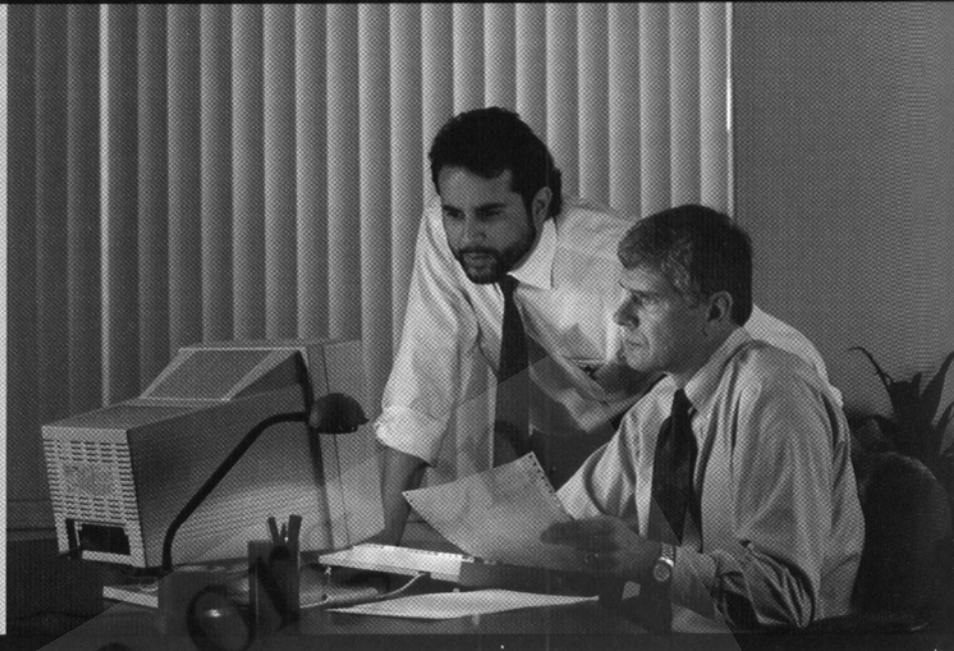


摘要: 软件失败的诸多事实反复提醒人们一个朴素而深刻的道理: 仅仅依靠技术是难于提供较完整的开发方案, 是难于驾驭复杂的软件工程。祭奠了长期实践经验, 领悟软件管理的内在规律, 从指导软件开发的实践出发, 借鉴了CMM的过程管理技术, 论文探讨软件开发的融合方法。该方法不仅开拓了理论的研究领域, 而且具有重要的实际指导意义。文章阐述了项目质量的两个重要支撑点, 陈述了融合框架的主要内容。

关键词: 软件工程 风险 CMM 融合



软件工程与软件能力成熟度的融合

杨一平 马慧 (首都经济贸易大学信息学院 100026)

1 项目质量的两个支撑点——管理与技术

我们必须承认: 太多的软件项目开发失败了。以往虽然大多数项目也有计划, 并具有项目里程碑, 项目也提供了软件工程要求一系列文档(例如逻辑模型、物理模型等), 然而, 由于项目缺乏有效管理与控制方法, 不少项目尽管花费了大量人力、物力和财力, 但开发到最后, 却被描述成既超出预算, 又质量很差。项目管理仍然很困难。孙子兵法: “知己知彼, 百战不殆”。对于软件项目管理者而言, 这个“彼”指的就是“风险”。在IT行业中, 时常会发生这样的情况: 两个资深专家, 对某个项目的可行性分析大相径庭, 甚至相反的意见, 而双方的意见都有相当充分的论据做依托。这就是IT行业的特殊性与风险, 对于大多数人应首先了解风险。那么, 主要风险表现在如下几个方面。

1.1 项目管理常见的问题

典型的问题主要包括: 产品发货迟、项目规模与成本不断增加、产品不能满足用户的要求、软件分包商的水平太差、薄弱、混乱、漏洞百出

甚至项目无法控制等等, 其中常见的管理软件问题归纳起来可分成资源(成本)问题、进度问题、质量问题三大类, 见表1所示。

事实上, 以上典型问题的描述只是表面现象, 内在管理上的混沌与不成熟等实质性问题往往被隐藏起来, 造成较大的风险, 造成严重损失, 见图1所示。

1.2 管理产生漏洞的主要因素

1.2.1 重视技术, 轻视管理基础工作

在相当长的一段时间里, 人们把软件项目的开发仅仅看作是计算机技术和网络技术的应用, 开发人员把软件产品看作是一种艺术产品, 视开发项目为“交钥匙工程”。事实上, 信息系统开发过程本质上是一个社会过程, 在信息系统的开发过程中, 用户、开发人员、监理等参与者相互联系、相互影响。所有参与人的相互沟通、通力合作乃是项目开发成功的基础条件。换句话说, 项目管理依赖于技术支持与管理的支持。如果其中一方存在薄弱环节, 项目将面临很大的风险, 而有些项目经理过分强调迅速发展活跃于技术领

表 1

问题分类	典型问题描述
资源(成本)问题	成本 / 预算问题
	计算机资源的使用问题
	人事 / 员工的分配问题

进度问题	开发计划问题
	测试进度问题
	递增、修改进度问题
	里程碑出现的问题
	生产率问题
管理质量问题
	配置管理薄弱
	错误的捕捉需求
	错误的结构设计
	错误的编码管理
	错误的测试方案设计
	错误的转换方式
	培训管理薄弱
错误的递增、修改管理	
外包后薄弱的管理	
缺乏基线的管理	
.....	

