

新一代的过程改进模型:CMMI

A Process Improvement Mode of Next Generation——CMMI

谢 超 马士龙 林梦香

(北京航空航天大学计算机科学与工程系 北京100083)

Abstract CMMI is a new process improvement model that is being developed by SEI. It is the evolution of process improvement and it may be representative of next generation of process improvement model. This paper provides a brief overview of the history of CMM family, the need for integration of those models, the status of the CMMI effort, and the different descriptions of CMMI.

Keywords CMMI, CMM, process improvement, Process area, Capability level, Maturity level

CMMI (Capability Maturity Model Integration) 是 SEI 正在开发的一种过程改进模型, 该模型的出现是过程改进领域的一次重要变化, 会对今后软件组织的过程改进工作产生巨大影响。本文简要介绍了 CMMI 的产生背景、当前状况、主要内容, 以及和其他 CMM 模型之间的对比。

一、CMMI 产生背景

自从70年代软件危机以来, 学术界和企业界对软件工具和技术的研究投入了大量的人力和物力, 也取得了许多成果。然而, 直到80年代末, 一个不同于以往的概念逐渐被接受: 一个软件组织的软件能力取决于该组织的过程能力。如果一个组织过程能力越成熟, 那么该组织的软件生产能力就越有保证。

什么是过程? 过程指的是为达到一个预定目的所实施的一系列步骤(IEEE-SDT-610)。

SEI (Software Engineering Institute, 卡基梅隆大学软件工程学院) 提出了持续的软件开发过程和维护过程改进, 并在1991年发布了 SW-CMM 1.0 (“软件能力成熟模型1.0”), 主要用于指导软件组织的过程改进工作。该模型的发布, 在计算机领域掀起了软件组织的过程改进高潮。

自 SW-CMM 发布以来, 该模型的原理和作用已得到公认, 成为一种事实上的国际标准。国际标准化组织以 CMM 模型思想为基础, 组织制定了相应的标准——ISO 15504, 随着 SW-CMM 的推广, SEI 也致力于开发一系列相应的能力成熟模型, 包括: SE-CMM (系统工程能力成熟模型)、IPD-CMM (产品集成开发能力成熟模型)、P-CMM (个人能力成熟模型)、SA-CMM (软件获取能力成熟模型) 等, 适应不同的用途。多种能力成熟模型的开发和应用受到了广泛的欢迎和支持, 过程改进活动也涉及到了更多的规范。经过过程改进

的软件组织能够更好地开发和维护他们的产品与服务。

当一个软件组织采用多种能力成熟模型用于改进产品质量并提高生产能力时, 从理论上讲, 不同的能力成熟模型必须能够和谐地共存。但实际上, 通过实践发现, 不同的能力成熟模型, 特别是 SW-CMM、SE-CMM 和 IPD-CMM 之间在内容上有所重叠, 并在结构和指导规范上存在差异, 以至于培训、评估、改进活动经常被重复执行, 对过程改进工作造成不利的影响。可是, 如何消除冗余活动, 没有任何指导可以借鉴。显然, 软件组织急需一种简单易行的方法, 能够将基于各种 CMM 的改进活动集成起来。即, 将各种不同的模型进行集成。

二、CMMI 现状

2.1 CMMI 源模型、目标和产品组成

为了适应软件过程改进的发展需要, 美国国防部 OUSD/AT&L (the Office of the Under Secretary of Defense, Acquisition, Technology, and Logistics) 和 NDIA (National Defense Industrial Association) 的系统工程委员会联合发起了“能力成熟模型集成”项目, 由 SEI 负责实施。来自政府、工业界、SEI 等不同组织、拥有不同背景的100多名专家一同致力于建立一个能够适应当前和未来过程改进的模型框架——CMMI (能力成熟模型集成)。CMMI 项目组决定以 SW-CMM2.0 草案 C、EIA/IS 731 和 SE-CMM、IPD-CMM v0.98a 三种模型为基础进行 CMMI 的开发工作^[3]。

由于 CMMI 的三种源模型应用于不同的规范, 其结构也不尽相同, 重点也不一致, 导致集成过程改进模型成为一件很困难的工作。CMMI 项目组将目标定为: 集成不同的源模型, 消除相互之间的不协调, 减少重复; 降低基于模型的过程改进工作的费用; 采用通用

的术语、一致的风格、统一的结构和规则、通用的组件,以使模型的结构更清晰,内容更容易理解和掌握;所有的CMMI产品与国际标准ISO 15504保持兼容;使已经使用CMMI源模型进行过程改进工作的软件组织能够保留他们的工作成果^[2,4]。

