一种支持软件过程控制和改进的主动度量模型*

王 青+, 李明树, 刘 霞

(中国科学院 软件研究所 互联网软件技术实验室,北京 100080)

An Active Measurement Model for Software Process Control and Improvement

WANG Qing+, LI Ming-Shu, LIU Xia

(Laboratory for Internet Software Technologies, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62544128, E-mail: wq@itechs.iscas.ac.cn, http://www.iscas.ac.cn

Received 2003-06-03; Accepted 2003-11-26

Wang Q, Li MS, Liu X. An active measurement model for software process control and improvement. *Journal of Software*, 2005,16(3):407−418. DOI: 10.1360/jos160407

Abstract: This paper presents an active measurement model (AMM) for software process control and improvement, and the related measurement method. AMM formally describes the goal, feature, and indicator of software process, and their relationship. AMM also provides some principles, methods, and techniques to determine the measurement of software process. Based on AMM, on one hand, software organizations can deduce the appropriate measurement process according to the goals of the processes they focus; on the other hand, they can also identify the opportunity and roadmap of improvement according to the result of measurement. It will provide an effective support for software organization to make an accurate decision and achieve a successful result.

Key words: software process measurement; measurement model; quality feature; measurement process; decision-making support

摘 要: 提出了一种支持软件过程控制与改进的主动度量模型 AMM(active measurement model)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.基于该度量模型,软件组织一方面可以依据确定的过程目标,主动导出合适的度量过程;另一方面还可以依据度量的结果,识别过程改进的机会,并主动导出过程改进的方向.为正确的决策和成功的结果提供有效的支持.

关键词: 软件过程度量;度量模型;质量特征;度量过程;决策支持

中图法分类号: TP311 **文献标识码**: A

软件过程度量是软件过程质量控制的主要手段,过程度量可以使管理者能够洞察产品开发过程,掌握项目

* Supported by the National Natural Science Foundation of China under Grant Nos.60273026, 60473060 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant Nos.2001AA113080, 2002AA116060 (国家高技术研究发展计划(863))

作者简介: **王青**(1964−),女,重庆人,博士,研究员,主要研究领域为软件过程技术与质量管理;**李明树**(1966−),男,博士,研究员,博士生导师,主要研究领域为软件过程技术与需求工程;**刘霞**(1978−),女,博士生,主要研究领域为软件过程技术与质量管理.

© 中国科学院软件研究所 http://www.jos.org.cn

的进度、开销、产品质量状态等,使整个项目的开发过程处于受控状态,为管理者制定决策提供可量化的依据.度量还可以帮助我们确定整个组织的开发效率,识别组织潜在的过程改进区域,为组织实施过程改进提供客观的信息支持.

软件过程本身具有不确定性和知识密集型等特征,使得软件过程的度量一直成为影响软件工业化发展的主要障碍.传统的过程度量方法,例如 GQM[1],GDSM[2]等,都是将组织的商业目标逐级分解出过程、活动的目标,再按照计划→计划实施→结果分析的步骤提出度量的要求.这些方法得到了广泛的认同,但他们只是给出了建立度量的基本方法和步骤以及一些基本的原则,缺少建立度量的形式化支持和度量过程的算法支持,无法实现度量过程重复性和自我优化改进.近两年,人们的注意力主要集中在基于 SPC(statistical process control)[3−5]和 6 Sigma[6]的软件过程度量和控制技术方面.Jim Lawler 等人提出的度量模型技术[7],为解决度量的标准和度量数据的有效性方面做出了有益的贡献,该模型技术中提出的度量协议,使得度量可重复,并保证度量结果只取决于度量因子,而不会受到其他不确定因素的影响.

本文提出一种支持软件过程控制和改进的主动度量模型(active measurement model,简称 AMM),支持在软件组织关注的过程目标驱动和引导下,自动生成软件项目的度量过程,并保持度量的可重复性以及度量结果的公正性和可比性.此外,AMM 模型还支持度量模型本身的自优化和改进,为软件组织建立有效度量和合适度量提供了方法和技术上的支持.

