



全国工程硕士专业学位教育指导委员会推荐教材

林广艳 编著

Lin Guangyan

软件工程过程 (高级篇)

Advanced Software Engineering Process



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

清华大学出版社

全国工程硕士专业学位教育指导委员会推荐教材

林广艳 编著

Lin Guangyan

软件工程过程 (高级篇)

Advanced Software
Engineering Process

清华大学出版社
北京

内 容 简 介

本书以软件工程知识体 SWEBOK 2004、软件工程教育知识体 SEEK 和软件生存周期过程标准 ISO/IEC 12207 为依据,介绍了软件生存周期过程的基本概念、软件工程过程中包含的主要活动和软件工程过程模型。通过两个过程模型的应用案例说明了过程中各要素间的关系,介绍了软件工程过程模型的三个层次和建立软件工程过程的一般步骤、过程监控中需要考虑的基本问题,以及应用于软件工程过程改进的三类典型的标准评估模型。以极限编程 XP 过程模型为例介绍了敏捷过程倡导的向用户交付价值的思想,对计划驱动过程和敏捷过程进行比较,总结了其各自适用的领域,为更好地应用这些过程模型提供了指导。

本书作为“十一五”全国工程硕士研究生教育核心教材,其内容翔实,结合实际,实例丰富,论述深入浅出,且书中内容已经过多轮教学验证,既可作为软件工程专业高年级本科生、研究生及计算类相关专业的教材,又可作为专业软件技术人员的参考用书。

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

软件工程过程(高级篇)/林广艳编著. —北京:清华大学出版社,2011.10

(全国工程硕士专业学位教育指导委员会推荐教材)

ISBN 978-7-302-24149-2

I. ①软… II. ①林… III. ①软件工程 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2010)第 234439 号

责任编辑:魏江江 李 晔

责任校对:时翠兰

责任印制:何 芊

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

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62795954, jsjic@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×230 印 张:18 字 数:394 千字

版 次:2011 年 10 月第 1 版 印 次:2011 年 10 月第 1 次印刷

印 数:1~3000

定 价:32.00 元

前言

做任何事情都需要过程,软件开发也不例外。尽管我们都不否认过程对软件产品质量的作用,但我们常常会看到这样的现象:在进度压力下,首当其冲被简化的工作还是过程。规范化的过程使过程要素的执行更加严谨,导致短期的活动实施时间拉长,同时需要人力等资源的投入,也无形中增加了软件开发的近期成本。若选择的过程不合适,还会对开发起到阻碍作用。现实中很难有两个软件产品的开发过程完全相同。软件工程过程是对过去经验的总结,僵化地照搬任何一个软件工程过程的结果都可能事与愿违,这也是大家对软件工程过程敬而远之的原因之一。

软件工程过程的应用与所开发的产品的关键度、参与的人员状况、技术成熟度等因素相关,还与企业的文化相关。过程应用的关键是如何充分体现“平衡”的理念。在一个具体过程中,一个过程或活动是否应该有?若有,应执行到什么样的一个“度”?其中的“拿捏”讲究的就是平衡。平衡做得好,则可为软件工程过程中的各要素创造和谐环境,使项目或产品开发按预期进行;平衡做得不好,则过程各要素间不但相互制约导致预定义过程被束之高阁,当然项目或产品的目标也很难达到。软件工程过程应用中的这些现象也为弱化软件工程过程的人们提供了很好的理由。如今交付环境日趋复杂,稍具规模的软件企业都清醒地认识到:制造软件产品的过程和软件产品本身一样重要。因为高质量的软件产品的背后一定有一个合理的过程来保证。从长远角度看,软件产品的高产出和低成本背后一定有持续的过程改进做后盾。

如何理解软件工程过程?软件工程过程到底包括哪些内容?如何组织这些内容?如何正确地认识软件工程过程?为什么过程如此重要却未能得到足够的重视?针对目前软件工程过程应用中的这些问题或困惑,以及有关软件工程过程概念的不规范,要求我们把软件工程过程的相关内容说清楚,让学生明白软件工程过程中各要素间的内在联系,以及软件工程过程与企业文化的关系。

