



普通高等教育“十一五”国家级规划教材
软件工程专业核心课程系列教材

丛书主编 孙家广 刘强

软件协同设计

赵逢禹 曹春萍 编著

- ◆ 教育部高等学校软件工程专业教学指导分委员会推荐教材
- ◆ 根据教育部“软件工程课程体系研究”项目成果《中国软件工程学科教程》及专业规范组织编写
- ◆ 与最新ACM和IEEE CCSE同步
- ◆ 汇集示范性软件工程专业教学成果

清华大学出版社





普通高等教育“十一五”国家级规划教材
软件工程专业核心课程系列教材

丛书主编 孙家广 刘强

软件协同设计

赵逢禹 曹春萍 编著

清华大学出版社
北京

内 容 简 介

本教材介绍了在软件协同开发中,如何建立开发团队,分配团队成员的角色与职责,在项目开发过程中团队成员之间的协调,计划的制定与进度的管理,需求的分析及软件的设计,软件产品的审查与质量的控制,测试与基于度量的团队过程改进等内容。通过实例指导学生如何管理团队、制定项目计划,如何应用传统的结构化分析方法、面向对象的分析方法解决实际应用问题。结合课程实践项目,学生可以掌握项目规划、工作量估算与分配、产品的质量缺陷跟踪度量与软件过程改进。

本书内容系统、翔实,适合作为高等院校计算机专业本科生教材,同时也适合从事软件开发与系统分析人员阅读。

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

软件协同设计/赵逢禹,曹春萍编著. —北京:清华大学出版社,2011.10

(软件工程专业核心课程系列教材)

ISBN 978-7-302-26435-4

I. 软… II. ①赵… ②曹… III. ①软件设计—高等学校—教材 IV. TP311.5

中国版本图书馆 CIP 数据核字(2011)第 163424 号

责任编辑:梁 颖

责任校对:胡伟民

责任印制:杨 艳

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:北京市清华园胶印厂

经 销:全国新华书店

开 本:185×260 印 张:12.5 字 数:280 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:23.00 元

产品编号:040523-01

前言

随着软件规模加大与软件项目开发管理越来越复杂,软件组织对软件从业人员的要求也随之增加。一个软件工程技术人员,除了具备专业知识与技术以外,还要求具有团队协作能力、交流沟通能力、文档写作表达能力、互相学习的能力、项目组织与管理能力、应用软件过程改进的能力等。在软件开发技术的许多课程中,主要培养学生的软件设计、编码与测试能力,而软件工程课程教授软件生命周期内使用的工程技术、辅助工具。但是这些课程在培养学生团队协作、交流沟通、过程管理与控制技术等方面存在明显不足。本教材根据软件协同开发教学的特点,将团队的规模设定为5~6人,项目也通常限制于15~20人。在团队中每名成员都担任一个经理角色,而每名经理又都是一个设计开发工程师。通过这么一个特定的团队协同开发项目,使每个成员熟悉项目计划的制定、每名成员工作规划和工作量的分配与平衡、每项工作的质量缺陷跟踪度量、个人工作的生产率与缺陷。通过这个协同开发实践,总结出在管理与技术层面的不足,并为进一步的改进找到方向。

本教材遵循工业界软件开发实践并根据软件协同开发过程组织教学内容。第1章介绍了软件开发过程、团队开发的实践与协同软件设计的过程。第2章给出了高效团队所应具备的特征,阐述了团队成员的角色及其目标,介绍了团队成员与开发步骤的对应关系,并就团队成员间如何进行良好的沟通进行了探讨。第3章根据软件项目的特点介绍了项目规划的必要性,分析了项目规划的内涵及制定过程,介绍了项目规划中产生的项目范围说明书和项目章程以及项目范围说明书的撰写方法与内容。通过案例向读者介绍了项目章程的制定和项目范围说明书的编写。第4章介绍了计划对项目开发的作用及重要性,并重点介绍了软件协同开发的计划过程以及脚本的编制;最后就如何编写进度计划、质量计划,如何进行工程跟踪进行了讨论。第5章首先分析了需求开发阶段的主要工作,包括获取用户需求、制定需求优先级、建立业务规则、制定通用词汇表、获取外部接口需求、制定非功能性需求、撰写软件需求规格说明、需求确认;同时描述了需求开发阶段相关角色、交付物以及交付物的模板;最后给出了相关案例分析。第6章论述了设计原则和设计过程,分析了设计阶段所关注的系统总体结构。讨论了软件设计说明书(SDS)的制作以及高层设计的文档撰写。第7章描述了产品实现过程,包括详细设计与编码、缺陷与预防、详细设计审查与代码走查、部件质量复核等内容,介绍了依据产品总体设计开发产品的流程。第8章介绍了协同软件开发中软件测试测略与开发过程的集成,在软件开发的不同阶段,测试角色所具有的工作目标和任务。描述了集成测试策略、系统测试策略、测试

