

软件开发组织的沟通与协调

杨 崑 王二平

(中国科学院心理研究所, 北京 100101) (中国科学院研究生院, 北京 100015)

摘 要 在对软件开发中人员与组织问题进行回顾的基础上, 该文首先就软件开发组织沟通与协调的重要性进行分析, 然后对软件开发组织中的沟通与团队结构、沟通与项目成员、沟通频次分布、沟通方式等问题, 以及整体协调、外部协调、内部协调等问题进行阐述。最后就今后软件开发组织沟通与协调问题可拓展的研究方向进行了讨论。

关键词 沟通, 协调, 软件开发, 组织心理学。

分类号 B849 : C93

1 人员与组织因素

计算机软件产品的开发是信息技术的一个重要方面, 也是构成信息产业的主要部分之一。随着软件产品的规模不断扩大, 复杂程度不断提高, 软件开发出现了大量问题, 被称为“软件危机”(software crisis)^[1]。从组织心理学的角度来看, 软件开发是一种典型的聚合型作业——成员虽然形式上独立完成分配的作业, 但最后聚合为完整的作业。最终作业对各部分作业的技术标准和完成时限有要求, 形成成员活动的相互制约。因此, 软件开发的组织心理因素显得尤为重要。Ewusi-Mensah 和 Przasanskyd 对开发失败的软件项目进行研究表明, 管理是影响软件研发项目全局的因素, 而技术只影响局部^[2]。Doherty 和 King 等人在对大型软件组织的调查中发现, 软件开发中的组织问题甚至比技术问题更为重要^[3]。

软件开发人员与组织因素的研究始于 20 世纪 70 年代。Weinberg 认为软件编程既是个体活动也是社会活动。作为个体活动的编程行为包括: 智力、问题解决能力、动机、经验等问题; 作为社会活动的编程行为包括编程群体、团队和项目的组成与协调沟通等问题^[4]。Brooks 认为软件的复杂度、不一致性、可变性和不可见性是软件的本质困难, 导致人员沟通困难、产品瑕疵、成本超支和进度延迟^[5,6]。鉴于此 Boehm 提出“软件生产率改进机会树”(software productivity improvement opportunity tree), 将人员和组织因素列为提高软件生产率的首要机会, 包括人事安排、工具配备和管理三个要素^[7,8]。1976 年 Newman 以《人件系统》为名探讨了软件开发中的人员风险, 第一次提出与软件、硬件相对应的“人件”(Peopleware)概念^[9]。1987 年 DeMarco 和 Lister 出版了《人件: 富有成果的项目和团队》, 引起业界对这一概念的重视。作者指出, 知识工人需要思考, 每个人都是独特的和不可替代的; 强调工作环境对软件生产率有重大影响; 提倡团队成员的互补性, 认为给知识工作者以自由和信任最

收稿日期: 2004-01-10

通讯作者: 王二平, E-mail: wangep@psych.ac.cn

有可能形成优秀的团队^[10,11]。此后, Constantine 扩大了“人件”概念, 将软件的使用者也包括进来^[9]。卡内基梅隆大学软件工程研究所(SEI/CMU) 综合软件开发人员与组织问题研究成果, 推出软件能力成熟度模型(Software Capability Maturity Model, SW-CMM)^[12]和人员能力成熟度模型(People Capability Maturity Model, P-CMM)^[13,14], 对软件开发过程和人员管理活动进行规范。