本书是作者多年来从事一线软件项目开发、管理与咨询过程中积累的经验与软件工程教学实践的结晶,书中内容结合实际案例对软件工程过程进行比较全面、清晰的论述。

本书在写作过程中,得到了很多人士的帮助。

麦中凡教授对本书的框架提出了非常中肯的建议并审阅了本书的大部分内容,在此表

Downloaded from

示感谢。

北京航空航天大学姚淑珍教授、杨文龙教授和国防科技大学的齐治昌教授,正是他们的信任、鼓励与支持,才使本书得以问世,在此表示感谢。

前人的工作是本书写作的基础,本书在写作过程中借鉴了前人已有著作和论文的内容,在此对列入参考文献部分的引用文献清单的作者表示感谢。

教学与研究工作是本书写作的基础,软件工程过程是北京航空航天大学软件学院重点建设的现代软件工程课程系列之一,自2004年开设以来,得到了学校精品课程建设的资助,学院给予了大力支持。在教学过程中,800多名本科生和2000多名研究生对课程的学习和反馈为本书的写作提供了帮助,在此表示感谢。

最后,特别感谢清华大学出版社给予本书的支持,感谢各位编辑为本书的策划和出版付出的心血。

因工作做得不够细致,书中未能明确标记文献的引用。限于编者的水平,难免存在错误与不妥之处,衷心希望广大读者指正赐教,联系 E-mail: lingy@buaa.edu.cn。

作 者

2011年6月于北京

目 录

第 1 章 绪论 /1

1.1 软件制造是个复杂的过程	1
1.2 软件产品与软件工程过程	2
1.3 研究软件工程过程的意义	4
1.4 软件生存周期过程标准	6
1.4.1 基本概念	6
1.4.2 ISO/IEC 12207 软件生存周期过程标准	7
1.5 本书结构	12

第 2 章 软件开发的主要活动 /13

2.1 需求工程	13
2.1.1 需求获取	14
2.1.2 需求分析	15
2.1.3 需求规格说明	15
2.1.4 需求验证	15
2.1.5 需求跟踪管理	16
2.1.6 需求变更管理	17
2.2 设计	17
2.2.1 高层设计	17
2.2.2 详细设计	18
2.3 构造	18
2.4 测试	18
2.5 运行与维护	19
2.6 软件项目管理	20
2.6.1 项目管理活动	20

2.6.2	软件开发计划	22
2.6.3	风险管理	22
2.7	配置管理	24
2.7.1	配置项和基线	24
2.7.2	配置库	26
2.7.3	配置管理流程	27
2.7.4	配置项标识	27
2.7.5	版本控制	29
2.7.6	配置控制	29
2.7.7	状态簿记	30
2.7.8	配置审计	31
2.8	验证与确认	31
2.8.1	V&V 的目标	32
2.8.2	计划 V&V 过程	32
2.8.3	软件 V&V 实施	33
2.9	软件质量保证	36
2.9.1	计划 SQA 过程	36
2.9.2	软件产品保证	37
2.9.3	软件工程过程保证	37
2.9.4	SQA 实施考虑	38
2.10	联合评审	46
2.10.1	计划评审过程	46
2.10.2	项目管理评审	47
2.10.3	技术评审	47
2.11	审核	47
2.11.1	审核过程规划	48
2.11.2	审核的实施	48
2.12	软件文档管理	48
2.12.1	计划文档过程	49
2.12.2	文档的设计和开发	49
2.12.3	文档的生产和发行	49
2.12.4	文档的维护	50
2.13	基础设施过程	50
2.14	改进过程	50
2.15	培训过程	51