域的新技术,忽略了对项目环境、项目团队以及管理过程的重视。

1.2.2 管理工作随意性强,缺乏规范性

项目的成败过分依赖于项目经理与用户领导的能力。管理工作随意性强,缺乏规范性,一旦人员稍有变动,项目将面临较大的压力。

1.2.3 信息管理不规范

项目组织间缺乏交流,信息交流不畅通,信息缺乏统一的管理,信息的纪录、修改滞后。

1.3 融合的必要性

1.3.1 软件开发方法的创新与突破刻不容缓

软件开发中出现的一系列成本、进度、质量问题会严重影响软件产品的交付使用,项目开发中暴露出来的问题不断揭示着管理上深层次的混乱。软件产品质量与技术和管理均息息相关,软件产品的质量包括技术含量与管理的含量,因此,开发技术应考虑技术和管理两个方面,所以研制与实践一种新的开发方法已迫在眉睫。

1.3.2 形成黑箱的管理存在较大风险

管理的随机性强,可视性差,它就好像看不见,摸不着,风险大的黑箱。软件产品是不可标识的,是不受控的,变更也是不受控的。融合技术强调过程管理,是在一定程度上能缓解上述矛盾。

1.3.3 项目管理的可视性有待提高

项目管理的可视性有待提高,项目管理具有共性,有着其内在的规律,通过CMM的过程管理使软件开发的可视化管理、规范化管理、自动化管理提供了可能。

2 过程管理与软件工程技术的融合

2.1 融合框架的主要内容

软件产品质量与技术和管理息息相关,软件产品的质量包括技术含量与管理的含量。软件开发方法应全面考虑技术和管理两个领域的问

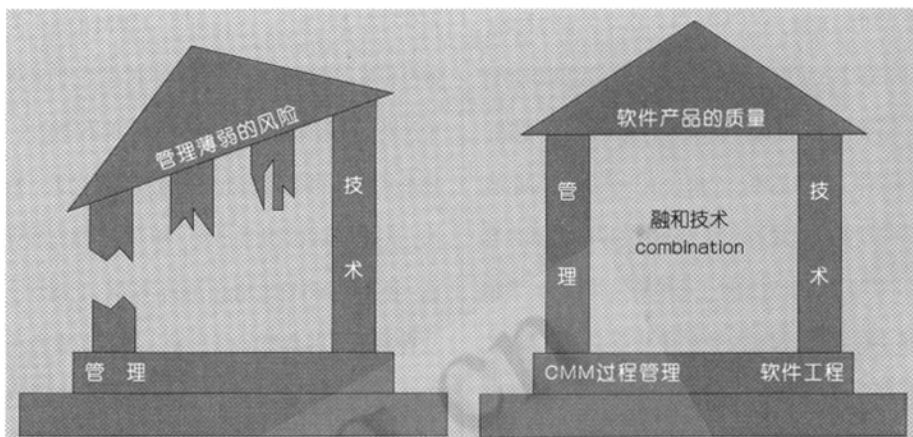


图1 管理薄弱导致风险

图2 技术与管理支撑软件的质量

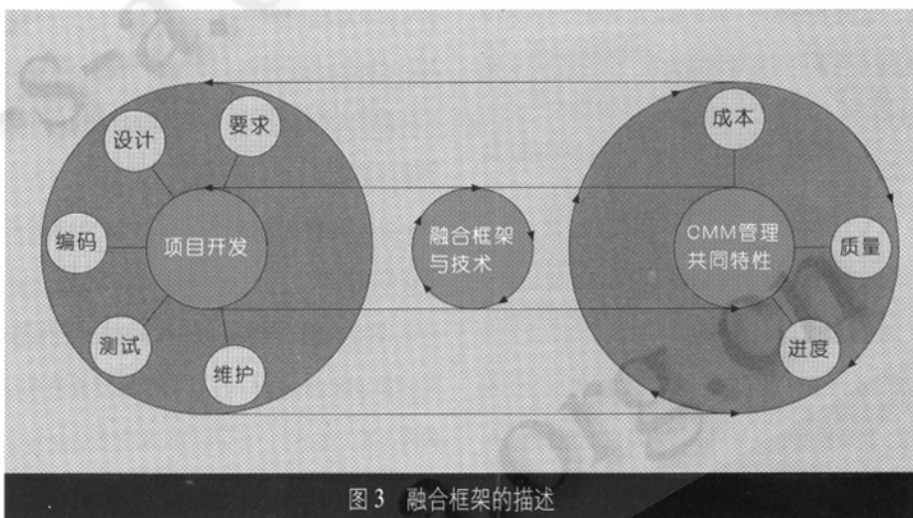


图3 融合框架的描述

题。技术和管理保证性活动的有机融合构成了软件过程的两个重要支撑点,见图2所示。在软件过程中插入CMM的共同特性,即软件项目管理与软件开发生命周期之间建立链接,构成了软件产品质量的重要支撑。

CMM和软件工程的融合框架能在一定程度上抑制了风险是有其内在的原因。我们讨论不是简单地将软件工程与CMM形式与表面上的融合,在软件开发和CMM的关键域之间存在着内在本质性的联系,这个桥梁与结合点就是软件过程的共同特性与关键实践,通过对共同特性的测量、评价构成了CMM的关键过程域(KPA)与开发结构的双向通信和融合,它指导人们明确指责和目的,提供必要的基础设施和执行重要的活

动。如图3所示。

如同车和路之间的密切联系一样,软件开发生命周期与CMM管理之间存在着内在的联系。CMM提供关键过程域的结构,这个结构包括共同特性、目标、约定、能力、活动、测量与验证。CMM模型它能帮助人们解决诸如什么、何时、为何、谁的等问题,它是质量管理的重要支撑。