2 沟通与协调的重要性

在软件开发的各人员与组织因素中沟通与协调问题是影响软件生产效率和可用性的重要问题。沟通(Communication) 是指人们分享信息、思想和感情的任何过程。狭义的沟通仅指信息的交流, 是使信息发挥积极作用和达到目标的手段。Brooks 指出沟通是软件开发计划中必须考虑的工作量: 人数 n 的变化, 将引起沟通路径 $n \times (n-1) / 2$ 倍的变化。这使得软件开发中人员与时间两个要素不能进行线性变换而是呈现出错综复杂的关系^[6]。此后, 一些实证研究也说明了沟通对软件开发的重要性。Curtis 用访谈法研究了 17 个大型软件开发项目, 沟通和协调中断是导致软件项目失败的主要原因之一^[15]。McConnell 通过文献分析, 总结出软件开发的 12 项典型错误, 其中和沟通问题相关的有 4 项: (1) 嘈杂的办公环境不利于沟通, (2) 领导与问题雇员沟通不畅, (3) 用户信息输入不足 (4) 项目后期加入人员, 增加了明确共同目标和知识所需的沟通成本^[16]。Sonnetag 对德国 29 个软件项目中优秀人员的必备要素进行调查, 发现社会技能(主要指沟通能力) 仅次于专业技能成为优秀软件人员必备的第二大要素^[17]。在随后的研究中, Sonnetag 对优秀软件人员的行为特征进行分析, 结果表明: 优秀软件人员的绩效与对沟通活动的投入程度有显著相关。在同事的表述中, 优秀软件人员有较高的社交能力; 优秀的软件人员自身也把沟通作为完成任务的重要策略; 他们投入团队磋商和评审会并且积极从同事那里获得反馈的实际行为也比一般人员要多^[18]。

协调(Coordination) 是指“对相继的和同时的互依活动进行调和的过程”^[19]。狭义的协调仅指“对完成目标各活动间的相互依赖关系进行管理”^[20]。通用领域常见的依赖关系有以下几种: 先决条件; 资源共享; 同时性。Toffolon 和 Dakhli 从软件开发活动复杂的背景分析其协调的重要性, 认为: 首先, 软件开发组织从事的商业活动是持续变化的; 其次, 软件开发活动是在一个包含技术、经济、人员和组织等因素的大型环境中进行的; 再者, 软件开发活动本身是在一系列相互依赖过程的基础上进行的^[21]。McConnell 通过文献分析总结出的 12 项软件开发典型错误中, 与协调问题相关的有 3 项: (1) 软件开发没有明确的目标和动机 (2) 随意取消开发的上游活动, 如为了增加编码时间而压缩甚至取消设计活动 (3) 软件质量保证(SQA) 部门协调能力薄弱^[16]。Brooks 指出, 软件开发过程是一个熵不断增加的过程^[6]。因此, 软件开发需要大量的协调活动, 从外界导入负熵流。

3 沟通问题

3.1 沟通与团队组织结构

Schneiderman 将软件开发团队概括为三种类型: 传统团队、无我开发团队和主程序员制

团队^[22]。这三种团队的沟通形式如图 1 所示（双向箭头表示信息的沟通）。Schneiderman 认为，上述三种开发团队结构在沟通方面各有千秋。传统团队只允许必要的人际沟通，比较适合项目本身就是层次结构的大型开发项目。无我的团队成员沟通十分充分，这种团队被认为适合于研制时间长、开发难度大的项目。而主程序员制团队突出主程序员的领导，属于集中领导形式，主程序员是沟通的核心，能否取得好的效果，很大程度上取决于主程序员的技术水平和管理才能。



图 1 沟通与团队组织结构

随着对软件产品可用性(Usability)的重视 ,用户成为日益受到关注的信息源。Castka 和 Bamber 在总结软件开发的团队策略时指出最好的创意通常来自客户或用户，他们认为 Kidd 在 1994 年提出的“人机综合制造”结构 (Human And Computer Integrated Manufacturing ,