2.16 本章小结	52
第3章 软件生存周期模型	/53
3.1 编码修正模型	54
3.2 瀑布模型	54
3.2.1 瀑布模型的优缺点	55
3.2.2 V模型	56
3.3 增量模型	57
3.4 演化模型	59
3.5 螺旋模型	60
3.6 原型构造在生存周期模型中的应用	62
3.7 生存周期模型中并发的作用	62
3.8 商业组件和复用的作用	63
3.9 统一软件工程过程模型	64
3.9.1 过程框架	65
3.9.2 核心元素	67
3.9.3 制品集的进化	68
3.9.4 项目计划	71
3.9.5 质量内嵌于过程中	77
3.9.6 主要困难与基础保障	78
3.10 MSF 过程模型	78
3.11 本章小结	80
第4章 瀑布模型应用实例	/81
4.1 过程实例活动	81
4.1.1 Infosys 过程模型概述	81
4.1.2 需求规范	83
4.1.3 高层设计	87
4.1.4 详细设计	89
4.1.5 构建(编码)与单元测试	90
4.1.6 集成测试计划与实施	91
4.1.7 系统测试计划与实施	93
4.1.8 验收测试与安装	94
4.1.9 维护支持阶段	95

4.2	文档编制	96
4.3	WAR 系统开发过程实施案例	96
4.3.1	WAR 系统需求概述	97
4.3.2	高层设计阶段	106
4.3.3	详细设计阶段	108
4.3.4	构建与单元测试阶段	109
4.3.5	系统测试	111
4.3.6	验收和安装	114
4.4	本章小结	115

第 5 章 协同过程模型 /116

5.1	模型概述	116
5.2	实际应用案例需求	117
5.3	初始阶段	118
5.3.1	基本活动	118
5.3.2	实施考虑	120
5.4	细化阶段	132
5.4.1	基本活动	133
5.4.2	实施考虑	136
5.5	构造阶段	149
5.5.1	主要活动	149
5.5.2	实施考虑	152
5.6	移交阶段	159
5.6.1	基本活动	159
5.6.2	实施考虑	161
5.7	本章小结	161

第 6 章 软件工程过程的建立与监控 /162

6.1	软件工程过程的层次	162
6.1.1	U 级过程模型	163
6.1.2	A 级过程模型	165
6.1.3	W 级过程模型	166
6.1.4	各级别过程模型的应用层次	166
6.2	软件工程过程的建立	167

6.2.1	定义软件工程过程的一般步骤	167
6.2.2	软件开发过程的定义示例	174
6.3	软件工程过程剪裁示例	182
6.3.1	概要级剪裁指南	183
6.3.2	详细级剪裁指南	184
6.3.3	WAR 项目的剪裁示例	186
6.4	项目计划的编制	188
6.5	过程的监控	188
6.5.1	过程变更处理	189
6.5.2	变更实施	190
6.6	过程改进	190
6.7	过程基础数据积累	191
6.7.1	过程数据库	191
6.7.2	过程能力基准	194
6.7.3	过程资源	198
6.8	本章小结	200

第 7 章 软件工程过程改进 /201

7.1	ISO 9001	201
7.1.1	质量体系框架	202
7.1.2	生存周期活动	203
7.2	CMM/CMMI	203
7.2.1	CMM 内部结构	204
7.2.2	CMM 成熟度级别	204
7.2.3	CMM 关键过程域	207
7.2.4	CMM 公共特征	211
7.2.5	CMM 关键实践	213
7.2.6	CMMI	213
7.2.7	评估模型的应用	217
7.2.8	其他过程评估模型	220
7.3	ISO/IEC 15504	221
7.3.1	ISO/IEC 15504 的组成	221
7.3.2	ISO/IEC 15504 的过程类别	223
7.3.3	ISO/IEC 15504 的能力等级	231