计划的制定、测试用例与测试说明书的撰写规范、测试数据收集以及测试分析等概念。介绍了软件协同开发的集成与系统测试草案。第9章介绍了项目的计划跟踪与度量对团队协同开发项目的产品质量形成、项目管理过程以及团队软件过程改进的意义。通过对开发进度、开发效率、产品的缺陷密度、引入缺陷的比率等质量指标,跟踪并分析质量问题的来源,识别造成质量缺陷的系统因素,采取相应的措施予以排除,使软件开发过程更加稳健,进而提高开发过程的能力。

作者于2004年首次为软件专业学生开设协同软件设计课程,经过5年的实践与探索,在教学内容、课程组织、课程实践等方面积累了较丰富的经验。从2008年起,该课程在计算机科学与技术专业本科生中开设,取得了良好的效果。参与本书编写的教师要么是该课程主讲,要么是该课程中团队开发指导教师。因而,教材在内容组织上,力求针对计算机专业本科生的能力与知识,重点围绕软件项目开发中管理过程、计划与任务分配、高质量的产品设计、团队的协作与交流等薄弱环节组织教材内容,并通过一个具体的软件项目,指导学生进行初步需求分析、项目规划、质量规划、任务分解,最后制定详细的开发计划。根据开发进度计划,指导并驱动项目分析、设计与实施,并根据质量计划,采用产品复审技术,指导开发团队审核每个阶段的产品,并采取措施以保证其符合质量计划。最后通过项目的事后分析与讨论,找出团队开发活动的薄弱环节与缺陷,然后制定改进措施予以改进。

赵逢禹教授编写了本书的第1、8章,并负责全书的统稿;曹春萍副教授编写了第3、4章;欧广宇编写了第2、9章;赵海燕编写了第5章与附录A;郝聚涛编写了第6章;霍欢编写了第7章。

本书可作为普通高等院校计算机科学与技术专业、软件工程专业“软件协同设计”课程的教材,也可供从事相关工作的技术人员参考。由于本教材涉及的内容较多,各学校的教学情况也不尽相同,因此在组织教学时,教师可以适当调整学习内容。

由于作者水平有限,难免有错误与不当之处,恳请读者见谅并不吝赐教。

编 者

2011年7月

目 录

第 1 章 绪论	1
1.1 软件过程与过程管理	1
1.2 软件协同开发团队	4
1.3 软件协同开发的原理、阶段与流程管理	5
1.3.1 软件协同开发过程原理	5
1.3.2 软件协同开发流程	6
1.3.3 软件协同开发的主要阶段	8
1.4 软件协同设计的目标与课程组织安排	10
1.4.1 软件协同设计的目标与评估方法	10
1.4.2 课程的组织与安排	11
第 2 章 软件协同开发中的团队建设	13
2.1 高效团队应具备的特征	13
2.2 团队成员角色及其目标	16
2.2.1 客户接口经理的职责	17
2.2.2 设计经理的职责	18
2.2.3 实现经理的职责	18
2.2.4 测试经理的职责	19
2.2.5 计划经理的职责	20
2.2.6 过程经理的职责	21
2.2.7 质量经理的职责	22
2.2.8 支持经理的职责	23
2.3 团队成员与开发步骤的对应关系	24
2.4 团队的协作与沟通管理	26
2.4.1 沟通的效率和要素	27
2.4.2 促进沟通的方法	28
2.5 小结	29
第 3 章 软件协同开发的总体规划	30
3.1 项目的规划	30