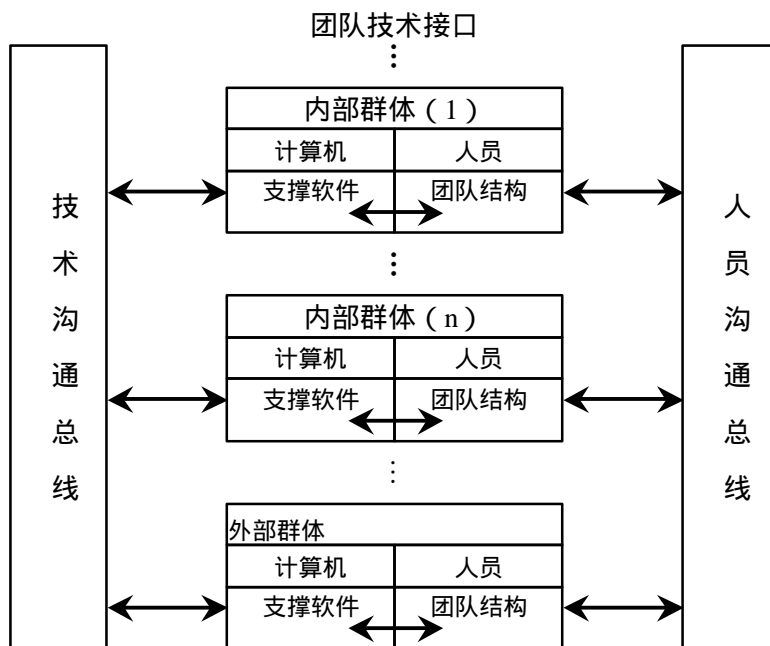


图 2 人机综合制造结构

HCIM),正是考虑了与客户沟通的因素实现快速制造范例的团队策略。“人机综合制造”结构如图 2 所示^[23],这样的结构能够使企业实现组织内部围绕流程形成自然群体,并提供了将企业外部的用户、客户、供货商包含进来的空间。HICM 结构强调了这些群体之间的沟通(人员和技术沟通总线),同时描绘了人员与技术之间的接口。

3.2 沟通与项目成员

Brooks 认为“向进度落后的项目增加人手,只会使进度更加落后”,增加的沟通工作量可能完全抵消任务分解所产生的效率^[6]。Curtis 通过对某个开发项目 19 次项目会议的持续观察,发现了符合“Brooks 法则”的现象^[24]。该项目开发的后期,有两位分析员加入到被研究的项目中来,但对团队的绩效没有起到明显的效果,也没有提出新的设计方法。Curtis 认为,在项目之初团队处于学习阶段,沟通和协调较活跃。如果一个新成员在项目的学习期结束后才加入团队,他带入的新知识对团队产生的作用是十分有限的。如果这些新的成员必须加入团队,项目经理应当更多地加强团队内部的沟通,确保新成员带入的知识能够与团队现有的想法综合在一起。

Curtis 还发现,在开发团队中各成员对沟通活动的参与程度是不同的。在对 17 个大型项目的研究中发现项目早期的沟通是由少数个体占据的^[15],而对某个开发项目 19 次会议的持续观察也发现 10 个团队成员中有 3 人(一个客户代表,两个设计人员)似乎支配着整个项目,成为沟通的中枢^[24]。Curtis 对这些项目的实际领导者的背景进行分析发现,这些人员同其他人员相比拥有更广的知识和经验领域,而不是在某个领域经历的时间更长。Sonntag 对德国和瑞士的 29 个软件开发项目的研究也发现类似的现象,积极沟通、绩效优秀的项目成员比平庸者从事过更多项目,掌握更多种开发语言,而不是有更长的工作年限^[17]。这表明,知识和经验的广度比资历更能预测成员在团队沟通中实际扮演的角色和他的绩效。

Clegg 通过访谈研究发现,最终用户和项目经理这两个角色在软件开发沟通中的作用未得到发挥^[25]。大部分受访者强调,最终用户是系统开发的关键性角色,尤其在需求分析和应用领域知识的沟通方面。但在传统的开发过程中,最终用户只是参与需求获取和确认测试阶段,结构化方法和工具限制了用户专家发挥其角色的作用。在大部分访谈中,开发人员仍然被认为是软件开发的主体力量,即使对管理和新系统运行负责的客户经理也经常被认为缺少对项目的影响力。大部分受访者还强调,高层经理较缺乏对新技术及其长远影响的理解,成为技术沟通的瓶颈;项目经理缺乏对系统中人员和组织方面的重视,导致内部沟通不畅。

