 SOAP 概述	317
13.1 设计模式概述	273	16.1.1 Web 服务与 SOAP	
13.2 设计模式的组成	276	的关系	317
13.2.1 设计模式的基本成分	276	16.1.2 SOAP 客户端和服务器	318
13.2.2 设计模式的描述	277	16.1.3 SOAP 与 CORBA、COM/DCOM	
13.3 模式和软件构架	279	的比较	318
13.4 设计模式方法分类	280	16.1.4 SOAP=RPC+HTTP	
13.5 MVC 构架的设计与实现	284	+XML	319
13.5.1 MVC 构架	284	16.2 解析 SOAP	320
13.5.2 MVC 的设计与实现	285	16.2.1 SOAP 的消息封装	320
主要参考文献	287	16.2.2 SOAP 的编码规则	321

16.3 SOAP 应用	324	应用现状	374
16.3.1 在 HTTP 中使用 SOAP	324	18.1.4 工作流和传统管理软件	375
16.3.2 在 RPC 中使用 SOAP	326	18.1.5 工作流和业务流程重构	376
16.4 SOAP 的安全性	328	18.1.6 工作流系统实现	378
16.4.1 Internet 的安全要求 与隐患	328	18.2 工作流与 ERP 系统的应用	383
16.4.2 SOAP 安全解决方案 简介	329	18.2.1 以工作流实现 ERP 和 OA 集成	383
16.4.3 SOAP 安全扩展:数字 签名	331	18.2.2 集成方案介绍	384
主要参考文献	333	18.2.3 发展前景	385
第 17 章 企业信息化	334	主要参考文献	386
17.1 诠释企业信息化	334	第 19 章 软件产品线	387
17.1.1 技术信息化	335	19.1 软件产品线概述	387
17.1.2 管理信息化	338	19.2 软件产品线的过程模型	388
17.2 企业资源计划	343	19.3 软件产品线的组织结构	391
17.3 供应链管理	345	19.4 软件产品线的建立方式	393
17.3.1 供应链的概念	346	19.5 框架和应用框架技术	394
17.3.2 供应链管理的概念	348	19.6 软件产品线基本活动	397
17.3.3 供应链管理系统	350	19.6.1 产品线分析	398
17.4 财务管理	357	19.6.2 产品开发	400
17.4.1 财务管理软件的发展	357	19.7 软件产品线构架的设计	401
17.4.2 财务管理软件的功能	359	19.7.1 产品线构架简介	401
17.5 客户关系管理	363	19.7.2 产品线构架的标准化 和定制	403
17.5.1 客户关系模型	363	19.8 软件产品线构架的演化	404
17.5.2 CRM 的功能	366	19.8.1 背景介绍	405
17.6 信息化项目实施的风险和控制	368	19.8.2 两代产品的各种发行 版本	407
17.6.1 来自人的风险的规避	368	主要参考文献	411
17.6.2 来自流程的风险和 规避	370	第 20 章 XP 与敏捷编程	413
17.6.3 来自项目管理的风险 和规避	371	20.1 敏捷方法论简介	413
17.6.4 来自数据的风险和规避	372	20.1.1 敏捷联盟	413
主要参考文献	372	20.1.2 敏捷原则	415
第 18 章 工作流	373	20.1.3 主要的敏捷方法	416
18.1 工作流概述	373	20.2 极限编程简介	426
18.1.1 工作流定义	373	20.2.1 解析 XP	426
18.1.2 工作流特征	373	20.2.2 XP 的价值观	427
18.1.3 工作流自动化的发展与		20.2.3 XP 的原则	429
		20.2.4 XP 的 12 个最佳实践	430
		20.2.5 XP 的实践阶段	437

主要参考文献	438	22.2.1 网格的系统结构	450
第 21 章 实时系统和嵌入式系统	440	22.2.2 网格计算的环境 Globus	453
21.1 实时系统和嵌入式系统定义	440	22.3 网格计算研究现状	454
21.2 实时系统的基本特征	442	22.3.1 研究现状	454
21.3 实时系统的研究现状和发展	443	22.3.2 网格计算存在的问题	455
21.4 嵌入式实时操作系统特征	444	22.4 普适计算简介	456
主要参考文献	445	22.4.1 普适计算系统简介	456
第 22 章 网格计算与普适计算	447	22.4.2 普适计算系统实现	458
22.1 网格计算系统简介	447	22.4.3 普适计算系统组成	458
22.1.1 网格计算的定义	447	22.4.4 普适计算的特性	459
22.1.2 网格系统的特点	448	22.4.5 普适计算的关键问题	460
22.1.3 网格技术应用的领域	449	主要参考文献	461
22.2 网格计算的组成	450		