1 软件过程度量的基本元素

度量的目的是通过对度量对象的刻画、评价和规划,实现对象的改进,如图 1 所示,度量是一个典型的 PDCA 循环,其中对度量对象的刻画是度量工作的第 1 步.软件过程是一个可度量的对象实体,与其他对象实体一样,软件过程有很多可度量的特征,这些特征反映了软件过程这样或那样的行为、状态直至目标.随着软件组织的经营目标、商业环境以及过程改进的要求的不同,组织所关注的过程目标也不相同,与此同时,组织要度量和监控的过程特征也不相同.

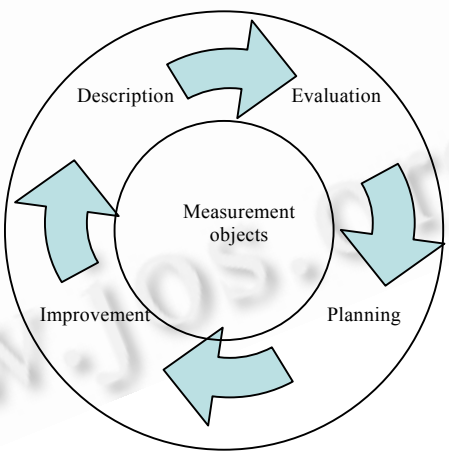

Fig.1 The reason and goal of measurement

图 1 度量原因和目标

遵循 CMMI[8,9]的分类方法,将软件过程分为 4 类,即:软件工程过程、项目管理过程、过程管理过程和支持过程.组织中每个过程都承担着其某个环节的生产目标,不同的过程除了有着一些共性的目标以外,还有一些其本身特有的目标.与之相对应,每个软件过程,除了具有一些公共的质量特征以外,还有其固有的、特殊的质量特征.不同类的软件过程、甚至同类中不同的软件过程,由于其过程目标不同,其过程的质量特征也不尽相同.

图 2 给出了软件过程的特征描述,其中目标代表在一般质量管理体系中要求的过程目标,每个过程的目标都由一组特征来刻画,从图中可以看出,软件过程的质量特征可以分为公共特征和特殊特征.