Steven 和 Henry 根据 Belbin 的团队角色模型,对软件开发团队中“培养者”角色进行了实验研究。Belbin 曾经提出,有效决策团队的核心由八个重要角色组成,其中“培养者”(Plant)角色是指团队中能够洞察行为过程、提出进一步建议和新思维的角色,其特点在于个人主义、思考严肃、有见识、非正统、智慧。Steven 和 Henry 的结果表明,全部由“培养者”组成的团队在完成软件开发任务的时间上明显少于对照组。同时,通过观察发现,这样的团队有较强的沟通能力,他们不但自己能够产生创新的思想,而且能够很快理解队友的新想法^[26]。

3.3 沟通的频次分布

软件开发中的沟通不仅限于团队内部，而是涉及整个组织内部和外部。Curtis 通过现场研究，给出一个软件组织沟通的分层结构^[15]。如图 3 所示，共有五个层面：(1) 个人层面 (2) 团队层面 (3) 项目层面 (4) 公司层面 (5) 商业环境层面。

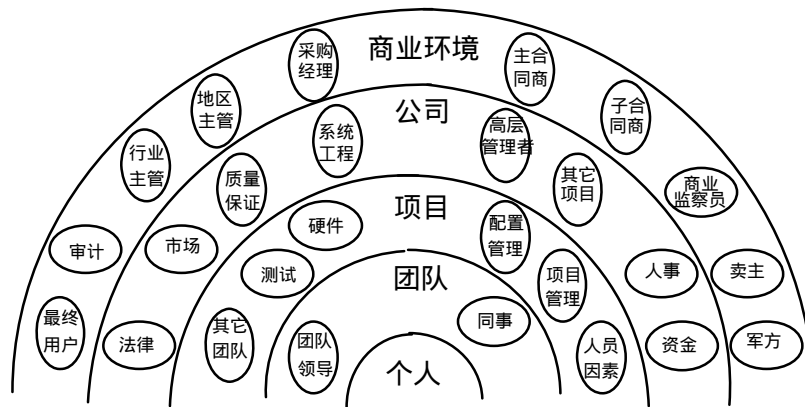


图 3 软件组织沟通的分层结构

该模型指出，软件开发的正式沟通最常发生在团队成员之间；同一项目的不同团队之间沟通相对较少，有一定合作关系的群体之间沟通更少，群体与外界的沟通最少。跨越不同层面之间的沟通渠道经常会过滤掉很多信息或对所传递的信息的意义产生曲解。除了正式的沟通结构引起的障碍，沟通的困难还在于地理上的分隔，文化差异和环境因素等。

Dutoit 和 Bruegge 通过现场观察发现，成效良好的软件开发组织，其团队之间的沟通频次高于团队内部的沟通频次。另一方面，从时间坐标上来看，开发成效良好的团队在需求和联机调试两个阶段的沟通频次明显高于开发效果不佳的团队^[27]。Gopal, Mukhopadhyay 和 Krishnan 对离岸软件开发的沟通问题进行了研究。离岸开发 (offshore development) 是指不在美国本土进行的软件开发。印度目前是美国最大的离岸软件开发承接国。研究者对 34 个在印度开发的软件项目进行问卷调查，结果表明成功的项目经理与软件提供商的沟通频次和与最终用户的沟通频次保持平衡^[28]。

3.4 沟通的方式

在软件开发项目中，沟通的方式因沟通者处于软件组织的不同层面，开发项目处于不同阶段以及开发采用不同的方法而各不相同。Curtis 在软件组织沟通分层结构中 (图 3) 指出，不同层面的内部和群体之间的具有不同的沟通方式^[15]。(1) 个人层面，正式沟通主要靠文档，并存在多个跨越组织边界的个人网络。(2) 团队层面，文档沟通效果不佳，经常靠建立一种正规程序来促进沟通。(3) 项目层面，最有效的沟通手段是正式评审，一个或多个项目成员横跨几个团队也有助于项目团队的沟通。(4) 公司层面，非正式的人员接触比公司的评审程序更有利于沟通。(5) 商业环境层面，最重要的是与各个客户洽谈，开发出一致认可的需求。