7.3.4	ISO/IEC 15504 的能力度量	233
7.3.5	ISO/IEC 15504 的应用	234
7.3.6	用 ISO/IEC 15504 开发与之兼容的评估方法	236
7.4	评估模型的发展	237
7.4.1	各种模型的比较	237
7.4.2	ISO/IEC 15504 标准发展趋势	238
7.5	本章小结	240
第 8 章	敏捷过程	/241
8.1	敏捷联盟	241
8.2	XP 概述	244
8.3	XP 过程模型	247
8.3.1	模型总框架	247
8.3.2	迭代的过程细化	251
8.3.3	开发的过程细化	252
8.3.4	集体拥有代码的过程细化	255
8.4	XP 项目小组模型	257
8.5	本章小结	257
第 9 章	软件工程过程的发展趋势	/259
9.1	计划驱动过程	260
9.2	敏捷过程	261
9.3	计划驱动过程与敏捷过程的比较	262
9.3.1	应用特征	262
9.3.2	管理特征	263
9.3.3	技术特征	265
9.3.4	人员特征	266
9.3.5	总结	268
9.4	敏捷过程与计划驱动过程的平衡	271
9.5	本章小结	273
参考文献	/275	

绪 论

尽管全世界已有数百万计的软件职业人,但软件工程作为合法的工程学科存在只不过是几年前的事情。软件工程过程作为软件工程学科的一个重要组成部分,必然越来越得到人们的重视。本章将首先回顾软件工程学科的发展历史,然后介绍软件工程过程的基本概念及其相关标准。

1.1 软件制造是个复杂的过程

软件伴随着第一台电子计算机的问世诞生了,随后以编写软件为职业的人也开始出现。由于受当时计算机硬件的限制,只有很少数的专业人员才能使用计算机,他们多是工作严谨的数学家和电子工程师。编写软件的唯一目的是完成大量的科学计算或自动化某些重复性工作,把人们从繁琐耗时的科学计算、工业控制和数据处理等重复劳动中解脱出来。

在计算机系统发展初期,通常只编写在特定硬件环境下运行的程序,因此其通用性很差。大多数软件是由使用该软件的个人或机构研制的,个人色彩很浓,当然也没有什么系统的方法可以遵循,软件设计是在某个人的头脑中完成的一个隐式的过程,且除了源代码往往没有软件说明书等文档。

后来,人们逐渐认识到软件与硬件的差异:

(1) 软件更容易更改,且不需要昂贵的生产线进行批量生产。程序一旦被修改,只需要把修改后的程序再装到其他计算机即可。这种便于修改的特点,使编程人员和软件开发组织开始采用一种“编码和修正”的方式来开发软件。

(2) 软件不会被用坏。使用硬件模型来度量软件的可靠性是不够的,而“软件维护”是一个与硬件维护大不相同的活动。软件是无形的,维护费用高。因其不确定性使软件开发很难按计划进行。为了赶工而在软件开发后期加入更多人的做法只会使工期更长。软件一般有更多的状态、模式和测试途径,使之更难以规格化。Winston Royce 在 1970 年的论文中提到:“为了使用一台 500 万美元的硬件设备,需要编写一本 30 页的详细设计说明书来

为完全控制生产提供足够的细节；但是为价值 500 万美元的软件提供 1500 页的说明，也只能刚刚比较好地对软件生产进行控制。”

(3) 软件需求涉及范围快速扩大，远远超出了当时从事软件开发工作的工程师和数学家的能力范围，也造成了软件人才的短缺。这导致大批具有创造力、但缺乏工程经验的人由于企业、政府和服务业对软件的需要，而涌入软件开发行业。“编码与修正”的方式非常适合这些人，他们思想开放，更喜欢按照自己的想法去实现工程。常见的人员状况是“牛仔程序员”，为了保证最后期限，他们可以用整夜时间来快速弥补有缺陷的代码，人们通常会把这些程序员当作英雄。但这些被认为是优秀的程序通篇充满了程序技巧，可读性差，维护也更加困难。

软件的数量急剧膨胀，软件需求日趋复杂，维护的难度越来越大，开发成本之高令人吃惊，失败的软件开发项目却屡见不鲜，出现了“软件危机”。

软件危机迫使人们从技术、人员、管理、开发工具等诸多方面系统化地思考软件的生产过程与维护过程，于是出现了软件工程。软件工程是借鉴传统工程的原则、方法，以提高质量、降低成本为目的地指导计算机软件开发和维护的工程学科，它包括建造软件的过程、方法、工具和质量要求 4 方面的内容。