软件工程与CMM的过程管理构成了质量管理的两个重要支撑点,这两个支撑点又是由不同的阶段、方法以及分布在CMM模型的多层元素构成。为了提高融合技术的可操作性,我们从实际出发,把软件工程的几个阶段和CMM的关键过程域进行分析、归纳、重新分类,组合构成了一个二维表即融合框架,见表2所示。

我们知道,在CMM中有18个关键过程域,分布于2~5级中。他们在CMM的实践中均起到了重要作用,但每一个KPA的侧面是不同的。在软件工程中的每一个阶段的工作特点不同,容易出现的管理漏洞不同,应重点加强的管理内容不同,因此,针对软件工程的阶段管理风险我们建立与关键过程域的联系,在框架中描述了在软件开发的每一个阶段应重点借鉴的CMM的重点过程关键域。

2.2 关于框架的几点说明

上述表格描述了软件生命周期每个开发阶段应重点强调的过程管理的内容,下面是几点补充说明:

2.2.1 软件开发方法的选择

软件产品和其他产品一样,均经历着一个孕育、成长、成熟、衰退的生存过程。在前几章中已经就软件的诸多开发方法以及它们的各自特点做了较详细的介绍。开发方可以根据项目的规模、复杂程度以及用户的需求特点,选择生命周期法与原型法相结合,以及面向对象的开发方法等,在这里不在赘述。当开发方法选择后,可参考融合框架对其内容进行适当的剪裁和增补。

2.2.2 对融合框架进行适当的调整

CMM提供了一个阶梯式的进化框架,在每一级又包括了若干个关键过程域,这些关键域分

别分布在2~5级上。与此同时,CMM的每个关键过程域不是面对某一个特定的过程,它们对整个开发过程起作用。但在开发过程的不同阶段,技术要求则不同,容易出现的管理问题也是不相同。从指导实践的角度出发,为了便于操作和实施,我们有针对性地构造了融合框架,它揭示了从容易出现的典型问题出发,罗列了重点于CMM的过程关键域的联系。应该指出,模型的构造并不妨碍CMM每一个过程关键域对整个开发过程的指导意义,开发方可根据项目的实际情况对框架适当的进行调整。

2.2.3 框架应用的具体步骤

按照上述框架的内容,考虑到尽量简单并便于阅读和理解,我们总结了以下的内容:

- (1) 简单地描述软件工程的各个阶段的主要工作与特点。
- (2) 阐述某个具体阶段容易出现的典型问题。
- (3) 陈述CMM的某个过程管理域的核心内容。
- (4) CMM某个过程关键域的启示与指导意义。鉴于CMM的具体内容较多,在这里不再赘述。不难看出,即使融合技术像一切新生事物一样还存在尚待改进、细化的过程,但理论与实践文化底蕴,在规避风险过程管理的实践道路上必将发挥其进步的意义。 ■

表 2

关键过程域	主要参考的关键过程域	一般参考的关键过程域
开发阶段		
项目准备与计划	软件项目计划 软件配置管理 软件转包合同管理 培训程序	软件产品工程 缺陷预防
需求分析	需求管理	组织过程焦点 组织过程焦点
系统设计	软件项目跟踪和监督	同级评审
系统实施	软件质量保证	组间协调
系统评价与维护	软件配置管理、基线管理	过程改革管理 技术改革管理