Kraul 和 Streeler 对 65 个软件项目群体内部的沟通实践进行调查, 从使用频率和价值评价两个维度考察常用的沟通技术^[29]。调查结果表明, 由于需求评审、设计评审和客户测试等沟通方式允许担负不同职责的个人(需求分析员、系统工程师、构架师、程序员、测试员和客户)通过结构化的方式, 对相互的工作发表各种不同的意见, 因此其价值评价明显高于其使用频率所暗示的价值, 其中同行讨论是最常用且评价最好的沟通技术。根据统计数据, 被调查者对正式沟通的评价显著低于其使用频率所暗示的价值。这里所说的正式沟通既包括规范化的当面人际交往进行信息传递(代码检测、状态审查)也包括非当面的沟通(项目文档、源代码检查、里程碑时间表、错误跟踪等)。计算机辅助开发管理工具(CASE)则是使用频率最少且评价最低的沟通技术。

4 协调问题

4.1 整体协调

Toffolon 和 Dakhli 指出软件开发活动要协调四个方面的依赖关系: 股东/客户(S)、使用者(U)、软件构架师(A)和开发人员(D), 并提出全局软件模型(global software model), 如图 4 所示^[21]。研究者认为软件开发活动由四个空间组成的: 问题空间、方案空间、建造空间和操作空间。问题空间(problem space)主要是定义股东/客户需要解决的问题及其组织解决方案, 对股东/客户、使用者和构架师进行协调; 方案空间(solution space)主要是针对问题的计算机解决方案, 对软件构架师、股东/客户和开发人员进行协调; 建造空间(construction space)主要是解决方案的实现, 对开发人员、构架师和使用者进行协调; 操作空间(operation space)主要是从最终用户的角度评价软件的可用性, 从而也验证产品竞争力, 对使用者、开发人员和股东/客户进行协调。

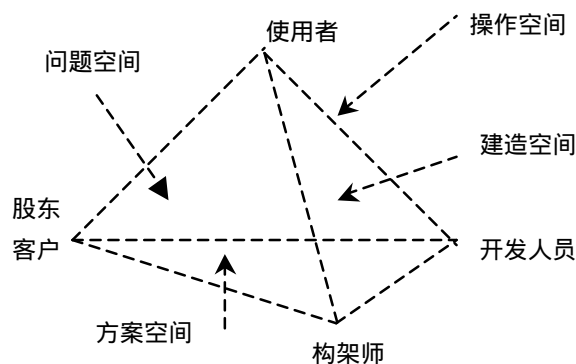


图 4 全局软件开发模型

4.2 外部协调

Toffolon 和 Dakhli 还指出, 软件开发的依赖关系可分为两类: 外部依赖(又称为纵向依赖)和内部依赖(又称为横向依赖)。对应这些依赖关系的协调也可分为两类, 即外部协调和内部协调^[21]。

软件开发的需求分析阶段是外部协调出现频次较高的阶段。Curtis 等人在对大型开发项目的现场研究中发现,确定需求的过程中,需要对外部的市场影响和公司自身的实力进行协调,同时还要考虑一些潜在的影响^[15],如图 5 所示。