CMMI产品套件由三个部分组成:CMMI规范模型、评估产品、培训产品^[2,5]。(1)CMMI规范模型是CMMI产品中最为重要的部分,包含多种规范模型,用于指导过程改进工作。CMMI模型以“过程域”为基本的结构单位,每个过程域描述了在本过程范围内应该进行的实践以及原因。软件组织可以根据自身需要,选择CMMI产品中的一个或多个规范,或不同的描述版本,以适应组织的实际情况。目前已涉及的规范有:软件工程、系统工程、IPPD等。(2)评估产品主要包括评估需求、评估方法等,用于CMMI的评估。(3)培训产品主要包括CMMI介绍、评估方法培训等,用于CMMI的培训。

2.2 CMMI的重要意义

CMMI的出现,不仅仅是将多种模型结合起来,更重要的是,CMMI是能力成熟模型发展史上的一次重要变革,CMMI提供了更广泛的过程改进范围^[4,7]。CMMI之前的能力成熟模型,都是偏向于对某一单独规范(如:软件工程或系统工程)过程的改进,而CMMI模型并不考虑规范本身的局限性,将重点放于产品或服务的开发和维护过程的改进之上,因此,有可能会对过程改进领域中的所有相关规范产生巨大影响。

使用CMMI模型进行过程改进工作的优越性在于:软件组织可以更有效地使用多种过程规范进行评估和改进工作;降低了过程改进工作的培训和评估费用;可对软件组织中的所有改进活动从整体上进行控制;为软件组织提供了一种新的描述特定规范信息的方法。

2.3 CMMI模型的几个重要阶段

CMMI项目于1997年正式启动;SEI于1999年11月发布了CMMI的意见征求版V0.2,该版本的发布目的在于征求各方面的意见和建议,以便对该CMMI产品进行修改;2000年8月,SEI发布了CMMI V1.0。该版本基于对V0.2的各种反馈意见,对CMMI产品进行了改进,可用于软件组织的过程改进工作;2000年11月,SEI发布了CMMI V1.01;2000年底,SEI将发布CMMI V1.1。(已发布的CMMI产品请参见SEI站点<http://www.sei.cmu.edu>)

三、CMMI模型描述

CMMI模型分成两种描述方式,以适用于不同的需要。按照两种不同的描述模型进行过程改进,会得到

不同的级别——能力级别或成熟级别。

3.1 连续描述和分级描述

CMMI采用两种描述方式:分级描述和连续描述。比如,SW-CMM是一种分级描述模型,而EIA/IS 731是一种连续描述模型。软件组织可以根据自身的过程改进目标,选择不同的CMMI描述模型。

在CMMI的分级描述模型中,基本组件可归纳为成熟级别。每个成熟级别中包括多个过程域,每个过程域包含有目标(可分为特定目标和一般目标)、公共特性和实践。实践描述了应该执行的活动,可以根据公共特性将实践分成五类,(1)实施承诺:确保过程的建立和执行;典型的实践有:组织方针等。(2)实施能力:为了完整地实施过程活动,创造所需要的条件;典型的实践有:计划、资源、组织结构和培训等。(3)实施活动:实施过程所需的活动;不同的过程域所实施的活动不一样,还可细分为典型工作产品和子实践。(4)实施指导:监督和控制过程;典型的实践有:将特定的工作产品置于配置管理之下,根据计划监督和控制过程,在必要时采取纠错行动等。(5)实施验证:确保活动与过程域所要求的相一致;典型的实践有:评审和审计等^[4]。

在CMMI的连续描述模型中,基本组件可归纳为过程域。过程域根据用途不同分成四类:过程管理过程、项目管理过程、工程过程、支持过程^[5]。每个过程域中包括特定目标和一般目标,其中,特定目标需要通过实施特定的实践活动予以实现,而一般目标只需通过实施一般活动就可予以实现。特定目标和特定实践在每个过程域中是唯一的;而一般目标和一般活动则可应用于多个过程域。每个实践隶属于某个能力级别。

3.2 能力级别与成熟级别的比较

成熟级别和能力级别都用于标识过程改进效果,二者的主要区别在于:采用的描述方式不同,应用的对象不同^[3]。其中,成熟级别属于分级描述模型,应用于组织内部全面的过程改进工作,分为1到5个级别,每个成熟级别包括一系列的目标,使用分级描述,软件组织可以通过成熟级别逐步建立一种组织优越性文化(Culture of Engineering Excellence),清楚地展现组织的过程成熟能力。同时,每个成熟级别形成了向下一级别前进的基础。

能力级别属于连续描述模型,应用于某个特定过程域的过程改进工作,分6个能力级别,从0级到5级。使用连续描述,软件组织可以在过程改进活动中建立并维护一个能力级别视图(例如:一个已达到能力级别的过程域列表),使软件组织清楚地意识到已取得的进步,并可用于指导下一步的改进活动。能力级别视图可有如下类型:成绩视图——在选定的过程域中,现已取得的能力级别;目标视图——希望取得的能力级别。