1.2 软件产品与软件工程过程

软件工程的实质是在合理的时间和费用开销内，为用户提供满意的软件产品。编写软件本身是一项智力型的劳动，软件开发人员在从无章法地编程到使用结构化方法、面向对象方法以及现在基于模型驱动的开发方法的探索中，始终保持了高度热情。在支持工具方面，从业务建模工具、需求管理工具到集成测试工具等应有尽有。但随着软件规模与复杂度的提升，纵使招募了集先进的软件开发方法、工具于一身的高素质的软件开发人才，如果软件制造过程很弱，也很难保证项目的成功；同样，过渡依赖软件工程过程也有风险。简言之，正如 Margaret Davis 评论的过程和产品的两面性：大约每隔 5~10 年，软件界就会重新定义“问题”，将其焦点从产品问题转移到过程问题。为了提高程序的可维护性，出现了结构化程序设计语言(产品)，之后就有了结构化的分析与设计方法(过程)，有了数据封装(产品)，之后是软件开发能力成熟度模型(过程)，之后是面向对象方法(产品)，之后是计划驱动的软件工程过程，之后是模型驱动的开发，之后是敏捷过程，再后就是计划与敏捷的平衡(过程)。

软件界关注的焦点在不断移动，就像钟摆一样。当上一次摆动停止后，又有新力作用上去。这些摆动本身是有害的，因为它们彻底改变了完成工作的方法，使普通的软件开发人员无所适从，更不用说能很好地使用它了。造成这些摆动的原因，不是只注重软件产品而轻视过程，就是认为过程能解决软件开发中的一切问题，孤立地看待软件产品和软件工程过程。当然这些摆动也不能解决“问题”，因为它们注定是要失败的，只要产品和过程被视为二分的

而不是二元的。

软件制品及其开发过程明显体现出了产品和过程之间的二元性,因此单从过程或产品角度,很难得到或理解整个软件,如它的语境、使用、含义或价值。实际上人们从软件创造的过程中和从最终产品中可得到同样的甚至更多的满足感。

软件工程不是一项高度智能化的过程,必须根据软件开发人员的技术水平及其任务的需要进行动态调整。所以需要在组织对标准化和一致性的要求与个人对灵活性的要求之间进行权衡。需要考虑的一些因素包括:

- 由于软件项目各不相同,相应的软件工程过程也要有所差异。
- 在缺乏通用的软件工程过程时,组织和项目必须定义满足其特定需要的过程。
- 用于给定项目的过程必须考虑人员的经验水平、产品当前的状态和可用的工具、基础设施。

一个过程定义了为达到某个确定的目标,需要什么人在什么时间以何种方式做何种工作。对软件工程而言,其目标是构建一个新的软件产品或者完善一个已有的软件产品。一个有效的过程为开发高质量的软件提供准则,它获取并提出当前技术条件下可行的最佳实践方案,因此可降低风险并增强预见性,总的效果是要发扬一种共同的构想与文化。

也正是基于这些共同的认识,在软件工程知识体(Software Engineering Body of Knowledge, SWEBOK)中将软件工程分成十大知识领域(如表 1-1 所示),软件工程过程是其中之一。

表 1-1 SWEBOK 的知识域

序号	知 识 域	子知识域/知识点(参考文献)
1	软件需求	7/28(10)
2	软件设计	6/25(14)
3	软件构造	3/14(7)
4	软件测试	5/16(9)
5	软件维护	4/15(16)
6	软件配置管理	6/17(11)
7	软件工程管理	6/24(7)
8	软件工程过程	4/16(20)
9	软件工程工具与方法	2/12(7)
10	软件质量	3/11(68)

在这十大知识域中,“软件工程过程”知识域反映了近年来软件工程过程技术的成果,即一个产品的开发不是某个固定的过程模型,而是根据产品应用领域、开发企业的文化和资产,专门设计一个最优过程,即不仅要设计产品,还要设计过程;不仅要量度产品的质量,还要量度过程。但 SWEBOK 采取较为保守的态度,强调过程的变更和改进过程,而不是设计新过程,这既符合 ISO/IEC 12207 标准,可行性又强。