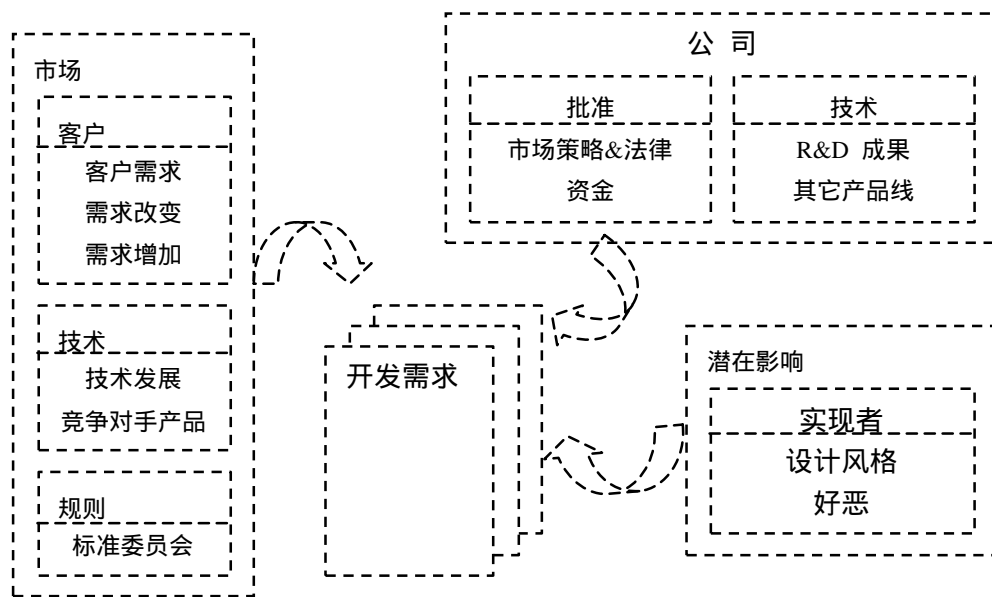


图 5 需求过程的协调

Crowston 和 Kammerer 在对需求过程中协调与信息搜集活动的研究中指出,需求分析包含四种相互依赖关系:生产者-客户依赖、任务-子任务依赖、任务-成员依赖和需求之间的依赖。针对生产者-客户依赖的协调,研究者建议采用原型产品标准化、询问客户输入信息和使用者卷入开发等协调方式。针对任务-子任务依赖的协调,研究者指出,需求分析实质上是一项任务计划活动,分析人员必须了解哪些任务已完成,并选择那些已经完成或已存在的低端任务来实现高端任务。针对任务-成员依赖的协调,研究者指出,需求分析阶段的协调应当更侧重于明确执行特定任务所需资源的类型,从而协调任务与人员的关系。针对需求之间依赖的协调,研究者指出应当对这种依赖进行分类,采取措施针对可能对开发活动产生不良影响的依赖进行抑制^[30]。

4.3 内部协调

研究者还对软件开发团队内部结构的协调职责进行了分析。传统的开发团队是由一个高级程序员指导若干初级程序员的工作,有经验的高级程序员作成项目领导者,指导若干成员或团队,而初级程序员对其编程软件负责,报酬是通过提升体现出来的。在这样一种层峰结构的开发队伍中,协调的工作量非常大,成本非常高。针对传统开发团队中程序员对其编程产品高度卷入和以提升作为承认程序员工作价值的唯一方式的状况,Weinberg 提出“无我编程”(Egoless programming)的概念。他根据 Festinger 的认知失调理论,认为程序员由于

对自己的编程产品高度卷入,会认为编出的程序是自己个人的产品,从而害怕别人发现其中的缺陷和错误,害怕暴露自己对任务的不胜任,不愿接受对其程序的改进建议。这不利于软件开发中的协调活动。因此,Weinberg 建议成立“无我开发团队”(Egoless team),强调个人的成功和失败都被看作是团队共同努力的结果,鼓励合作而不是竞争。如果一个团队成员执行任务遇到困难,提倡其他成员贡献出自己的力量^[4]。Baker 提出主程序员团队的模式。这种模式类似于外科手术的团队,主程序员是主要的设计者,编写重要的代码段落;后援程序员实施重要的操作,并为必要时接替主程序员做好准备;程序员在设计规范可实施的阶段加入团队。主程序员团队协调团队成员间的合作,提供了一个与医疗行业中围绕主治医师进行协调的相同的工作模式^[31]。

Levesque, Wilson 和 Wholey 等人对软件团队是否会形成共同心智模式(shared mental model)的问题进行了实验研究。结果表明,并不是所有开发团队都会随着时间的推移形成共同心智模式。复杂项目的不同分工将导致成员和群体在知识上的差异。这种差异有时是功能上的,因此通过召开会议建立团队的共同心智模式成本很高。以共同心智模式作为协调手段所获得的效益将随项目规模的增加而不断降低。研究者同时指出,以共同心智模式作为协调手段最适合于团队成员具有相近技巧和能力的情况;而团队中某些成员的技巧和能力高于其他成员的情况下,更适合通过项目经理的工作进行协调;对于任务简单明确的情况,更适合把其他成员的工作视为黑盒,采用严格的规范进行协调^[32]。