3.2	软件项目的初步范围	31
3.2.1	软件项目背景	31
3.2.2	项目利益相关者	32
3.2.3	项目的初步范围的确定	33
3.3	确定项目预算	34
3.3.1	自底向上的成本估算	34
3.3.2	自顶向下的成本估算	36
3.3.3	完成预算	36
3.4	制定 IT 项目章程	37
3.5	小结	38
第 4 章	软件开发计划	40
4.1	定义软件开发计划	40
4.2	项目的工作分解	41
4.2.1	定义产品范围	41
4.2.2	定义工作范围	42
4.3	项目的进度计划	44
4.3.1	制定项目进度计划	45
4.3.2	项目进度计划的表示	47
4.3.3	软件项目进度计划案例——校务通管理系统	48
4.4	项目的成本计划	50
4.4.1	项目成本的估算方法	50
4.4.2	项目成本估算案例	54
4.4.3	制定项目的成本计划	55
4.5	软件项目质量计划	55
4.5.1	项目质量计划的概念	56
4.5.2	项目质量计划的制定方法	57
4.5.3	项目质量的度量指标	58
4.5.4	项目质量计划的表示	59
4.6	协同设计的计划过程	59
4.7	项目开发计划书	61
4.8	小结	63
第 5 章	需求开发	64
5.1	需求的获取	64

5.2	需求开发的流程	65
5.3	撰写需求报告	67
5.4	需求开发的支持工具	71
5.4.1	数据流图	71
5.4.2	业务流程图	73
5.4.3	活动图	75
5.4.4	用例图	75
5.5	需求的复审	78
5.6	需求开发案例分析	79
5.7	小结	84
第 6 章	软件的总体设计	85
6.1	软件设计的重要性	85
6.2	总体设计的任务及原则	86
6.2.1	软件总体设计的任务	86
6.2.2	总体设计需遵循的原则	89
6.3	软件总体设计方法	90
6.3.1	结构化设计方法	90
6.3.2	面向对象的设计方法	95
6.4	设计的复核与检查	99
6.5	协同设计	101
6.5.1	协同设计应注意的问题	101
6.5.2	协同软件开发设计	102
6.6	小结	105
第 7 章	软件的详细设计	106
7.1	详细设计的概念	106
7.1.1	详细设计的目标	106
7.1.2	详细设计的任务	106
7.2	详细设计的方法	108
7.2.1	结构化详细设计	108
7.2.2	面向对象的详细设计	112
7.3	详细设计的标准和原则	115
7.3.1	详细设计标准	116
7.3.2	复用性原则	117

7.3.3	一致性原则	118
7.3.4	可用性原则	119
7.3.5	可测性(testability)原则	119
7.4	详细设计审查	120
7.4.1	详细设计审查策略	120
7.4.2	详细设计审查方法	120
7.5	设计举例	121
7.5.1	结构化详细设计	121
7.5.2	面向对象详细设计	124
7.6	详细设计报告	126
7.6.1	详细设计说明	127
7.6.2	详细设计审查报告	129
7.7	小结	130
第 8 章	软件测试计划与测试管理	131
8.1	软件测试的阶段划分	131
8.1.1	单元测试	131
8.1.2	集成测试	132
8.1.3	确认测试	132
8.1.4	系统测试	133
8.2	测试角色	134
8.3	测试策略	136
8.3.1	集成测试策略	136
8.3.2	系统测试策略	138
8.4	测试计划	139
8.4.1	测试设计说明	142
8.4.2	测试用例	142
8.4.3	测试日志	143
8.5	测试分析	143
8.5.1	发现有缺陷倾向的模块	143
8.5.2	追踪缺陷数据	144
8.5.3	测试总结	144
8.6	小结	144
第 9 章	计划跟踪与过程改进	146
9.1	项目计划与跟踪过程	146