© 中国科学院软件研究所 http://www.jos.org.cn

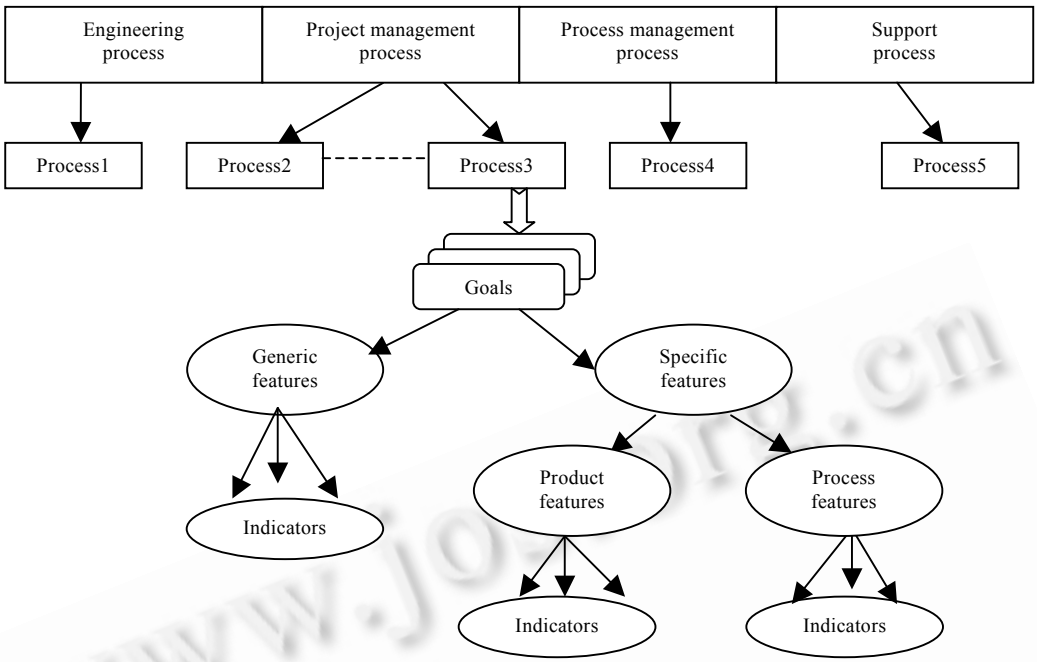

Fig.2 The features and its related elements of software process

图 2 软件过程的特征及其相关元素

1.1 软件过程度量元素的形式化定义

如图 2 所示,软件过程的特征包括两个部分:

(1) 公共特征:指所有过程都具有的特征,例如过程成本的稳定性、性能稳定性、过程计划的符合性等.

(2) 特殊特征:指过程具有的个性化的特征,由于每个过程都会生产该过程的产品,所以过程产品的质量特征也是考察过程能力的重要因素.

(a) 过程产品特征:过程产品是过程的输出,是衡量过程能力的主要因素.由于过程在组织生产环节中承担的责任和角色不同,过程的产品也就不同,所以不同的过程,过程产品的质量特征也不相同.比如过程产品的缺陷率,看起来似乎是一个公共特征,但是,不同的产品其缺陷率的度量方法也是不同的,因此,这里将过程产品的缺陷率归为特殊特征的范围.

(b) 过程特征:不同的过程本身还有其固有的特征,比如测试过程的用例覆盖率、需求开发过程的需求变更率等等.

从图 2 可以看出,过程特征是过程度量的核心内容,一方面特征刻画了过程的目标,使得我们在实施过程度量时可以有的放矢,使我们理解度量什么、为什么度量;而另一方面,要使得特征可度量,还需要对特征进行数学抽象,使我们知道如何度量.

针对图 2 中的度量元素,我们对特征、相关元素以及元素间的关系进行如下形式化的定义.

定义 1(取元素元组操作). 定义 GetElement 为取元组中的元素操作,对∀元组 $X=\langle C_1,C_2,\ldots,C_n\rangle$, X.GetElement(C_i)=C_i.Value,缩写为 $X.C_i=C_i$.Value.

首先,我们关注的第 1 个问题是过程目标.前面已经谈到,每个过程都有一组目标,这组目标实现的程度反映了过程的能力.过程目标的定义如下:

定义 2(过程目标 *Pg*).$Pg=\langle Gid,p,G\_cls,T\rangle$,其中 Gid 是目标的标识,p 是软件过程,$p\in P$,P 是软件过程的集合,T 是目标 Pg 对应的特征集合.

在定义 2 中,$P=\{p_1,p_2,\ldots,p_n\}$,G_cls 表示特征类型,$G\_cls\in G\_clsType$,$G\_clsType=\{SG,GG\}$,其中 SG 表示过程

© 中国科学院软件研究所 http://www.jos.org.cn

的特殊目标(specific goal),*GG* 表示过程的公共目标(general goal).

过程目标反映过程能力的状态,每个公共目标可以反映所有过程的状态,但在具体的度量分析中,我们可能只关注几个具体的过程,而且目标的度量是依赖于具体过程的,同样的公共目标,由于度量相关联的过程不同,其度量的结果也不同.所以目标和过程总是成对出现,一个目标的度量结果只表征其具体关联的过程的能力状态.

每个特殊目标则反映其关联的唯一过程的状态.

定义 3(目标-过程映射函数). *Pg* 表示过程的目标,$\forall Pg$,$\exists$函数θ,$\theta(Pg)=P_{\text{Goal}}$,$P_{\text{Goal}}\subseteq P$,我们称函数$\theta$为目标 *Pg* 的过程映射函数.