第 1 章 CMM 与软件过程改进

软件过程(software process)是人们建立、维护和演化软件产品整个过程中所有技术活动和管理活动的集合。目前,软件过程技术是一个非常活跃的研究领域,吸引了大批来自学术界和工业界的专家和学者。从 1984 年起每年都有软件过程国际研讨会(ISPW),从 1991 年起开始召开软件过程国际会议(ICSP),很多国家都有自己的软件过程改进网络(SPN)。软件过程技术的研究主要有三个方向:

- **软件过程分析和建模** 软件过程建模方法是软件过程技术的起点,其中形式化、半形式化建模方法有基于规则的方法和基于过程程序的方法等。过程分析和过程建模对于保证过程定义的质量、建立全面和灵活的过程体系具有重要的作用。软件过程的建模主要使用过程建模语言(Process Modeling Languages, PML)。PML 最基本的功能是用来描述和定义过程,建立过程模型。PML 的能力和表达方式直接影响着过程模型的质量和建模效率。所以,选择合适的 PML,成为过程分析、过程建模和选择建模工具的关键。
- **软件过程支持** 软件过程支持主要是指研究和开发支持软件过程活动的 CASE 工具。过程支持工具作为一种技术基础设施能够很好地支持、管理、规范软件过程。它的使用将使软件过程的透明度好,为项目的软件过程提供指导,使得开发者和管理者都有据可依,便于更有效地管理软件过程。软件过程支持工具主要包括软件过程流程工具、过程文档工具、评审工具和人员管理工具。
- **软件过程评估和改进** 软件过程评估和改进是指根据某种模型对现有软件过程进行考核和评价,找出其中的不足之处,然后加以改进。软件过程改进对生产高质量软件产品和提高软件生产率的重要性已被越来越多的软件开发组织所认同。由美国卡耐基·梅隆大学软件工程研究所(CMU/SEI)提出的软件能力成熟度模型(SW-CMM)除了用于软件过程评估外,还向软件组织提供了指导其进行软件过程管理和软件过程改进的框架。软件过程改进的基本原则是采用过去项目中成功的实践经验,因此,理解、记录和重用部分软件过程是软件过程改进研究的一个重要方向。

1.1 CMM 综述

CMM 是软件过程能力成熟度模型(Capacity Maturity Model)的简称,是 SEI 为了满足

美国联邦政府评估软件供应商能力的要求,于 1986 年开始研究的模型,并于 1991 年正式推出了 CMM 1.0 版。CMM 自问世以来备受关注,在一些发达国家和地区得到了广泛应用,成为衡量软件企业软件开发和管理水平的重要参考因素,并成为软件过程改进事实上的工业标准。据了解,美国、印度、日本等国家已有数十家公司通过了 CMM 不同等级的认证。中国政府自 2000 年加强对软件企业的重视,大力推荐 CMM 以来,已经有五十多家企业先后通过了 CMM 各种级别的认证。

1992 年 4 月,SEI 举行了一个 CMM 的研讨会,参加研讨会的有大约 200 名富有经验的软件专家。SEI 在广泛听取他们的意见之后,又于 1993 年推出 CMM 1.1 版。这也是目前世界上比较流行和通用的 CMM 版本。

十几年来,此项工作一直在不断进行。按照 SEI 原来的计划,CMM 的改进版本 2.0 应该在 1997 年 11 月完成,然后在取得版本 2.0 的实践反馈意见之后,在 1999 年完成准 CMM 2.0 版本。但是,美国国防部办公室要求 SEI 推迟发布 CMM 2.0 版本,而要先完成一个更为紧迫的项目 CMMI。

1.1.1 CMM 基本概念

本节介绍 CMM 中用到的有关概念和术语。

- **过程(process)** 为实现既定目标的一系列操作步骤。
- **软件过程(software process)** 指人们用于开发和维护软件及其相关产品的一系列活动、方法、实践和革新。其中相关产品是指项目计划、设计文档、编码、测试和用户手册。当一个企业逐步走向成熟时,软件过程的定义也会日趋完善,其企业内部的过程实施将更具有一致性。
- **软件过程能力(software process capability)** 描述在遵循一个软件过程后能够得到的预期结果的界限范围。该指标是对能力的一种衡量,用它可以预测一个组织在承接下一个软件项目时,所能期望得到的最可能的结果。
- **软件过程性能(software process performance)** 表示遵循一个软件过程后所得到的实际结果。软件过程性能与软件过程能力有区别,软件过程性能关注的是实际得到的结果,而软件过程能力关注的是期望得到的结果。由于项目要求和客观环境的差异,软件过程性能不可能充分反应软件过程整体能力,即软件过程性能受限于它的环境。
- **软件过程成熟度(software process maturity)** 是指一个具体的软件过程被明确地定义、管理、评价、控制和产生实效的程度。所谓成熟度既包含能力的一种增长潜力,同时也表明了组织实施软件过程的实际水平。随着组织软件过程成熟度能力的不断提高,组织内部通过对过程的规范化和对成员的技术培训,软件过程也会被它的使用者关注和不断修改完善,从而使软件的质量、生产率和生产周期得到

