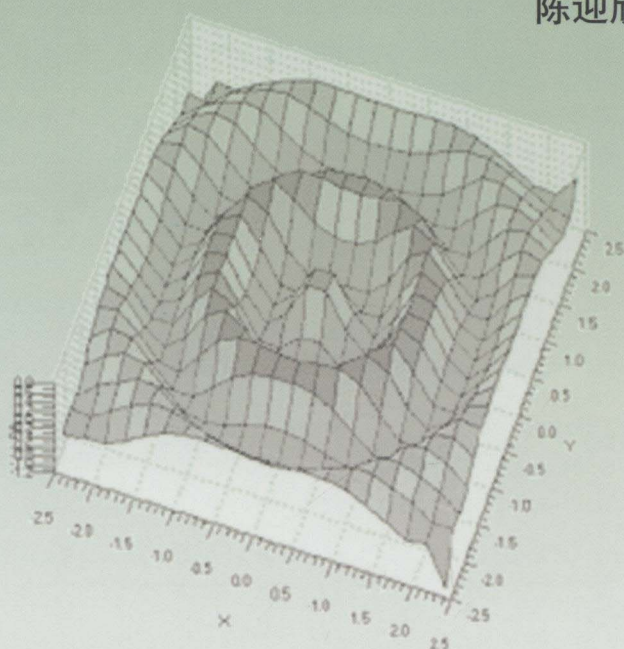


RUANJIAN GUOCHENG DE
JIANMO FANGZHEN YANJIU

软件过程的 建模仿真研究

陈迎欣 著



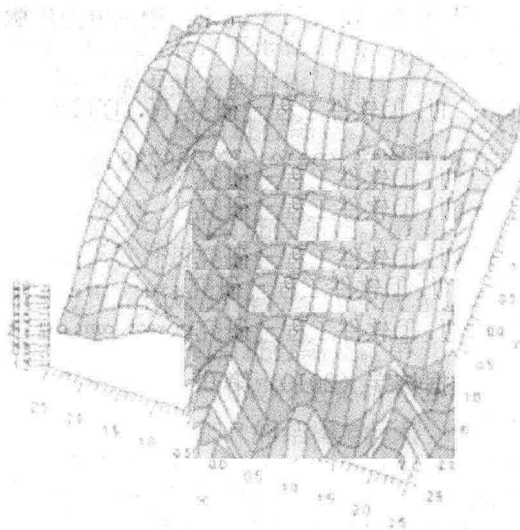
中国电力出版社

<http://jc.cepp.com.cn>

RUANJIAN GUOCHENG DE
JIANMO FANGZHEN YANJIU

软件过程的 建模仿真研究

陈迎欣 著
张 菁 主审



中国电力出版社

<http://jc.cepp.com.cn>

内 容 提 要

为了确保软件质量和提高产品竞争力,软件组织需要规范软件开发过程、实施软件过程管理。软件过程管理可以为快速开发高质量软件、有效地维护软件运行等各类活动提供指导性框架、实施方法和最佳实践。

本书从软件过程的应用实践出发,分析了软件过程应用中存在的问题;系统综述了软件过程改进以及软件过程建模仿真的研究动态;介绍了软件过程技术的产生、发展和主要研究方向,提出了软件过程技术的总体框架;阐述了小组软件过程 TSP 的产生和发展、TSP 的结构与流程,分析了 TSP 与 CMM/PSP 的关系;从目的、内容等方面介绍了软件过程建模仿真,并分析比较了典型的仿真建模方法;采用改进的基于多范例推理方法建立仿真模型,对范例表示、多范例派生、范例索引和检索、范例修改和储存方法进行了研究;采用离散事件建模和连续建模思想相结合的建模方法,将 TSP 抽象成内层和外层两个层次,并且对内层模型和外层模型的建立进行了阐述,对模型的形式化表示与仿真实现算法进行了研究;采用集合和关系代数演绎活动的执行及其对该过程的影响,将 TSP 抽象成多层排队的数学问题,对 TSP 静态特性仿真与判别以及 TSP 动态特性仿真进行了研究;提出了一种软件过程建模方法的评价体系框架,采用基于模糊集理论的综合评价方法,对三种 TSP 仿真建模方法进行了分析和比较;提出“过程—模式—元模型”递阶重用方法,利用演化计算自动生成基于元模型的软件过程,通过不同层次的过程模型重用实现过程开发的规范性、动态性和不确定性;设计实现了软件过程仿真平台,实现软件过程定义、软件过程执行以及软件过程评估功能,对软件过程高效实施起到重要的辅助决策的作用。

本书可作为高等学校计算机应用与软件专业和相关专业的本科生、研究生教材,也可以作为软件企业中各类管理人员和软件工程技术人员的参考书。

图书在版编目(CIP)数据

软件过程的建模仿真研究/陈迎欣著. —北京:中国电力出版社, 2010. 8
ISBN 978-7-5123-0443-7

I. ①软… II. ①陈… III. ①软件工程—研究 IV. ①TP311.5

中国版本图书馆 CIP 数据核字 (2010) 第 088738 号

中国电力出版社出版、发行

(北京三里河路 6 号 100044 <http://jc.cepp.com.cn>)

汇鑫印务有限公司印刷

各地新华书店经售

*

2010 年 8 月第一版 2010 年 8 月北京第一次印刷

787 毫米×1092 毫米 16 开本 11.5 印张 277 千字

印数 0001—1000 册 定价 18.50 元

敬告读者

本书封面贴有防伪标签,加热后中心图案消失
本书如有印装质量问题,我社发行部负责退换

版权专有 翻印必究

目前,包括中国在内的许多国家将软件业作为国民经济的支柱产业,由此软件的生产方式开始向全球化、规模化、工业化转变,软件生产开始进入以改善软件过程为中心的软件工程时代,因此,小组软件过程 TSP 等一系列帮助提高软件开发过程效率的方法应运而生。但是,软件过程改进是一项综合的且需要持续开展的活动,涉及软件过程工程的方法和过程领域的关键技术,对于 TSP 等过程框架实施的高效性、可靠性的检验只有在软件项目完成后才能获得,这种滞后性使软件过程的指导性大打折扣。因此对软件过程建立仿真模型并重用仿真模型的方法和相关技术的研究成为一个挑战性的、具有重要意义的理论和实际应用课题。该课题的研究对于提高软件项目管理和控制能力,提升软件企业的过程能力成熟度,促进我国软件产业的发展具有重要的指导意义。