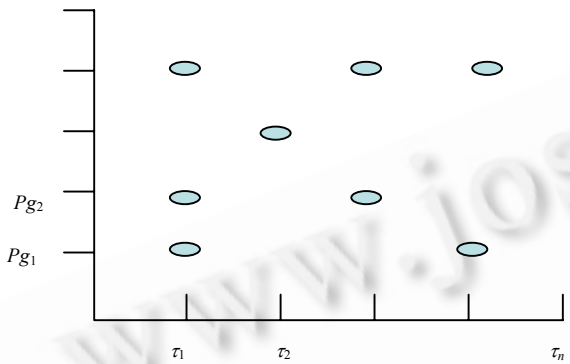

Fig.3 Mapping of process goals and features

图 3 过程目标与特征的映射关系

推论 1. 由定义 2、定义 3 可知,$\forall Pg$,if *Pg.G_cls=GG*,则$\theta(Pg)=P$,$\forall Pg$,if *Pg.G_cls=SG*,则$\theta(Pg)=\{p\}$,*p=Pg.p*.

定义 4(过程目标集合). 过程 *p* 的所有目标构成过程的目标集合 P_{goal},即,$P_{\text{goal}}=\{Pg_1,Pg_2,\ldots,Pg_n|p\in\theta(Pg_i),1\le i\le n\}$.

前面已经讨论,过程的目标由一组特征来刻画,反之,一个特征也可以刻画多个目标.比如需求的符合性,既可以反映需求分析过程的特殊目标——"有效的需求管理"达到的程度,也可以反映设计过程的特殊目标——"有效的需求实现"达到的程度.所以目标和特征是一个 *M-N* 的映射关系.如图 3 所示,过程特征的定义如下:

定义 5(过程特征τ). $\tau=\langle Fid,cls,I\tau\rangle$.其中,*Fid* 是特征标识,*cls* 是特征的类型,$I\tau$是τ对应的度量指标集合.

在定义 5 中,*cls* 表示特征类型,$cls\in clsType$,$clsType=\{Speci\text{-}prod,Speci\text{-}proc,Gener\}$,*Speci-prod*,*Speci-proc*,*Gener* 分别表示过程的特征类型为过程产品特征、过程特征和公共特征.

从前面的讨论可知,要实现过程的可度量性,还需要对过程特征进行数学抽象,将过程特征抽象为一组可度量的指标.该度量指标封装了特征相关的度量数据和数据的采集方法或采集活动.度量指标的定义如下:

定义 6(度量指标 *I*). 过程特征及其对应的评测数据、数据获取技术所构成的元组 $I=\langle\tau,D_I(\tau),A_I(\tau)\rangle$称为过程的度量指标,其中,$D_I$ 表示特征τ针对指标 *I* 的评测数据,A_I 表示特征τ针对指标 *I* 的评测数据获取技术或活动.

可以用 *Is* 表示过程特征度量指标的集合,$Is=\{I_1,I_2,\ldots,I_n\}$.

可以看到,指标是特征的量化表示,一个特征需要一个或者多个指标来表示,例如,对于过程的稳定性,需要通过过程成本、进度等与计划的符合,过程的变更情况等多个指标的计算来分析.同样,一个指标也可以对应于多个特征,例如常用的代码的复杂度,既可以反映产品的可靠性,也可以反映出产品的可维护性和可移植性.

指标抽象地封装了特征及其相关属性,最后我们定义施加于指标之上的度量算法,以形成度量程序,度量程序的定义如下:

定义 7(度量程序 Metric). 设 Metric 表示一个度量程序,$Metric=\langle I,f(I)\rangle$,$f(I)$表示指标 *I* 的度量算法.

*f*表示此度量程序的度量算法,针对具体的指标 *I*,通常用$f(I)$表示,$f\in F$,*F* 表示度量算法的集合.对于任何一个度量指标,都可能存在多种度量方法,以从不同的角度去度量和分析该指标所反映的趋势.

1.2 软件过程度量元素之间的关系

评估软件过程成熟能力的主要方法是考察其对应的过程目标是否实现,软件过程不同,它们在软件生产过程中要达到的目标也各不相同,前面我们定义了过程目标、特征和度量指标等元素,这些元素之间的关系如图 4 所示.