改善。

- ❑ **关键过程(区)域(key process area)** 是指一系列相互关联的操作活动,这些活动反映了一个软件组织改进软件过程时所必须满足的条件。也就是说,关键过程域标识了达到某个成熟程度级别时所必须满足的条件。在 CMM 中一共有 18 个关键过程域,分布在 CMM 第 2 级~第 5 级中。
- ❑ **关键实践(key practices)** 是指关键过程域中的一些主要实践活动。每个关键过程域最终由关键实践所组成,通过实现这些关键实践达到关键过程域的目标。一般情况下,关键实践描述了应该“做什么”,但没有规定“如何”去达到这些目标。
- ❑ **软件过程评估(software process assessment)** 是用来判断一个组织当前所涉及的软件过程的能力状态,判断下一个组织所面向的更高层次上的与软件过程相关的课题,以及利用组织的鼎力支持来对该组织的软件过程进行有效的改进。
- ❑ **软件能力评价(software capability appraisal)** 用来判断有意承担某个软件项目的软件组织的软件过程能力,或是判断已进行的软件过程所处的状态是否正确或者正常。
- ❑ **软件工程组(software engineering group)** 负责一个项目的软件开发和维护活动的团体。活动包括需求分析、设计、编码和测试等。
- ❑ **软件相关组(software related groups)** 代表一组与实施软件工程项目相关的团体,它们支持但不直接负责软件开发或维护工作,如软件质量保证组、软件配置管理组和软件工程过程组等等。在 CMM 的关键实践中,软件相关组通常应该根据关键过程域和组织的上下文来理解。
- ❑ **软件工程过程组(software engineering process group)** 是由专家组成的组,他们推进组织采用的软件过程的定义、维护和改进工作。在关键实践中,这个组织通常是指“负责组织软件过程活动的组”。
- ❑ **系统工程组(system engineering group)** 负责下面工作的个人或团体,其作用是分析系统需求;将系统需求分配给硬件、软件和其他成分;规定硬件、软件和其他成分的界面;监控这些成分的设计和开发以保证它们符合其规格说明。
- ❑ **系统测试组(system test group)** 是一些负责策划和完成独立的软件系统测试的团体,测试的目的是为了确定软件产品是否满足对它的需求。
- ❑ **软件质量保证组(software quality assurance group)** 是一些计划和项目实施的质量保证的团体,其工作目的是保证软件过程的步骤和标准确实得到遵守。
- ❑ **软件配置管理组(software configuration management group)** 是一些负责策划、协调和实施软件项目的正式配置活动的团体。
- ❑ **培训组(training group)** 是一些负责协调和安排组织培训活动的团体。通常这个组织负责准备和讲授大多数培训课程并协调其他培训方式的使用。

1.1.2 CMM 的基本框架

1. 分级标准

CMM 模型描述和分析了软件过程能力的发展程度,确立了一个软件过程成熟程度的分级标准,如图 1-1 所示。

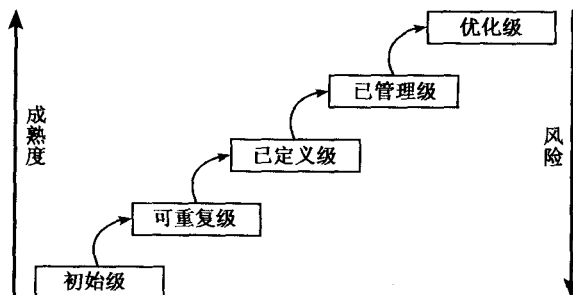


图 1-1 软件过程成熟度的级别

第 1 级 初始级 软件过程的特点是无秩序的,有时甚至是混乱的。软件过程定义几乎处于无章法和步骤可循的状态,软件产品所取得的成功往往依赖核心人物(个体)的努力和机遇。初始级的软件过程是未加定义的随意过程,项目的执行是随意甚至是混乱的。也许,有些企业制定了一些软件工程规范,但若这些规范未能覆盖基本的关键过程要求,且执行时没有政策、资源等方面的保证,那么它仍然被视为初始级。