9.1.1	项目跟踪及控制方针	146
9.1.2	项目跟踪及控制的内容	147
9.1.3	项目跟踪及控制中人员职责	147
9.1.4	项目跟踪活动	148
9.2	软件度量过程与度量指标	152
9.2.1	软件度量过程	152
9.2.2	软件度量的项目及其指标	153
9.2.3	软件度量面临的问题	155
9.2.4	软件度量注意事项	156
9.3	基于度量的软件过程管理	156
9.3.1	过程管理的职责	156
9.3.2	过程管理和项目的关系	157
9.3.3	基于度量的软件过程改进	158
9.4	小结	159
附录 A	EA 工具简介	161
A.1	EA 操作界面简介	161
A.2	新增一个 EA 项目	162
A.3	在 EA 中绘制活动图	164
A.4	在 EA 中绘制用例图	166
A.5	在 EA 中绘制类图	167
A.6	在 EA 中绘制序列图	172
A.7	在 EA 中绘制通信图	173
A.8	在 EA 中绘制对象图	175
A.9	在 EA 中绘制状态机图	176
A.10	在 EA 中绘制时间图	177
A.11	绘制包图	178
A.12	绘制交互概述图	179
A.13	绘制组合结构图	181
A.14	绘制组件图	184
A.15	绘制部署图	184
A.16	小结	187
参考文献		188

第1章 绪 论

在过去的几十年中,软件产业经历了从软件作坊生产向大规模工业生产的转变。在早期的软件开发项目中,软件是惟一交付的产品。由于软件与项目规模不大,通常单人就可以完成,无须考虑团队合作。随着软件规模不断增大与复杂性不断增加,软件开发项目在管理、产品质量与生产率等方面出现了许多问题,项目的失败率要远高于其他工程项目。遗憾的是,直至目前我们仍然看不到任何突破性的理论方法与技术能够很好地解决这一问题。因而,必须通过提高成员的工作能力、工作质量与团队管理水平,使软件产品的生产过程更加稳定且具有统计意义上的可预测性,通过借鉴其他领域成功的工程实践的方法、工具与管理,解决软件开发项目中的问题。

1.1 软件过程与过程管理

北约(NATO)的科技委员于1968年召集了近50名一流的编程人员、计算机科学家和工业界巨头,讨论和制定摆脱“软件危机”的对策。在那次会议上第一次提出了软件工程这个概念,试图借鉴工程化的技术与规范解决软件开发中的问题。

软件工程是一门研究如何用系统化、规范化、数量化等工程原则和方法去进行软件的开发和维护的学科。软件工程包括两方面内容:软件开发技术和软件项目管理。软件开发技术包括软件开发方法学、软件工具和软件工程环境。软件项目管理包括软件度量、项目估算、进度控制、人员组织、配置管理、项目计划等。

对许多软件组织,如何在计划规定的时间与预算的成本内开发出满足质量要求的软件产品是一个重大挑战。Standish Group 2009年的报告显示,软件项目的成功率为32%(即按照预算的费用、计划的交付时间、开发出满足功能与质量要求的软件产品),44%项目在完成时间、开发费用或产品的功能与质量方面存在问题,24%的项目在完成或应用之前被取消。Standish Group 信息主管 Jim Crear 指出,相比几年前的统计结果,项目的成功率有一定的下降,失败率则在上升,2009年的项目失败率是过去十年中最高的。

软件项目失败涉及许多因素,如需求的动态变化与不完整性、缺少用户参与、开发团队缺乏凝聚力、不现实的开发目标、缺少管理层的支持、开发过程混乱等。DeMarco 曾指出,项目的成败很少是技术上的问题,你几乎不用问自己编程技术是不是足够先进,因为它肯定是的。当项目中出现问题时,通常不是技术不足引起的,而是由于人的交流不畅、协同开发过程与团队出现了问题而引起的。

现代软件工业通过学习并参考在制造业成功应用的持续过程改进的方法,提出了适合软件行业的过程规范。在制造业中,过程(process)是把输入转变成对客户有用的输出的活动集合,是经过长期实践并证明是行之有效的操作规程。实际上,过程的概念适用于机构的组织活动与服务过程,如医疗过程、公共服务、教育、IT 项目、建筑工程等。IEEE-STD-610 把过程定义为达到某个目标而执行的步骤序列。在软件成熟度模型(Capacity Maturity Model,CMM)中,软件过程定义为在软件的生产和演进中的一组活动、方法和实践。它是一个软件组织拥有的、为成功完成软件项目所规定的一个公共活动框架。该活动框架把一些方法、工具和开发人员应用于软件项目的活动中,它识别角色、指定任务、建立度量,并给出主要步骤的入口与出口准则。一个明确定义的过程可以帮助软件开发组织确保每一个活动项目都安排了合适的责任人,可以通过对工程的跟踪发现问题并予以改进。该过程框架为开发团队的学习与提高提供了一个基础,当发现了当前的过程中存在问题或找到了更好的方法时,就可以把它结合到过程框架中。这样能帮助新的团队在借鉴前辈经验的基础上迅速取得进步。