本书从改善 TSP 的实施和提高项目的过程控制能力入手,给出三种仿真建模方法,并对三种方法进行评价,给出各自的适用性;将演化计算引入软件过程模型重用过程,对三种仿真建模方法的设计原理、实现算法、结果分析、软件过程模型重用方法及实现算法等进行了深入的研究;设计实现了软件过程仿真平台,分别对软件过程定义、软件过程执行以及软件过程评估进行模拟仿真。本书在结构上共分为八章:第 1 章是绪论,介绍本书的研究背景,并简要介绍了本书的主要工作、目标及成果,对软件过程改进、软件过程建模方法以及 PSEE 的研究现状及成果进行综述。第 2 章是软件过程及其仿真建模基础理论,介绍了软件过程技术的产生、发展和主要研究方向,提出了软件过程技术的总体框架,阐述了 TSP 的产生和发展、TSP 的结构与流程,分析了 TSP 与 CMM/PSP 的关系,从目的、内容等方面介绍了软件过程建模仿真,并分析比较了典型的仿真建模方法。第 3 章是基于多范例推理的仿真建模方法,将 TSP 规划的问题转变成多约束满足问题,改进了常规的基于范例推理方法,提出了范例表示、多范例派生、范例索引和检索、范例修改和储存方法。第 4 章是基于层次的混合仿真建模方法,将 TSP 抽象成两个层次分别对应于外层与内层仿真模型。外层从宏观的角度反映 TSP 的进展情况以及人员的调度情况,采用离散事件建模仿真思想;内层从微观的角度对不同的活动和产品进行定量的描述,采用连续建模仿真思想。第 5 章是基于多层排队建模和相关矩阵一致性判断的仿真方法,准确地分析出 TSP 的静态特性和动态特性,采用集合和关系代数演绎活动的执行及其对该过程的影响,将 TSP 抽象成多层排队的数学问题,运用多层排队和相关矩阵一致性判断的方法帮助管理者迅速做出正确的过程控制决策。第 6 章是仿真建模方法的评价体系框架与模型重用,提出了一种软件过程建模方法的评价体系框架,并给出一个基于模糊集理论的综合评价方法,采用模糊综合评价方法对文中阐述的三种 TSP 仿真建模方法进行了分

析和比较。通过“过程—模式—元模型”三个层次重用，体现了从大粒度到小粒度重用，以及由内容重用到结构重用的发展，将演化计算应用于元模型重用，通过繁殖、杂交和变异等遗传操作，生成符合需要的软件过程。第7章是软件过程仿真平台的设计与实现，设计和实现了软件过程定义、软件过程执行以及软件过程评估三大模块，详细阐述了三大模块的总体设计、主要功能模块的实现及主要界面。第8章是结论与展望，总结了本书的研究成果和创新点，并对未来研究进行了展望。书中的软件过程仿真建模方法已应用于研发项目，结果证明仿真模型可以用来分析软件过程的人员分配、工作进度、过程控制和决策支持等方面的情况，为应用软件开发过程的事前预测和分析提供了有效的途径。

本书的研究成果来源于国家自然科学基金项目（60803087）、黑龙江省哲学社会科学基金项目（08E004）、黑龙江省教育厅人文社科基金项目（11544056）、黑龙江省博士后基金项目（LBH-Z09221）和哈尔滨工程大学基础研究基金项目（002090260763）。在撰写过程中，得到了恩师的悉心指导，以及同事和师姐师妹们的热情帮助，在此表示由衷的感谢。由于作者的水平有限，书中难免会有不足之处，恳切希望广大同仁和读者给予批评和指正。

陈迎欣

2010年2月于哈尔滨

前言

第 1 章	绪论	1
1.1	研究的背景	1
1.2	研究的目的和意义	2
1.3	国内外研究现状	4
1.4	本书的主要内容和组织结构	17
第 2 章	软件过程及其仿真建模基础理论	20
2.1	软件过程技术	20
2.2	小组软件过程 TSP	24
2.3	仿真建模方法和技术	30
2.4	软件过程建模仿真	34
2.5	本章小结	39
第 3 章	基于多范例推理的仿真建模方法	41
3.1	基于范例推理概述	41
3.2	基于范例推理的软件过程仿真建模原理	46
3.3	软件过程的问题描述与计算模型	49
3.4	软件过程的基于约束满足的多范例推理	50
3.5	基于多范例推理的仿真建模方法分析	60
3.6	本章小结	61
第 4 章	基于层次的混合仿真建模方法	63
4.1	混合系统建模	63
4.2	基于层次的软件过程混合建模原理	70
4.3	外层模型	72
4.4	内层模型	75
4.5	基于层次的混合仿真模型的建立	81
4.6	仿真模型的实验验证	84
4.7	本章小结	87
第 5 章	基于多层排队建模和相关矩阵一致性判断的仿真方法	88
5.1	基本原理	88

5.2	软件过程的静态特性	90
5.3	软件过程的动态特性	92
5.4	软件过程静态特性的仿真与判别	96
5.5	软件过程动态特性的仿真与分析	99
5.6	仿真实验及分析	103
5.7	本章小结	105
第 6 章	仿真建模方法的评价体系框架与模型重用	106
6.1	引言	106
6.2	软件过程仿真模型的应用说明	107
6.3	软件过程仿真建模方法的评价体系框架	109
6.4	模糊综合评价方法及其在本书中的应用	111
6.5	三种仿真建模方法的比较分析	114
6.6	软件过程的递阶重用	117
6.7	软件过程递阶重用的设计和实现	122
6.8	本章小结	128
第 7 章	软件过程仿真平台的设计与实现	129
7.1	软件过程仿真平台的整体概述	129
7.2	软件过程定义模块的设计与实现	130
7.3	软件过程执行模块的设计与实现	142
7.4	软件过程评估模块的设计和实现	149
7.5	本章小结	164
第 8 章	结论与展望	165
参考文献		167