表1 CMMI 两种描述的比较

分级描述	连续描述
(1)软件组织可以按照一条预定的、并已被证明有效的改进之路进行过程改进。	(1)可以根据软件组织的商业目的,自由地选择最重要的方面进行过程改进。
(2)是建立在一个相对而言是经过较长时间考验的模型基础之上。	(2)将过程域独立出来,可增强改进工作效果的能见度。
(3)研究和数据表明采用此种描述的改进活动会对软件组织带来重大效益。	(3)对某个过程域进行改进时,软件组织可以将工作重点放于该过程域中最困难的部分。
(4)可以方便地将过程改进工作从 SW-CMM 转到 CMMI。	(4)可以方便地将过程改进工作从 EIA/IS 731 转到 CMMI ^[2] 。
(5)采用同种描述模型进行过程改进的软件组织之间可以相互比较,以判断改进效果。	(5)过程域的组织方式与 ISO 15504 基本相同,采用 CMMI 的过程改进活动可以同采用 ISO 15504 标准的过程改进活动进行直接比较,以判断改进效果。
(6)是一种持续性的改进活动,从最基本的管理实践开始,逐级前进。	(6)可以将高能力级别的一般实践应用于所有的过程域,以普遍提高组织各过程的能力。
(7)可使用成熟级别等级对过程改进效果进行总结。	

表2 成熟级别与能力级别

级别	分级描述,成熟级别	连续描述,能力描述
0级	无	未进行过程改进
1级	进行了过程改进	进行了过程改进
2级	可管理级	可管理级
3级	可定义级	可定义级
4级	量化管理级	量化管理级
5级	不断优化级	不断优化级

3.3 模型的选择

事实上,两种模型包含的基本信息是一致的。每种模型都描述了各种过程的主要方面,之所以采用两种不同的描述方式,关键在于为了满足不同软件组织的不同需求。

如果一个软件组织希望改进该组织各个方面的过程能力,考虑到各过程域之间的相关性,应该选择 CMMI 的分级描述模型。使用分级描述模型,该组织可以将改进工作重点放在某一成熟级别的所有过程域上。当达到了这些过程域所规定的目标,就可认为改进工作取得了成功,该组织相应地达到了某个成熟级别。例如:如果该组织成功地达到了 CMMI 分级描述模型的第2级的所有目标,就意味着:该组织处于成熟级别2级。

如果该组织认为大多数的过程处于一种可接受程度,并希望提高某个或多个特定过程域的能力,可以选择 CMMI 的连续描述模型。使用连续描述模型,该组织只需将改进工作的重点放于与特定目标过程相关的过程域上。当该组织达到了该过程域的特定目标,同时达到了某个能力级别的一般目标,就可以认为该组织在所改进的过程域上已获得相应的级别能力。例如:如果该组织成功地达到了 CMMI 连续描述模型的第2级和第3级的一般目标,同时达到了产品集成过程域第3级的特定目标,就意味着:该组织处于产品集成

能力级别3级。

因此,软件组织过程改进活动出发点的不同决定了所选择模型的不同。

四、CMMI 模型与其他模型/标准的关系

4.1 模型关系框架图

图1显示了 CMMI 与其他 CMM 以及计算机领域中相关的其他标准、规范之间的相互关系。可以看出,CMMI 与 ISO 15504 都是从 SW-CMM 发展出来的,这种先天因素决定了二者之间是相互兼容的。

CMMI 是一个处于开发中的产品,CMMI 产品组决定尽量将 CMMI 与新的国际标准(包括:ISO 15504、ISO 12207、ISO 15288等)兼容,为使 CMMI 尽早成为一个国际标准而努力,文[2]对这一问题进行了分析。

4.2 CMMI 与 CMM

关于 CMMI 较之以前 CMM 的优点,我们的看法与文[1][2][4][6]和[10]中的观点相近。

1)CMMI 是“集成的”,将互补的规范集成为一个整体,会对整个企业范围的过程改进带来更大的效益。CMMI 将系统工程和软件工程的过程改进紧密地结合在一起。对于一个希望进行过程改进以提高软件能力的软件组织,采用 CMMI,可以对系统工程和软件工程同时进行改进,甚至可以考虑关于产品集成方面的改进问题。可以通过一次性的过程改进,从多个方面提高组织的软件能力。如果采用以前的 CMMI 模型,是不可能得到这种效果的。

2)CMMI 过程改进方法学能够帮助组织消除内部过程改进环境的冗余,如果按照几种不同过程模型进行改进工作,就不得不围绕着这些模型分别建立过程,这会造成改进环境的冗余,并增加了过程改进费用。例如:系统工程组要建立系统工程过程改进环境,软件工程组要建立软件过程改进环境。但如果使用 C-