在 SWEBOK 中,软件工程过程包括两个层次:

第一层与技术和管理相关,实施的活动是软件获取、开发、维护、退役。第二层也称元层(Meta),涉及定义、实现、测量、管理、变更、改进等活动,后者也称为软件工程过程。

本书内容仅涉及第一个层次,即将用户需求转化为软件所需要的一整套软件工程活动,重点讨论软件产品开发与维护中涉及的核心工程活动。

1.3 研究软件工程过程的意义

做任何事情都有一个过程,软件开发也不例外。但因早期软件开发的主要障碍是技术——只要能做出来就行。因此,20 世纪 50 年代的软件系统主要用于解决科学计算问题。20 世纪六七十年代,为了使编程更加容易,出现了新的语言;为了使普通的编程者能把精力集中在应用系统开发上,出现了专门的操作系统来屏蔽计算机硬件系统的复杂度;为了使数据处理更加容易,在操作系统提供的文件系统基础上出现了数据库管理系统。但此时开发的软件对某些特定硬件的依赖性较大,独立的软件较少,因此软件开发技术处于主导地位,工程处于被动状态。

尽管早在 1968 年软件界已倡导软件工程,但计算机技术普及的主要障碍还是其技术的发展无法满足人们对软件的需要。直到 20 世纪 90 年代中期,人们主要的关注点还是技术,也因此使软件技术和开发方法呈现出飞速发展势头。软件工程的发展主要经历了以下几个阶段。

(1) 20 世纪 80 年代中期以前,人们主要使用基于结构化编程和瀑布模型的开发方法。

(2) 从 20 世纪 80 年代中期以后,结构化过程编程被面向对象编程取代。

(3) 网络技术成熟,基于单主机计算的开发开始向分布式客户/服务器计算的开发方法转移,然而却没有成熟的规范可以借鉴。瀑布模型受到批评以后,螺旋模型以其通过多次迭代接近实际,而备受推崇。但如何实施管理与进度预算、如何进行风险评估、如何实施里程碑评审等都没有公认的规范可以借鉴,可操作性弱,造成使用困难。

(4) 20 世纪 90 年代中期,软件工程过程、软件生存周期及软件评估规范开始成熟,如 ISO/IEC 12207。之后计算技术又发生重大变化,进入新的网络计算时代,软件开发的目标是提供 Web 服务。应用开发以集成服务为主、自行编码为辅,面向对象深化为构件和接口的链接,通信协议成为编程的重点。这种深刻的变化不要说没有成熟的规范和方法学,就连应用模式也还在探索之中。然而正是这种正在探索中的方法和模式却又在某些行业已经投入使用,且显示了巨大的优越性。

(5) 2000 年之后,技术发展已经不再成为人们应用计算机的障碍,软件开发关注的焦点也从原来的单系统开发到多系统开发以及系统中的系统,即群集系统的开发。多系统间的无缝衔接问题、遗留系统问题,系统的稳定性、易用性等要求,再加之快速变化的业务需

求、多个涉众利益的均衡等使软件开发与维护中的管理协调工作量变得越来越大,使软件制造过程变得越来越复杂。这些也迫使人们凭借以往的开发经验,在项目启动之初,制定足够详细的软件开发流程和规范,约束项目有关各方,规范所有参与人的行为,即通过规范化的软件工程过程来组织参与系统建设的各元素,使其按预定的轨道前进,保证项目的成功。

尽管开发组织不否认好的开发过程对软件项目的成功的保障作用,但当受到技术挑战、需求变更、交付时间和成本等方面的压力时,首先被忽略、弱化或破坏的还是软件工程过程。随着软件技术的基本成熟与稳定,大家逐渐意识到:拥有一批掌握先进技术的开发队伍和先进的工具也不一定能保证项目的成功交付。而技术、方法、工具的有机结合,人的配合,采用什么样的过程来组织这些资源对项目的成功交付所起的作用越来越大,对软件工程过程的研究也逐渐热起来。毋庸置疑,最终产品的建造过程对交付产品的质量起着至关重要的作用,因此在软件工程中要获得完全的成功,必须注意两个主要的子目标:

- (1) 实现一套成功的软件产品。
- (2) 执行一个成功的软件开发与维护过程。

而成功的软件工程是在这些目标间实现适当平衡的结果,即软件产品和软件工程过程之间的平衡。

软件工程理论的发展和实践经验的积累都表明了软件工程的复杂性和交叉学科的特点,到了2004年,软件工程成为一门独立的学科从计算机科学中分离出来,确立了其应有的地位。

软件工程是总结软件实践的科学。它力图总结实践中的普遍规律,以指导今后的实践,因而注定要滞后于工程实践。软件工程是一种层次化的技术,如图1-1所示。任何工程方法必须以有组织的质量承诺为基础,全面质量管理和类似的理念培养了不断的过程改进文化,正是这种文化导致了更成熟的软件工程方法的不断出现,而支持软件工程的根基就在于对质量的关注。



图 1-1 软件工程层次图

软件工程的方法提供了建造软件在技术上需要“如何做”。方法附加了一系列的任务:需求分析、设计、构建、测试和支持。软件工程方法依赖于一组基本原则,这些原则控制了每一技术区域且包含建模活动和其他描述技术。

软件工程的工具对过程和方法提供了自动的或半自动的支持。当这些工具被集成起来使得一个工具产生的信息可被另外一个工具使用时,就建立了支持软件开发的计算机辅助工程系统(CASE)。CASE集成了软件、硬件和一个软件工程数据库,从而创建了一个软件工程环境。

软件工程的基础是过程层。软件工程过程是将技术层结合在一起的凝聚力,使得计算机软件能够被合理地、及时地开发。过程定义了一组关键过程区域的框架,这对于软件工程

技术的有效应用是必需的。关键过程域构成了软件项目管理控制的基础,并且建立了一个语境,其中规定了技术方法的采用、工程产品(模型、文档、数据、报告、表格等)的产生、里程碑的建立、质量的保证及变更的适当管理。

软件工程过程提供了一系列软件人员的行为规范,这些规范开始作为一种约束行为管理对象,到最后变成软件从业者的自身素质和修养来自发指导其工作。

1.4 软件生存周期过程标准

1.4.1 基本概念

1. 过程

过程是针对一个给定目标的一系列运作步骤(IEEE-STD-610),是在过程环境中的一系列有序活动。活动(activity)是过程对象的一次状态改变,也叫过程步(step)。活动的起始态和活动结果态表征了过程的进行。可以说一切事物的发生、发展、消亡都离不开过程,都寓于过程之中。过程也是客观事物运动规律的体现。只有符合客观事物运动规律的过程才是正确的过程。过程的好坏由结果状态与预期状态的差异决定,也就是由目标成果质量的好坏来决定。

2. 规程

规程(procedure)是人们对客观事物运动规律的理解和掌握,是规范了的过程。过程中的活动有时必须顺序实施,如零件不加工完就无法装配,不装配好就无法试车。这种过程的模型是线性的活动序列。有时为了缩短工期尽量将逻辑上无上下文关系的活动(或子过程)并行实施,此时的过程模型是 AOV 网(管理科学中最常见的数学模型)。有时活动实施后有反复,如试车发现某个弹簧易损坏,则重新换成好材料加工,装配再试车,最后才是油漆、重装、交付。

3. 阶段

管理者有时为了管理方便,将若干子过程/活动组成为一个阶段(phase),阶段是一类活动可见成果完成的时间段、里程碑(这一阶段应达到的目标),察看过程实施情况,例如发标、投标、签约可以叫合同阶段,交出签字有效合同及相关文件就是这个阶段里程碑的标志物,里程碑也是协调并行活动的同步点。

4. 软件工程过程

软件工程过程是为了获得高质量软件产品所需要完成的一系列任务的框架,它规定了