绪 论

本章的内容包括：研究的背景，研究的目的和意义，国内外研究现状，本书的主要内容和组织结构。

1.1 研究的背景

软件工业已经经历了几次浪潮。到目前为止，其中有三次特别明显的发展浪潮。自从 1968 年北大西洋公约组织的计算机科学家在联邦德国召开的国际学术会议上第一次提出了“软件危机”（Software Crisis）以来，人们就开始着力想办法解决如何开发软件来满足不断增长、日趋复杂的需求和维护数量不断膨胀的软件产品的危机。这一阶段出现了以瀑布式生命周期和结构化的方法为特征的第一次浪潮。随后，软件生产转向以过程为中心的软件工程时代，掀起了过程成熟度运动，出现了软件业的第二次浪潮。第三次浪潮是软件工业化，它标示着软件的自动化生产时代。

目前，大多数国家的软件工业发展正处于以过程为中心的成熟度运动。以过程为中心的成熟度运动根源来自于全面质量管理，目的在于提高软件质量，提升软件的生产能力。随着“过程思维”的深入，各国软件企业都意识到有必要对软件过程加以改进，软件过程改进的程度与产品的质量保证和企业的生产能力及发展有着必然的联系。为了更好的促进软件过程改进，国内外许多研究人员都着力于理论上的研究分析并加以企业实践的结合提出了更好的建议：他们主要关注过程改进的策略、实施步骤，关注软件过程改进的投资回报率（Inspection Return of Investment, ROI），为此他们提出了软件能力成熟度模型（Capability Maturity Model, CMM）、小组软件过程（Team Software Process, TSP）、个体软件过程（Personal Software Process, PSP）等一系列帮助提高软件开发过程效率的模型。他们在基于 CMM/CMMI（能力成熟度模型集成）、ISO/IEC15504 和 BOOTSTRAP 等国际公开标准上提出了对软件过程改进各阶段的措施和改进办法。与此同时，软件企业也在不断注入新的思想，引入新的技术、方法运用到软件生产过程中来。

我国从 20 世纪末开始在政府的大力推动下全面地展开了软件过程改进，建立了一整套系统，并大力推广 ISO 9000 和 CMM/CMMI 模型。从 CMM/CMMI 模型推广来看，我国的速度在世界上是最快的，也就是被评估的组织和被评估的项目数量发展是最快的。从 2004 年开始，我国开始进入 CMM/CMMI 评估认证高峰期，据美国卡内基梅隆大学（Carnegie Mellon University）软件工程研究所（Software Engineering Institute, SEI）资料显示，国内通过 CMMI 评估的企业有 117 家，覆盖了 CMMI L2~L5 的所有等级。

2005 年底,中国通过 CMM/CMMI 评估的企业数量已位列全球第四。到 2006 年 6 月 30 日为止,通过评估企业的总数达到 448 家,其中四级及四级以上的高成熟度企业达到 52 家。现在国内著名的多数软件企业,如联想、用友、金山、北京日立华胜、方正科技、深圳华为、上海宝信、惠普等都已经通过了 CMM/CMMI 高等级评估认证。

尽管我国软件企业一直保持高速成长,但还是存在一系列问题,如产业规模小;软件企业创新能力不足,造成附加值高和产业链上游软件产品市场大多被国外企业垄断;软件企业工程化管理水平偏低,没有形成完善的管理体系,难以适应竞争和发展的要求;软件人才结构不合理,知识更新慢,不适应国际市场要求等。我国软件企业现在所存的问题归根结底也正是软件过程管理能力欠缺所引起的。因此,大力加强软件管理能力,改进软件过程是解决问题的必须途径。虽然,现在我国在软件过程改进上有一定成效,但是有些方面的不足是不容忽视的,如中小型软件企业对过程改进的认识还不是很清楚,管理上存在大量漏洞。由于国内企业的软件过程能力比较落后,过程改进的基础比较落后,很多的企业对过程改进的认识还存在误区,很大部分企业只注重追求最终结果,以达到 CMM/CMMI 等级级别而作为目的。他们进行 CMM/CMMI 评估的目的只是借一本证书来打开产品销路,提高企业地位。这样就会出现企业为评估而制定的一系列规范和制度在通过评估后就抛之于脑后,走的就是一个形式,软件过程改进的最终目的并没有实现,企业的软件过程没得到相应改进,管理能力也没有得到提高。经过多年的探索和实践,软件企业也逐渐意识到实质性问题缺乏管理软件过程的能力。起初由于软件企业过分重视技术,强调技术统领一切,盲目扩大开发队伍,不断压缩管理成本,项目的开发过程处于无序的混乱中,导致无法开发符合预算和进度要求的高可靠性和可用性软件。随着软件过程改进概念的引入,很多企业根据改进模型对企业内部进行改进,通过改进后不仅加强了开发过程的管理,提高了产品质量、生产能力,也为企业带来了商业上的收益。

本书的研究工作正是基于这样的背景提出的。建模仿真的本质是对于过程的改进而言,更强调软件过程的重要性,把整个软件视为可控制、度量和改进的过程。而过程改进的重要意义也就是对于效果良好的项目实践要推广应用,对于问题较多的项目实践要变更调整,通过改善软件开发和管理的方式来达到企业的某个业务目标。因此为了促进软件过程的不断持续改进优化,有必要对软件过程进行建模仿真,对软件过程运作中出现的問題进行分析和研究,为软件企业更快更好地开发项目提供借鉴。

1.2 研究的目的是意义

国际上采用的软件过程改进模型 CMM/CMMI 是软件过程改进领域的重要成果,它是适用于软件企业质量管理和过程改进的重要标准。但是,这些模型主要是针对大型软件企业的,根据中国软件行业协会 CSIA 的调查统计,从软件企业的数量上看,中国软件企业的数量并不少,但是大部分软件企业规模较小,50 人以下的企业占 67%,50~100 人的企业占 18%,100~300 人的企业占 14%,300~500 人的企业仅占 1%。这些中小企业根据 CMM/CMMI 模型来改进自己的软件过程确实存在一些实际困难。首先,我国中小软件企业员工总数比较少(绝大多数少于 50 人),企业组织结构大多是扁平型,没有专门的软件过程和软件质量保证部门。往往需要一个人承担几个角色的职责,无法适应 CMM/CMMI 模型中要求的多层次的组织结构,这使得在中小软件企业中进行基于