从 20 世纪 70 到 90 年代,以质量与过程改进为中心的运动在日本与欧美广泛开展,实践也证明基于过程能力与过程持续改进的思想对产品质量的提高具有重要成效。Humphrey 把这些质量原理应用到软件开发过程,提出软件开发与制造业中组装和制造工艺流程一样,高质量的产品依赖于高质量的过程。在他的《管理软件过程》以及他在卡内基·梅隆大学软件工程研究所(Software Engineering Institute,SEI)的早期研究中,把 Deming 与 Crosby 的过程能力与过程持续改进的思想运用到软件组织中,提出了过程成熟度模型(Process Maturity Model,PMM)。对 PMM 进一步研究与提炼,形成了 CMM 能力成熟度模型。尽管开发 CMM 的主要原因是由于美国国防部(DOD)需要一个评估方法对软件承包商的能力进行度量与评估,但它可用来指导软件组织评价自身的软件过程能力、识别薄弱环节,是指导组织如何改进的路线图。

自从 1994 年 SEI 正式发布软件 CMM 以来,相继又开发出了系统工程、软件采购、人力资源管理以及集成产品和过程开发方面的多个能力成熟度模型。虽然这些模型在许多组织中都得到了良好的应用,但对于一些大型软件企业来说,可能会出现需要同时采用多种模型来改进自己多方面过程能力的情况。一个组织使用多个模型,随之也带来了一些问题。首先,不能集中其不同过程改进的能力以取得更大成效;其次,要进行一些重复的培训、评估和改进活动,因而增加了许多成本;第三,不同模型中对相同事物的说法不一致,或活动不协调,甚至相互抵触。SEI 在总结 CMM 实践经验并集成各种 CMM 衍生模型的基础上,在 2000 年推出 CMMI(Capability Maturity Model Integration):集成的能力成熟度模型/能力成熟度模型集成。

CMMI 模型的表达方式有 2 种,阶段式与连续式。连续式与阶段式所包含的过程域是完全一致的,两者的区别主要在于过程域的组织方式不同,阶段式是用来描述组织整体上的成熟度,而连续式关注的是组织单个过程域的能力。如果组织注意力

主要集中在某几个过程域上,则采用连续式比较合适。SEI 建议在多数情况下采用 CMMI-SE/SW 作为组织过程改进的模型。

CMMI-SE/SW 的阶段式表示法中,提供了一个软件组织的软件过程能力成熟度级别框架,该级别框架根据能力由低到高划分为 5 级,如图 1.1 所示。它们分别是初始级、已管理级、已定义级、量化管理级和优化级。CMMI 从过程域所有的实践提炼出了多个过程域所共有的实践,称为一般实践,将其目标称为一般目标,其余特定于某个过程域的实践与目标称为特定实践与特定目标。这样模型取得了相对 SW-CMM 更高的抽象度与适应范围。目标(一般目标与特定目标)首次作为模型构成成分出现,这表明 CMMI 对过程活动的结果投入了更多的关注。

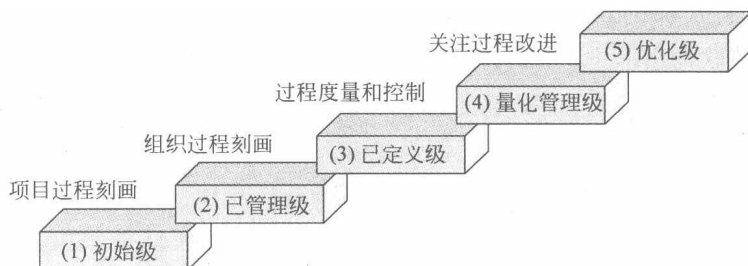


图 1.1 CMMI 的成熟度与过程能力

对于图 1.1 中的过程能力 2 级以上的每个成熟度水平,CMMI 都规定了达到此水平的过程域。每个过程域都定义了所应达到的特有目标、通用目标,以及实现特有目标的特有实践。对于通用目标,又进一步划分公共特性和通用实践。这种基于过程域的最佳实践活动的过程改进方法为组织达到成熟度级别定义了评判标准。图 1.2 给出了阶梯式能力水平的过程域。