5 今后研究可拓展的方向

综上所述,西方学者对软件开发的沟通与协调问题进行了较为深入的研究。这些研究使我们对软件开发中沟通的组织结构、频次分布、沟通方式等问题,以及软件开发组织的整体协调和外部内部协调等问题有了一定的了解。今后研究可拓展的方向主要包括:

(1) 软件开发组织的沟通协调行为与软件生命周期关系的研究。软件从孕育到衰亡的生存过程称为生命周期(life cycle)。一般把软件生命周期分为:软件计划、需求分析、软件设计、软件编码、软件测试和运行维护等六个阶段。不同的开发框架又可形成不同的生命周期模型。以往涉及软件生命周期的沟通与协调行为研究多集中在前期的几个阶段。既然软件开发活动是随生命周期的增长而熵不断增加的过程,今后的研究应当拓展到软件生命周期的全程,尤其应加强对软件生命周期后期的沟通与协调问题研究。

(2) 软件开发组织与用户的沟通与协调研究。以往对用户这一角色的沟通协调研究仅局限于需求分析阶段,随着市场对软件产品可用性的重视,开发过程日益强调全程的用户参与(user involved),今后应当对软件组织与用户的沟通与协调行为在软件生命周期各个阶段的方式和作用进行更为广泛的研究。

(3) 虚拟开发环境下的沟通与协调行为研究。技术的进步使得以虚拟团队的形式进行软件开发成为可能。在虚拟环境下,软件开发组织中沟通与协调的动机、认知过程以及行为过程会发生哪些改变是一个需要深入探讨的问题。

(4) 西方沟通与协调问题研究成果对我国软件开发组织的适用性的研究。我国软件组

织规模较小,管理水平相对较低,开发过程缺乏层次较为混乱。我国对软件开发的人员与组织因素重视程度不足,目前仅有少量规范研究,尤其缺乏对软件开发组织沟通与协调问题的实证研究。我国学者除了需要关注上述的研究拓展方向,还需要考察西方已有的软件组织沟通与协调问题研究成果(如 SW-CMM 模型、P-CMM 模型中关于沟通与协调的规范)是否适合我国软件开发的组织环境。从而尽快形成适合我国软件开发组织的沟通与协调规范,促进软件开发生产率和产品可用性。