CMM/CMMI 的软件过程变得非常复杂。第二, CMM/CMMI 模型中的许多关键实践对这些软件企业并不合适。因此企业无法按照 CMM/CMMI 模型来实施。第三, CMM/CMMI 模型要求大量的文档, 建立和维护这些文档需要企业投入大量的资源, 这是中小软件企业难以承受的。最后一点, 也是最重要的一点, 基于 CMM/CMMI 模型的过程改进和进行 CMM/CMMI 评估需要大量的资金支持, 这是许多中小企业所难以承受的。一般一次评估大约需要投入在 50~100 万元左右, 评估周期一般是 8 个月。正是由于以上种种原因, 目前在世界范围内很少有中小软件企业成功地实现基于 CMM/CMMI 模型的软件过程改进。为了解决这些问题, TSP 非常适合我国中小软件企业, TSP 是一个具体的软件过程, 它是针对一个项目组进行软件过程的实施和改进, 同时它支持基于 CMM 的软件过程改进。TSP 明确定义了一套可操作的过程, 为软件工程师和项目管理者导航, 通过 TSP 对开发过程的指导, 使得开发人员有条不紊地完成小组建立、小组成员的协调等工作, 从而获得高生产效率并生产出高质量的软件产品。

TSP 技术已经应用于软件项目的开发, 但是这些企业远未取得预期的目标, 中小软件企业仍然面临“项目延期交付、费用超支、效率不高”的共性问题。分析造成这些问题的原因, 一是由于我国中小软件企业实施 TSP 的历史较短, 企业的经验较少; 二是 TSP 关注组织过程能力成熟度的提高, 强调软件过程具有稳定性、有规律、可量化的特点, 而往往忽视具体项目目标的实现。在按照 TSP 进行软件开发时, 会由于环境、人员、技术、资金等诸多因素的不确定性, 使其管理下的项目具有复杂性和多样性, 因此人员、时间、设备等因素必须随时对环境的变化做出调整, 这些问题都没有得到很好地解决; 三是 TSP 最终目的在于指导开发人员如何在最少的时间内, 以预定的费用生产出高质量的软件产品, 然而这种指导的可靠性、正确性、高效性只有在整个项目按照 TSP 完成后才能评价, 这种滞后性往往使 TSP 的作用大打折扣, 我国中小软件企业缺乏有效的模拟仿真和事前预测方法与技术手段。如何在软件项目开发前获得项目计划与跟踪、进度的管理和质量控制、资源的分配等数据, 预先分析 TSP 实施的效果成为软件开发企业普遍关心的问题。

本书正是解决上述问题, 专门研究针对中小软件企业的基于 TSP 的软件过程改进优化, 建立了三种仿真模型, 并且对这些模型及建模方法进行了分析比较, 提出了模型重用的方法, 并设计和实现了软件过程管理仿真平台。主要有两个方面的意义: 一方面, 可以通过仿真模型使 TSP 更好地发挥其指导作用, 进而提高团队协作的开发能力, 事前预测开发过程中可能存在的缺陷和瓶颈, 有利于消除缺陷和优化软件开发过程, 并且可以协助决策制定、监督和控制项目的开发过程, 改进开发过程。另一方面, TSP 仿真模型具有重用性。每个软件项目都是独特的, 这种独特性意味着各个项目的实施过程将各不相同, 针对每一个软件项目都建立软件过程模型是比较烦琐和费时的的工作, 因此重用已有的软件过程模型不仅能缩短开发周期, 提高开发效率, 也能提高软件的可维护性和可靠性。

若要有用地指导软件工程实践特别是探索出一条适合我国中小软件企业特点的过程改进方法并解决由此引出的相关技术问题, 软件过程建模仿真、重用方面还有众多的研究工作要做。因此对软件过程建模仿真、重用方法和相关技术方面的研究是一个挑战性的、具有重要意义的理论和实际应用课题。该课题的研究对于提高软件项目管理和控制能力, 提升中小软件企业的过程能力成熟度, 促进我国软件产业的发展具有重要的指导意义。

下面介绍软件过程改进和软件过程建模方面的国内外研究现状、存在的不足，以及本书的一些基本观点。

1.3.1 软件过程改进国内外研究现状

通过软件开发实践，人们逐步地认识到软件产品的质量在很大程度上依赖于产品开发时所使用的过程，即生产高质量的软件需要有一个高质量的软件过程。由于影响软件开发的各因素，比如商业环境、开发技术以及开发人员，总是在持续不断地变化，因此一个高质量的软件过程也必须是一个持续改进的过程，而软件过程改进也构成了软件过程管理活动的核心。下面对于软件过程改进的国内外研究现状进行综述。

1.3.1.1 国外研究现状

为了支持软件过程的改进，研究者们提出了过程改进的不同周期模型，其中，Dion (1993) 提出了一个非常典型的三阶段过程改进模型，用于指导具体的过程改进活动，如图 1-1 所示。

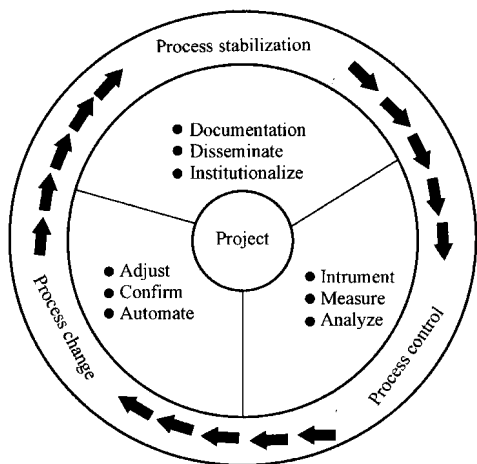


图 1-1 一个典型的软件过程改进生命周期模型

过程制定 (Process Stabilization) 是该周期模型的第一个阶段，在该阶段将描述所要执行的过程，并通过特定的方式发布过程的文档，同时确保所发布的过程在整个组织内得到执行；在随后的过程控制 (Process Control) 阶段，通过相应的工具支持项目数据的收集，在这些数据的基础上决定如何控制过程的执行；在最后的变更 (Process Change) 阶段，根据度量和分析的结果决定从哪些方面改进所执行的过程，在实施新的过程之前可能在小范围内进行实验和确认。此时，软件过程改进活动完成了一个周期，而后进入下一个改进周期。与此相类似，Humphrey (1995) 指出软件过程改进活动包含如下 3 个方面：过程定义