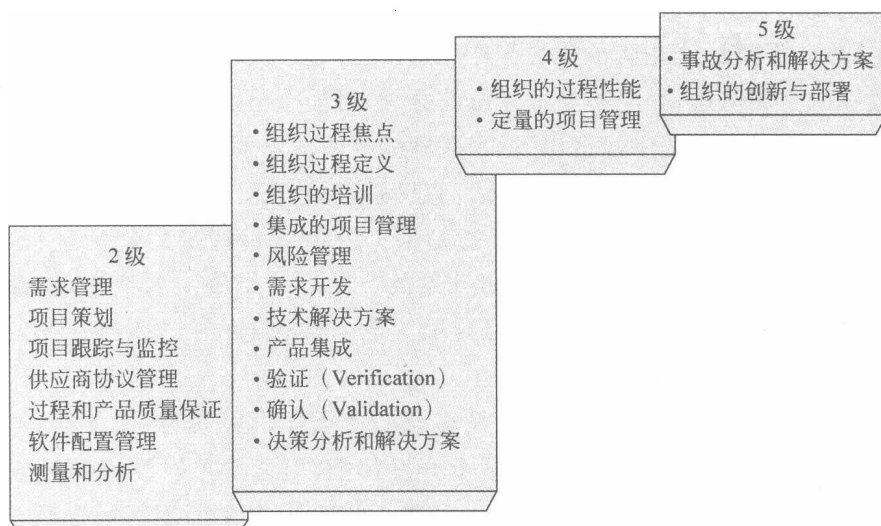


图 1.2 CMMI 过程域

Standish Group 的研究表明,从统计数据来看,IT 项目已经有所改进。成功的 IT 项目数量比例已经从 1994 年的 16% 增长到 2002 的 34%。失败项目的数量比例从 1994 年的 31% 减少到 2002 年的 15%。成功项目的增加有多种原因:有了更好的项目进程的监控工具,掌握了更好的管理过程,有了更有经验的项目经理,认识到 IT 项目管理是一个过程等。

1.2 软件协同开发团队

当前的软件产品的规模越来越大,软件涉及的领域越来越广,软件的实现技术日益复杂,从而使软件组织中形成了许多开发团队,如用户接口、需求分析、架构设计、开发与测试、技术支持与培训、过程与质量管理等团队。如何高效地组织安排这些团队的工作、度量其工作与产品质量,如何使团队的过程活动能力不断提高,从而使团队在预计的时间与费用内开发出符合质量要求的产品,是当前软件组织追求的重要目标。

微软的 Windows 2000 项目是一个成功开发实例,该项目有 2500 名全职的员工,他们被分成了 500 多个团队。为了监控与管理项目进展,设立了一个会议室,被称之为“作战室”,实际上是项目管理中心。项目的关键人员经常在这里会面,根据项目的进展情况有时一天的会晤达三次之多。会议的目的,往往是为了研究新出现的重大问题,及时予以决策并集中兵力解决。这样,就能够审查项目所取得的进展及所面临的挑战,并让每位成员及时掌握项目现状。

构造一个高效的团队是一个长期复杂的过程。团队中队员的个人能力与技巧只是一个方面,更为重要的是团队必须有一个明确定义的、所有队员都必须理解与自觉遵守的过程规范。例如,组建一个篮球队时,如果部分队员是橄榄球选手、足球选手、手球选手,球队需要让这些队员学习篮球规则。在很多情况下,把个人能力都很优秀的队员放在一个团队中,如果不经系统的合练,可能会形成一个较差的团队。这种现象在足球与篮球的比赛中经常出现,当联赛的冠军队与明星联队比赛时,取得胜利的通常不是个人技术更突出的明星联队,而是团队协作更好的冠军队。