参考文献

- [1] Pressman R. Software Engineering: A Practitioner's approach. 5th ed.. 北京:清华大学出版社,2000
- [2] Ewusi-Mensah K, Przasanyski Z. Factors contributing to the abandonment of information system development project, Journal Of Information Technology, 1994, 9: 185~201
- [3] Doherty N F, King M. The consideration of organizational issues during the system development process: an empirical analysis, Behavior & Information technology, 1998, 17(1): 41~51
- [4] Weinberg, G M. Psychology of Computer programming. New York: Van Nostrand Reinhold, 1971
- [5] Brooks F P. No silver bullet: Essence and accidents of software engineering, the proceedings of the IFIP tenth world computing conference,1986. 1069~1076
- [6] Brooks F P. Mythical Man-Month: The Essays on Software Engineering, Anniversary Edition, Addison Wesley Professional, 1995
- [7] Boehm B W. Improving Software Productivity, IEEE Computer, 1987, 20(9): 43~57
- [8] Boehm B W. Software Engineering Economics, Prentice-Hall Inc., Englewood Cliff, NJ,1981
- [9] Constandine. The peopleware paper, Yourdon Press, New Jersey, 1995
- [10] DeMarco T, Lister T. Peopleware: Productive Project And Team, Second Edition, Dorset House, 1999
- [11] Pitty K, Joyner L. Peopleware Revisited., <http://www.acs.org.au/nsw/articles/1999073.htm>, Jul.2000
- [12] Software Engineering Institute, Carnegie Mellon University, Capability Maturity Model , <http://www.sei.cmu.edu>
- [13] Software Engineering Institute, Carnegie Mellon University, People Capability Maturity Model ,version2.0, <http://www.sei.cmu.edu>
- [14] Carnegie Mellon University, Software Engineering Institute, People CMM-Based Assessment Method Description, <http://www.sei.cmu.edu>
- [15] Curtis B, Kransner H, Iscoe N. A Field Study Of The Software Design Process For Large Systems. Communication of the ACM, 1988, 31: 1268~1287
- [16] McConnel S. Classic Mistake, IEEE Software, 13(5), <http://www.stevemconnel.com/ieeesoftware/bp05.html> ,1996
- [17] Sonnentag S. Excellent Software Professional: Experience, Work Activities, And Perceptions By Peers. Behaviors And Information Technology, 1995, 15: 213~225
- [18] Sonnentag S. Excellent Performance: The Role Of Communication And Cooperation Process, Applied Psychology: An International Review, 2000, 49(3): 483~497
- [19] Marks M A, Mathieu J E, Zaccaro S J. A Conceptual Framework And Taxonomy Of Team Processes. Academy of Management Review, 2001, 26: 356~376
- [20] Malone T W, Crowston K. What Is Coordination Theory And How Can It Help Design Cooperative Work System, CSCW90 Proceedings, 1990. 357~369
- [21] Toffolon C, Dakhli S. A Framework For Studying The Coordination Process In Software Engineering, SAC'00 March,19~21, Italy(c)2000, ACM. 851~857
- [22] Schneiderman B. Software psychology : human factors in computer and information systems, Winthrop Publishers,Cambridge, Mass,

- 1981
- [23] Castka P, Bamber C J, Bamber D J. Teamwork As A Strategy Leading Towards The Involvement Of Users In Technology Design Process- A Manufacturing Perspective. Proceeding Of International Summer Academy On Technology Studies,2001
- [24] Walz D B, Ealm J J, Curtis B. Inside A Software Design Team: Knowledge Acquisition, Sharing, And Integration. Communication of the ACM, 1993, 36: 63~77
- [25] Clegg C, Axtell C, Damodran L. Information Technology: A Study Of Performance And The Role Of Human And Organization Factors, Ergonomics, 1997, 40(9): 851~871
- [26] Stevens K, Henry S. Analyzing Software Team Using Belbin's Innovative Plant Role ,<http://www.radford.edu/~kstevens2/ISTall.pdf>, 2002
- [27] Dutoit A H, Bruegge B. Communication Metrics For Software Development, IEEE Transactions on software engineering,1998, 24(8): 615~628
- [28] Gopal A, Mukhopadhyay T, Krishnan M S. The role of software process and communication in offshore software development, Communication of the ACM, 2002, 45(4): 193~200
- [29] Kraul R E, Streeler. Coordination in software development. Communication of the ACM, 1995,38(3): 69~81
- [30] Crowston K, Kammerer E E. Coordination And Collective Mind In Software Development, IBM System Journal, 1998, 37(2): 227~245
- [31] Baker F T. Chief Programmer Team Management Of Production Programming, IBM sys .J. 1972, 11(1)
- [32] Levesque L L, Wilson J M, Wholey D R. Cognitive Divergence And Shared Mental Model In Software Development Project Team, Journal of Organization Behavior,2001, 22: 135~144

Communication and Coordination Issues in Software Developing Organization

Yang Kun, Wang Erping

(*Institute of Psychology, Chinese Academy of Sciences, Beijing 100101*)

Abstract : Based on the review of human and organization issues in software development organization, the present article first analyzed the importance of communication and coordination issues in software development organization, then introduced both communication issues such as communication and team structure, communication and team member, distributing frequency of communication and communication methods, and coordination issues such as global coordination, external coordination and internal coordination in software development organization. Finally indicated some possible direction on further study of communication and coordination issues in software development organization.

Key words: communication, coordination, software development, organization psychology.