(Process Definition)，需要明确地定义所要执行和改进的过程，包括清楚地定义过程内所有的活动、活动执行顺序以及活动的出口标准等；过程使用 (Process Use)，通过使用所定义的过程，在过程执行中发现一些改进机会；数据收集和分析 (Data Collection and Analysis)，通过收集和分析过程执行时的数据，确定软件过程中的问题并决定如何进行改进。其他过程改进周期模型包括 Ishikawa 1985 年提出的 PDCA (Plan-Do-Check-Act) 和 QIP (Quality Improvement Paradigm)，以及 Gremba 等 (1997) 提出的描述了更多组织层次内容的模型 IDEAL 和 ISO 15504—7 等。

从软件过程改进的周期模型可以看出，软件过程改进覆盖了从过程定义、执行、分析到过程变更 (或者演化，Evolution) 的一系列过程管理活动，因此，Kinnula (2001) 直接称软件过程改进为软件过程工程 (Software Process Engineering)。对于软件过程改进的研究，国外研究人员从不同的方面提出了如何有效地进行软件改进。

(1) 不同规模软件企业进行改进的研究。虽然软件过程改进的提出是针对所有软件企业的,但是它是否能运用于不同规模的企业并取得成效呢?针对这一问题,Declan P. Kelly、Bill Culleton (1999) 和 Tore Dyba (2003) 等人从软件企业的规模入手进行了研究。Declan P. Kelly 等人对拥有 150 名员工的小型企业 S3 取得软件改进做出研究分析,提出了改进的关键步骤:要正确制定目标,做好前期评估,对过程加以定义并实施好计划。在实施好计划这项中,重点强调了在按计划实施时要时刻测试和监控发生的改变,指出增加细节的层次能完善进度。Tore Dyba 对北欧 55 家大小型软件企业的 120 名软件和质量管理人员进行抽样调研,得出这些企业改进之所以成功与商业定位、领导能力、员工参与、度量、开发和对新知识探索这些独立变量有关,改进成功的大小由这些因素在大小型企业所占的主导地位不同;并提出软件过程改进可以作为加强竞争力的策略。

(2) 软件过程改进整体实施研究。软件过程改进是适应于任何规模的企业,根据企业实施软件过程改进后做出的改变,一些研究者对过程改进的重要性做出了进一步强调。Bill Curtis (2000) 认为实施软件过程改进能为企业带来效益,它能消除开发中的重复工作,减少成本和提高生产率;并可利用软件改进的标准来改变组织;要从改进的成功或失败吸取教训,并提出把软件过程改进看作项目来管理。还有部分研究人员对知名的软件企业成功进行软件过程改进的情况进行了全程的调查与分析,如对大型企业东芝进行改进的方法过程的研究,对 New World Commerce、Motorola Cork、Silicon and Software Systems 和 Allied Irish Bank 四家软件企业实施软件改进进行的调研等。Fran O. Hara (2000) 就对四家软件企业从各企业现状分析、启动时的定位、评估方法、采取策略和管理方法及人员问题都做出全面的分析和对比,最后从这些企业的成果得出了重要结论,其中就提出了在改进可见度和跟踪过程方面要视软件过程改进为一项目,也就意味要应用类似于产品开发项目的结构化的编制和跟踪机构。同样, Bill C. Hardgrave、Deborah J. Armstrong (2005) 对美国一家通过改进取得成功的软件公司也做了分析,发现了存在的问题:如管理者决定在某一时间要完成改进目标,而实际的时间要长得多;初期阶段缺乏管理。并提出了要注意的地方:要有正确的指导;有管理者的支持;同时组织中的成员都要理解软件改进;把启动软件过程改进当成一个项目并提供必要资源;根据组织自身情况进行改进等。尤其 Bill C. Hardgrave (2005) 等人提出了软件过程改进是一个长期的过程而不是最终的目的,更明确地指出了改进的方向。

(3) 软件过程改进管理研究。管理是软件过程最薄弱的一个部分。软件过程改进的正式启动到软件过程的持续改进是一个漫长的过程,这个过程可以划分为不同的阶段,而且在每个阶段都会遇到不同类型的管理问题。Marcroft K. M. (1991) 通过分析给出了一些管理技术和程序来减少开发风险。Michael Deck (2001) 提出对差异性的过程进行管理,认为如果不认真做计划和认识到管理过程的多样性,那么很快将再无法管理,文中他利用风险分析的方法对过程改进目标进行选择 and 计划差异性过程。Lars Mathiassen、Ojelanki K. Ngwenyama (2005) 也指出虽然每个组织的情况不同,但是他们所面对的关于管理的改变这方面的挑战是相同的。他们对四个丹麦的软件企业从 1997~2001 年的管理上出现的改变做了分析,找出管理中改变的元素:过程、人员、组织结构和管理方式。Timo Varkoi (2002) 通过从一些企业进行改进的情况指出了软件改进在管理上面临的典型问题在于不能保证改进的时间、缺乏实施改进计划的实际能力;他还对改进的三大部分需求分析、过程评估和改进实施分别提出了应注意的一些问题。除此之外, Alessandro Barbieri

等 (1992)、Denis Avrilionis 等 (1996)、Karlheinz Kautz 等 (2000)、David S. Hinley 等 (1993)、Siew Hock Ow 等 (1997)、Magne Jorgensen 等 (2003) 从软件过程改进的模型、方法, 改进中的风险、评估人员能力和成本管理方面做了相应的研究。