一个团队可能是自然形成的,也可能是根据需要通过特定的创建过程形成的。创建团队的第一步是让每个成员对即将开发的产品有一个统一的认识,在此基础上对产品进行定义与规划。在对目标产品定义完后,成员们再对产品的开发战略与开发计划达成一致意见。

项目开始时,团队成员都不知道产品将会是什么样,也不知道该怎样去完成它。虽然他们还不能一致同意某个产品或某个开发计划,但通常可以就目前的未知因素与问题,以及如何澄清这些问题得到统一意见。然后他们采用下面的循环步骤:确认混淆和不一致的意见,就如何解决这些问题达成共识,再逐一解决它们。当他们的

理解趋于一致时,对开发过程和将要开发的产品的细节认识也就同时达成了一致。

微软在总结了其成功的项目实践经验后,提出了一个微软解决方案框架(Microsoft Solution Framework,MSF)。MSF的目标是为企业的IT系统的规划、建设与管理提供工作模型与开发准则。MSF认为,技术并不是项目成功与否的惟一决定因素。在一个成功的IT项目中,开发人员、开发过程以及风险管理等因素都起着至关重要的作用。项目组如何有预见地、持续地管理与控制项目风险,如何在项目执行过程中有效地进行协作与沟通,如何保证技术方案与用户需求的一致,这些问题都直接关系到IT项目的成败,它们通常也是项目组最难把握、最难处理好的关键问题,大多数IT项目失败的主要原因也在于此。在MSF中,针对这些关键问题,设计了团队管理、过程管理与风险管理等模型,这些模型可以为项目组提供有效的指导。

可以把团队理解为由一组人员组成的,在一个协作与交流的环境中,为了一个共同的任务或目标,按照事先定义的过程运作的实体。为了便于交流、管理与快速决策,团队的规模通常不大,一般为5~6人。正如比尔·盖茨所说的那样,只有在那些有着严格的经费预算和确定的时间期限的项目实施中,团队成员才能更好地充分协作、拥有最高的生产率与产品质量。尽管对如何建立高效的软件团队没有太多的专门研究,但团队凝聚力、成员交流与共同的工作框架是必不可少的。

1.3 软件协同开发的原理、阶段与流程管理

团队协同开发被设计为完成复杂的、有创造力工作的最有效的方法。当团队成员有非常好的合作交流时,团队将会非常高效;反之,团队如果缺乏明确的管理制度、充分的交流与反馈、必要的支持与引导,团队内部以及团队间各自为战,极有可能导致在设计方案、产品质量、开发进度与费用上的差错,进而导致整个项目的失败。

1.3.1 软件协同开发过程原理

团队软件开发过程(Team Software Process,TSP)是SEI于2000年提出来的适用于小型软件团队进行软件工程过程改进的模式。小型软件团队的人员规模通常在20人左右,项目持续时间不超过1年。团队软件过程是一个良好定义的构建和管理团队的最佳实践,指导跨功能团队中的成员如何有效地规划和管理所面临的项目开发任务,告诉管理人员如何指导协调软件开发团队。团队软件过程包括一个完整定义的团队作业过程、已定义的团队角色、一个结构化的启动与跟踪过程;它被用来解决软件开发团队所面临的项目承诺、过程控制、产品质量和团队合作等问题,事实证明它在利用团队成员技能、按照计划的时间进度和承诺费用生产高质量的产品方面非常有效。

软件协同开发过程是一个简化的团队软件开发过程。在该过程中,团队的规模

减少到 5~6 人,项目也通常限制于 15~20 个人月。在团队中每个成员都担任一个经理角色,而每个经理又都是一个设计开发工程师。通过这么一个特定的团队协同开发项目,使每个成员熟悉项目计划的制定、每个成员工作规划和工作量的分配与平衡、每项工作的质量缺陷跟踪度量、个人工作的生产率与缺陷。通过这个协同开发实践,总结出团队与个人在管理与技术层面的不足,并为进一步的改进找到方向。

1.3.2 软件协同开发流程

对于软件项目或项目中的每个项目阶段,如目标定义阶段、计划阶段、设计阶段等,都会经历以下 5 个过程,它们分别是启动过程、计划过程、执行过程、监控过程以及收尾过程。