第 2 级 可重复级 已经建立了基本的项目管理过程,可用于对成本、进度和功能特性进行跟踪。对类似的应用项目有章可循,并能重复以往所取得的成功。焦点集中在软件管理过程上。一个可管理的过程则是一个可重复的过程,一个可重复的过程则能逐渐演化和成熟。从管理角度可以看到这是一个按计划执行的且阶段可控的软件开发过程。

第 3 级 已定义级 用于管理和工程的软件过程均已文档化、标准化,并形成整个软件组织的标准软件过程。全部项目均采用与实际情况相吻合的、适当修改后的标准软件过程来进行操作。要求制定企业范围的工程化标准,而且无论是管理还是工程开发都需要一套文档化的标准,并将这些标准集成到企业软件开发标准过程中去。所有开发的项目须根据这个标准过程,剪裁出项目适宜的过程,并执行这些过程。过程的剪裁不是随意的,在使用前须经过企业有关人员的批准。

第 4 级 已管理级 软件过程和产品质量有着详细的度量标准,软件过程和产品质量得到定量的认识和控制;已管理级的管理是量化的管理;所有过程须建立相应的度量方式,所有产品的质量(包括工作产品和提交给用户的产品)须有明确的度量指标。这些度量应是详尽的,且可用于理解和控制软件过程和产品,量化控制将使软件开发变成为一个真正的工业生产活动。

第5级 优化级 通过对来自过程、新概念和新技术等方面的各种有用信息的定量分析,能够不断地、持续地进行过程改进。如果一个软件开发企业达到了这一级,表明该企业能够根据实际的项目性质、技术等因素,不断地调整软件生产过程以求达到最佳成果。

除第1级外,每一级都设定了一组目标,如果达到了这组目标,则表明达到了这个成熟级别,自然可以向上一级别迈进。因为从第2级开始,每一个低级别的实现均是高级别实现的基础,所以 CMM 体系不主张跨级别的演化。SEI 建议,从低一级别向高一级别演化的时间须用 12~30 个月。

CMM 分级标准有两个方面的用途。一方面,软件组织利用它可以评估自己当前的过程成熟度,并以此提出严格的软件质量标准和过程改进的方法和策略,通过不断努力去达到更高的成熟度。另一方面,该标准也可以作为用户对软件组织的一种评价标准,使之在选择软件开发商时不再是盲目的和无把握的。

2. CMM 的主要内容

CMM 为软件企业的过程能力提供了一个阶梯式的演化框架,它采用分层的方式来解释其组成部分。在第2~第5个成熟等级中,每个等级包含一个内部结构的概念。

每一级向上一级迈进的过程中都有其特定的改进计划,具体情况如图 1-2 所示。

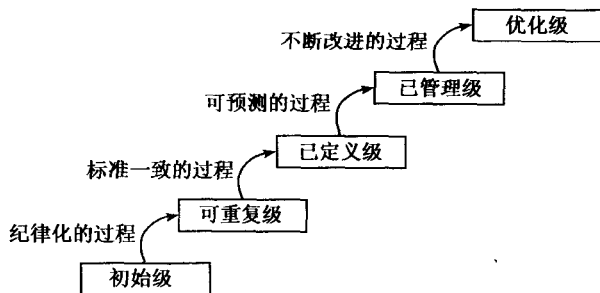


图 1-2 不断改进的过程

(1) 初始级的改进方向 建立项目过程管理,实施规范化管理,保障项目的承诺;进行需求管理方面的工作,建立用户与软件项目之间的沟通,使项目真正反映用户的需求;建立各种软件项目计划,如软件开发计划、软件质量保证计划、软件配置管理计划、软件测试计划、风险管理计划及过程改进计划等;积极开展软件质量保证活动(SQA)。

(2) 可重复级的改进方向 不再按项目制定软件过程,而是总结已实施的各种项目的成功经验,使之规则化,把具体经验归纳为组织的标准软件过程,把改进软件组织的整体软件过程能力的软件过程活动,作为软件开发组织的责任;确定全组织的标准软件过程,把软件工程及管理活动集成到一个稳固确定的软件过程中,从而可以跨项目改进软件过程,也可以作为软件过程剪裁的基础;建立软件工程过程小组(SPEG),长期承担评估与调整软件过程的任务,以适应未来软件项目的要求;积累数据,建立组织的软件过程库及软件过程相关