1.3.1.2 国内研究现状

国内对软件过程理论的讨论与实践目前正在热烈开展, 目标是使国内的软件质量管理和控制达到国际先进水平, 使中国的软件产业获得可持续发展的能力。自 2004 年以来, 由工业和信息化部、国家发改委、科技部、中国软件行业协会等部门主办了一系列“系统与软件过程改进国际论坛”, 2006 年论坛探讨了从过程到系统, 从高成熟度的大企业到中小企业的过程改进以及 TSP、PSP 的应用; 2007 年, 以基准数据与度量分析、软件基准数据库应用与发展研究、测试与绩效考核、企业基准数据建立与交付能力的保障等为内容, 研究与探讨软件度量基准数据公共服务平台; 2008 年, 对软件服务化时代的生产力、度量与造价的估算、高成熟度过程改进实践、项目管理、过程改进模型及工具等进行探讨; 2009 年, 面对席卷全球的金融危机, 探讨如何降低 IT 服务成本、提高 IT 服务质量, 进而支撑企业业务的发展需求, 指出要提高 IT 服务的水平, 就必须对 IT 服务管理进行量化, 量化能更好地发挥 IT 服务管理的威力。论坛共同分享软件过程改进成果, 共同促进中国软件过程改进迈向国际水准。随着软件过程的不断深入研究, 软件过程改进领域涌现出一批专家, 例如, 中国软件过程改进专业委员会 (CSPIN) 专家王树文, 精通 PMBOK、ISO、CMMI, 对企业管理、项目管理、软件过程改进等有深入的认识和独到的见解; 清华大学教授、北京软件行业协会软件过程改进分会会长郑人杰, 参与制订、审定一些软件工程国家标准, 配合质量认证机构参与若干软件企业 ISO 9000 质量管理体系的认证审核, 为软件企业实施 ISO 9000 及 CMM 提供咨询, 对基于软件能力成熟度模型的过程改进有深入的研究; 北京航空航天大学计算机学院教授、北京航空航天大学软件工程研究所名誉所长周伯生, 多年从事软件工程环境、过程工程 and 环境的研制与开发工作, 为多家企业实施 CMMI、过程改进进行培训、咨询和评估, 在过程模型和过程改进技术领域进行了大量的研究; 中国科学院软件研究所所长李明树研究员, 多年从事软件工程方法、软件过程技术、需求工程、软件工程经济学和可信软件过程等领域研究, 在软件过程改进与质量保障技术领域有独到见解。

与国外从各角度对软件过程改进进行研究和取得的成效来比, 由于我国对软件过程改进研究的出发点不同, 大多数学者把进行软件过程改进的方法作为研究对象。由于 CMM/CMMI 在国内各软件企业流行开来, 不少企业为追崇 CMM/CMMI 的等级, 盲目进行改进。为此, 一些研究人员从 CMM 的具体内容着手, 把 CMM 与其他标准 (如 ISO 9000 标准) 进行比较, 如杨一平 (2002) 对 CMM、ISO 9000、Clean room、ISO/IEC 15504 进行了比较, 分别对他们的特点、缺陷以及适用性进行了研究; 林琳等 (2004) 对 CMM、ISO/IEC 15504、BOOTSTRAP、ISO 9000, 从发展历程、体系结构、模型特点和使用方面进行对比分析, 并对几种方法的使用范围和模型的优缺点阐述了自己的观点; 李致君 (2006) 分析了 CMM、TSP 和 PSP 标准, 以及其相互间映射关系, 提出了基于 CRT 项目的切实可行的软件过程方法; 王日宏等 (2007) 分析了 TSP 的概念、创建及流程框架, 并给出了 TSP 与 CMM/CMMI 在实践上的关系; 王易等 (2009) 分析了 CMM/CMMI 的缺陷, 指出了敏捷软件过程的特点和优势。有些学者分析了国内软件企业存在的问题, 讨论了现有企业实施软件过程改进方法的不足, 以及在改进中存在的误区, 对改

进的方法进行研究和设计,例如,林益虹(2004)在对CMM研究的基础上,结合我国大多数软件企业的实际情况,提出了我国软件企业进行软件过程改进的方法和步骤。吴忠懿(2009)详细描述了软件改进工程的特点和该工程的组织和管理形式,从企业组织级讨论了如何通过软件过程改进工程的进行达到提高企业软件能力的目标。周伯生(2005)指出过程改进要注意几方面问题:一是质量,过程改进取决于几方面的因素:政府的政策、企业的领导、过程改进的专业队伍、项目级和广大员工,还有责任评估师;二是思想问题,过程改进并非立竿见影,要进行逐步改进;三是要区分软件规模;四是过程改进要有工具的支持。马良荔等(2004)总结了现有软件过程活动中存在的问题,提出了两种软件过程改进方法,即基于同行评审和测试的软件过程改进方法和基于CASE工具和文档的软件过程改进方法,这两种方法是分别针对现有软件开发机构实施过程改进时遇到的实际问题,结合CMM的关键实践而设计。蔡倩等(2005)给出了一种软件过程规范的框架,并就实施软件过程规范提出了参考意见。

在软件过程改进方法的指导下,一些研究者发现对于过程的改进不仅要有正确的方法和技术,还应注意开发过程中对各阶段的有效管理。由于软件的特殊性质,使得管理问题难度加大。因此,在对软件如何管理的问题上,宴燕(2003)从项目管理的角度探讨管理的关键因素如计划、流程、人员的沟通等,并对软件改进中的管理提供了建议;简浩(2003)等研究人员参与研发国内第一个可用于协作软件开发的管理平台On Team,提出了如何将软件实施流程优化的具体方案;方春春(2003)等人基于构件构架复用技术的现代软件开发的模式,提出了改进软件开发的过程管理中的一些有效措施和方法;范勇(2006)从管理原则、管理层次和管理的任务等方面进行研究,提出了全面软件过程管理模型;王青等(2005)提出了一种支持软件过程控制与改进的主动度量模型和度量方法,模型形式化描述了软件过程的目标、特征和度量指标等关键元素以及相互间的关系,给出了确定软件过程度量的原则、方法和步骤,为正确的决策和成功的结果提供有效的支持。田道云(2002)等在管理组织模式基础上,综合对软件管理的配置管理、人力资源管理和成本、进度管理等方面做出论述;许东(2007)将软件过程度量和过程产品度量相结合,提出一个新的过程管理模型GQ(I)M-D,解释了该模型的元素语义及其相互关系,并将其用于过程管理的实施,建立了一套软件过程和过程产品可操作的客观度量指标,以达到量化的过程管理,设计并实现了一个用于软件项目规模估计的源代码统计工具。同时,有的研究人员也对改进中出现的管理问题及风险做了专门的研究:郑中(2008)提出以软件项目风险大小作为评估当前软件项目过程优劣的依据,并从优化软件项目风险控制的视角对软件过程进行优化,描述并分析了基于CMM的软件风险管理特点,给出一个以风险数据库为核心的风险监控体系架构和一个通用的风险数据库模式,提出一个基于风险传递的软件风险优化控制模型和一个动态规划的软件风险控制离散优化算法;杨萍等(2007)从我国中小软件企业的软件项目管理现状出发,在分析CMMI和其他当前流行的风险管理模型的基础上,提出中小软件企业软件项目风险管理过程的设计思路。有的研究如何基于过程改进对软件质量进行管理,如徐鹏等(2006)通过对软件性能工程使用范围受限和应用复杂等方面问题的分析,在综合多种传统软件过程模型特点的基础上,提出了一种将软件质量属性建模、分析和反馈融入软件过程的双瀑布软件过程模型,并详细介绍了双瀑布模型在软件过程中各个阶段的任务,重点说明了其改进及引入的建模样式、建模知识库等技术。杨律青(2009)在分析现有软件质量数学模型的基础上,提出了软件项目开发三维模