1. 启动过程

对于任何软件项目或者项目阶段,必须有人启动活动以保证项目团队完成所有的工作,为团队分配资源、确定项目经理以及关键的团队成员。在项目启动后,项目经理与团队成员共同完成概念设计,制定项目章程。项目章程包括项目名称、项目经理、经费预算、完成时间、项目目标、角色与职责等。

2. 计划过程

计划过程包括制定和维护一个可执行的计划,以保证项目的目标得以实现。通过计划过程,可以产生不同方面的项目计划,如范围管理计划、进度管理计划、成本管理计划、质量计划和采购管理计划等。

3. 执行过程

协调人力和其他资源来执行项目的计划,以产生项目或项目阶段的产品、服务或结果。执行过程的例子包括组织项目团队、指导和管理团队、执行质量保证、信息发布等。

4. 监控过程

通过对项目相关指标的测量,如质量指标、进度指标、资金使用等,监控项目进程以保证项目团队能够达到项目目标。项目经理和项目成员监视和测量偏离计划的过程,分析造成问题的原因,在需要的时候采取正确的行动。

5. 收尾工程

对项目或项目阶段的正式验收,并有效地终止项目。该过程的关键结果是工作的正式验收和结束文档的撰写。例如,完成最终的项目报告和教训教训分析报告。

以上 5 个项目管理过程,适合所有领域项目管理。例如电影的制作,其制作过程可能包括编剧(启动)、制作(计划)、表演和导演(执行)、监督(监控)和将电影发行到市场(收尾)。

在大多数软件企业的开发管理体系里,瀑布模型与基于增量的敏捷模型是两种应用范围最广的过程模型。软件协同开发过程有效地结合了敏捷模型与瀑布模型的优点,首先通过多个开发周期来完成产品,周期 1 从教师布置的产品目标开始,然后

按照下面的 7 个步骤进行：项目启动、计划、需求、设计、实施、测试和最终检查。在周期 2 中,重复同样的步骤,只是把周期 1 中的基本产品进行了升级。因而,软件协同开发过程是从一个最小规模的产品版本开始,使用循环的过程开发产品。开发周期的数量由教师确定,但每个周期的内容由团队根据每个成员承诺的时间、成员的技术水平共同商讨决定。在开发过程中,一个基本原则是每个成员按时按质完成自己承诺完成的任务,团队对每个人的任务执行数据进行跟踪。通过一个开发周期的活动,每个人分析自己的成功与失败,找出进一步改进的目标并在下一个开发周期中实施。

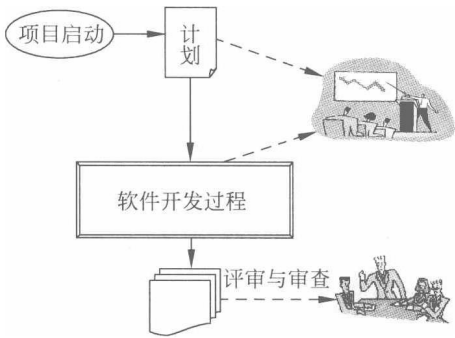


图 1.3 软件协同开发流程图

图 1.3 给出了项目启动、计划、软件开发过程的执行(完成需求、设计、实施、测试)与每个阶段工作产品审查的流程图。表 1.1 给出了每个步骤的主要活动。

表 1.1 软件开发周期的活动

顺 序	步 骤	活动(第 1 周期)
1	项目启动	• 建立项目团队、成员角色 • 确立项目的范围与边界,完成项目的概念设计 • 做出规模与风险分析
2	计划	• 完成团队进度计划 • 完成成员计划 • 完成产品质量计划
3	需求	• 与教师交流,获取产品的详细需求 • 完成系统测试计划 • 撰写支持材料与文档
4	设计	• 完成产品的总体功能设计 • 完成产品的详细设计 • 完成单元测试计划 • 编写测试用例与支持工具 • 完成设计审查 • 撰写设计报告
5	实施	• 完成编码 • 完成代码审查并记录审查结果
6	测试	• 完成单元测试并记录测试数据 • 完成集成与系统测试 • 分析测试结果
7	总结	• 撰写质量分析报告 • 写出周期报告 • 完成团队与成员角色评估