型框架,建立了软件质量最优化数学模型。模型将有限的成本和时间资源合理地分配到软件开发过程的各阶段,并使软件质量达到最优化,为项目管理者建立计划和分配资源提供量化依据,通过实例数据验证了建立的数学模型的正确性和有效性。有的研究人员对过程改进中的某一方面进行了研究,如王敏晰(2004)、杨崑(2002)、胡樱(2005)、王海阳(2005)分别对改进过程中的人员如何管理、人员能力怎样改进、改进中成本的估算和控制以及实现成本与管理的平衡等问题进行了研究。李明树等(2008)研究和讨论了软件缺陷预测技术的起源、发展和当前所面临的挑战,对主流的缺陷预测技术进行了分类讨论和比较,并提出了典型的软件过程缺陷预测模型。

当前软件的生产和软件项目的管理已经转变为以过程为中心,而软件过程改进已经成为软件过程研究的中心和热点,软件过程改进工程的提出和实施可以极大地提高组织级软件能力成熟度,对软件外包服务获得者来讲,可以对它的提供商进行客观现实的和预期的评价;对软件工程来讲,客观评价当前的或可能的软件开发能力,识别软件开发中各种行为并给他们安排合理的次序,这样可以提高软件开发水平规划软件过程改进行为;对软件开发商来讲,可以展示软件开发商的软件开发能力,形成自己优势,有利于开拓市场。为此软件过程改进是目前乃至将来国内外学者研究的方向。

1.3.2 软件过程建模国内外研究现状

软件过程建模(Software Process Modeling),它主要是通过特定的方法对软件过程进行抽象、表示和分析以增加对软件过程的理解,并通过直接或者间接的方式指导实际的软件开发活动,软件过程建模主要是为具体的过程改进活动提供方法和工具上的支持。软件过程建模的研究主要是围绕着过程建模方法和以过程为中心的软件工程环境(Process-Centered Software Engineering Environment, PSEE)展开的。

1.3.2.1 软件过程建模方法国内外研究现状

本书对软件过程建模方法领域最近10年的国内外主要研究进行了概括和分析,从一系列的相关研究中,选出来自20个会议(ACM Symposium on Applied Computing、International Conference on Software Engineering、Annual International Computer Software and Applications Conference、Asia-Pacific Software Engineering Conference等)和7种期刊(ACM Transactions on Software Engineering and Methodology、Automated Software Engineering、Information and System Technology、Journal of System and Software等)的100篇文献,作为对此领域研究现状总结的依据,从三个角度:①软件过程建模方法主要基于什么范式;②软件过程建模方法研究的主要目的集中在哪些方面;③软件过程建模方法的研究有哪些新的趋势,总体上概括和把握该领域的研究。

(1)从软件过程建模方法的范式角度研究。软件过程建模方法的范式(Paradigm)是指软件过程建模方法的基本类型,即通过什么方式对现有的软件过程建模方法进行分类。通过回答软件过程建模方法主要基于什么范式这一问题,可以了解该领域人员研究软件过程的基本方式;通过对比和分析,可以发现哪一种范式为更多的人所接受,或者能够解决更多为人们所关注的问题。

为了对软件过程建模方法进行系统的分析,人们提出了多种分类方式。根据建模方法的基本特点,Arbouï(2002)等人将其按照如下5个类别进行划分:逻辑(Logic)、过程(Procedure)程序、人工智能(Artificial Intelligence)、触发器(Trigger)、Petri网和面向对象(Object Oriented)。Montangero(1999)根据建模方法与计算机科学其他领域的

关系,则将建模方法划分为如下7个基本类型:项目管理(Project Management)、形式化规约语言(Formal Specification Language)、非形式化设计表示方法(Informal Design Notation)、编程语言(Programming Language)、数据库语言(Database Language)、工具集成机制(Tool Integration Mechanism)、工作流和群件(Workflow and Groupware)。Zmali在2001年和2004年的文章中根据过程建模语言的技术特点和对表示、演化、虚拟执行、评估和人力资源的支持程度,对现有的建模方法进行了分类。

上述分类方法主要是根据软件过程建模中所使用的理论工具,对建模方法进行分类。其主要优点是通过类别信息可以初步地把握一种建模方法的基本机制;而缺点主要在于,建模方法所使用的理论不胜枚举,因此上述分类方法很难准确和客观地覆盖所有的建模方法;而在进行建模方法选择时,人们主要是从自身的建模需求出发,因此,上述分类方法也很难为方法选择提供有意义的指导。本书提出了一种新的分类方法,首先根据软件过程的描述方式,将软件过程建模方法分为4个类别,每一类又划分为多种典型的实现方法。

首先,将软件过程建模方法分为如下4个类别,同时,每个类别又包含了一些典型的实现方法:

1) 静态关联的元素(Statically Connected Elements)。在基于该范式的建模方法中,软件过程被看作具有一定关系的元素(比如角色、活动和资源)的组合,而元素间的组成关系是静态。这类建模方法适合于表达软件过程的静态方面和软件过程元素在概念层次的联系,一般不涉及软件过程的执行。比较典型的实现方法包括:Object Management Group(2007)研究的面向对象的类图(Class Diagram, CD)、UML中的多种静态视图;Object Management Group 分别于2005、2006、2007年研究的软件过程工程元模型(Software Process Engineering Metamodel, SPEM)、元对象设施(Meta Object Facility, MOF)和本体(Ontology)以及Chen PP-S(1976)提出的实体关系图(Entity Relation Diagram, ERD)等。黄飞(2008)采用实体关系图对软件工程中的需求过程进行建模。蒋依欣等(2009)从分析以活动为中心的软件过程的特点以及活动之间的规则、关系和分类出发,通过UML活动图抓住度量关键域,简化了软件过程度量,同时避免了过程度量的盲目性。

2) 顺序执行的活动(Sequentially Executed Activities)。在基于该范式的建模方法中,能够描述软件过程的执行,但所描述的是一组顺序执行行为。比较典型的实现方法包括:Object Management Group(2007)研究的UML状态图(State Diagram, SD);Balust(2001)等研究的顺序关系(Precedence Relationship, PR)以及基于一般自动机的状态转换图(State Transition Diagram, STD)。朱贤成(2007)结合软件工程领域中的实际情况,提出了一种基于有限自动机的软件开发模型。该模型利用有限自动机理论,将单个软件活动行为映射为自动机的状态变迁,当软件项目更新后,用户只需开发相应的子服务组件或模块,便可进行系统扩充乃至系统重构。颜炯等(2005)定义了加入统计测试约束的UML用例图、序列图以及用例执行顺序关系,为基于UML的软件统计测试提供了一个形式化描述基础,在此基础上,给出一个从软件UML模型构造软件Markov链使用模型的算法,并给出了自动化支持工具UMGen的类图结构。

3) 并发交互的活动(Concurrently Communicated Activities)。在基于该范式的建模方法中,软件过程由一系列的并发元素组成,每个元素都有自己的内部行为;而元素之间通过通信或者同步实现相互间的协调。比较典型的例子包括:Object Management Group

(2007) 研究的 UML 活动图 (Activity Diagram, AD); Costin Badica 等 (2003) 研究的角色活动图 (Role Activity Diagram, RAD); Heimann 等 (1996) 研究的动态活动网 (Dynamic Task Net, DTN); 规约描述语言 (Specification and Description Language, SDL), 以及可以描述并发系统的形式化方法, 比如 Valk Rud (2003) 研究的 Petri 网 (Petri Nets, PN)、葛季栋等 (2008) 研究的对象 Petri 网 (Object Petri Net or Nets-within-nets, OPN); 任洪敏等 (2005) 研究的顺序通信进程 (Communicating Sequential Processes, CSP); Sangiorgi 等 (2001)、杨国伟 (杨秋松) 等 (2008) 研究的 π 演算 (π -calculus) 等。另外, 基于组件的软件过程建模 (Component Based Software Process Modeling, CBSPM)、并发状态转换图 (Concurrent State Transition Diagram, CSTD) 也可以归为这一类别。

4) 隐含的并发活动 (Implicitly Concurrent Activities)。在基于该范式的建模方法中, 软件过程的行为并没有被明确地表示出来, 而是通过其他方式推导出来。软件过程的行为是具有一定智能的实体之间通过协商或者通过集中逻辑推理的方式来产生的。比较典型的实现方法包括 Zambonelli F. 等 (2004) 研究的多 Agent 系统 (Multi-Agent System, MAS) 和 Belkhatir N. 等 (1994) 研究的基于规则 (Rule) 的过程执行。在软件过程仿真 (Software Process Simulation) 中, 软件过程一般被看作是一个宏观系统, 主要是通过过程的输入和输出来观察整个软件过程。比较典型的实现方法包括 Cassandras CG (2006)、张庆等 (2009) 研究的离散事件模拟 (Discrete Event Simulation, DES) 和 Kirkwood CW (1998)、吴明晖 (2007) 研究的系统动力学 (System Dynamics, SYDY)。

下面, 将分别论述每种范式中的主要实现方法以及每种实现方法的典型研究案例。

1) 静态关联的元素。对于软件过程元素间静态关系的描述, 人们主要采用非形式化的面向对象表示方法。Acuna 等 (2000) 通过面向对象类图中的 “is-a” 和 “be a part” 关联, 表示项目中角色、活动、产品之间的静态关系。UML 中的多种静态视图也可以用于描述过程元素间的静态关系。Jager 等 (1999) 使用 UML 包图 (Package Diagram) 表示软件过程的结构; Odeh 等 (2003) 把用例 (Use Case) 作为过程模型和系统模型之间的桥梁, 在两种模型之间建立联系。而对于描述模型语义和约束的元模型, 也被广泛地用于描述软件过程元素之间的静态关系: 李明树等 (2006) 采用扩展的软件过程工程元模型, 描述软件过程中活动、产品、角色之间的关系及其集成; Bendraou 等 (2006) 采用元对象设施描述建模语言 UML4SPM 的静态语义。此外, Ceravolo 等 (2003) 通过本体对 XP 中的概念和所涉及的实体进行建模, 以支持对相关概念的理解和实体的检索; Ahmed-Nacer 等 (2001) 采用实体关系图支持过程元素及相互间关系的表示、存储。

2) 顺序执行的活动。Cugola (1998) 采用基于有限状态自动机的状态转换图, 表示过程中制品的状态以及状态间的转换; Acuna 等 (2000) 采用状态转换图, 描述过程元素的内部和过程元素之间的顺序行为。Franch (1998) 和 Ribo (2001) 采用前趋关系表示活动间的前趋和后继关系。

3) 并发交互的活动。支持该类过程描述的建模方法, 一般与描述并发软件系统的形式化方法具有较为密切的关系, 前者往往是在后者的基础上, 根据软件过程建模的需求作适当的改进而提出来的。Podnar 等 (2000) 使用电信领域内广泛采用的规约描述语言, 对软件过程进行描述、仿真和验证。Bendraou 等 (2007) 和葛季栋、顾庆等 (2007) 分别采用 Petri 网、对象 Petri 网描述 SPEM 2.0 的语义和支持软件过程的多视图建模。对于