© 中国科学院软件研究所 http://www.jos.org.cn

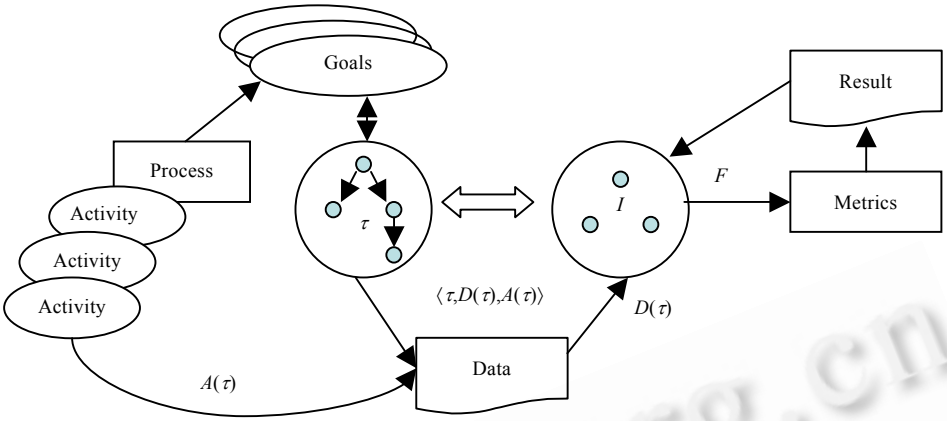

Fig.4 Relationship of elements for software process measurement

图 4 软件过程度量元素之间的关系

从图 4 可以看出,软件过程的每一项特征都可以反映过程的部分和全部目标,每一个特征的度量结果都可以对应到相应度量目标的分析和决策支持.过程目标和特征、特征与度量指标之间都是 *M-N* 的映射关系.

前面的定义给出从目标→特征→度量指标的单向映射关系.

如由定义 2 可知,过程目标 $Pg=\langle Gid,p,G\_cls,T\rangle$,则可以得到过程目标所定义的特征集合 $T_{Pg}=Pg.T$.

下面继续讨论这些元素之间逆向映射关系,以支持基于本度量模型的过程改进方法的研究.根据以上的定义,可以首先得到特征与目标的映射算法,见算法 1.

算法 1. 特征到过程目标的映射算法:*MapT*to*G*.

输入:任意特征 τ.

输出:过程、目标对集合 $PnG\tau=\{\langle p_1,Pg_i\rangle,\langle p_2,\text{Goal}_{p_2}\rangle\ldots\}$

步骤:

(1) 初始化:$PnG\tau=\varnothing$,P=过程全集

(2) Loop 1:

Break If 过程集合 $P=\varnothing$

取过程 $p\in P$,$P=P-p$

根据定义 4 得到过程 p 的目标集合 $P_{\text{goal}}=\{Pg_1,Pg_2,\ldots,Pg_n\}$

(3) Loop 2:

Break If $P_{\text{goal}}=\varnothing$

取 $Pg_i\in P_{\text{goal}}$,$P_{\text{goal}}=P_{\text{goal}}-Pg_i$

根据定义 2 得到 $Pg_i=\langle Gid,p,cls,T\rangle$

If $\tau\in Pg_i.T$ Then

$PnG\tau=PnG\tau+\langle p,Pg_i\rangle$

(4) End Loop 2

(5) End Loop 1

算法结束.

从算法 1,我们可以得到特征 τ 对应的过程、目标对集合:$PnG\tau=MapT\text{to}G(\tau)$.

从图 2 和图 4 可以知道,过程特征和度量指标之间也是 *M-N* 的映射关系.从定义 5 过程特征的定义可知:$\tau=\langle Fid,cls,I\tau\rangle$,所以从定义 5 可以得到过程特征所定义的度量指标集合 $I\tau=\tau.I\tau$.

反之,我们可以根据以上的定义,得到指标与特征的映射算法.见算法 2.

© 中国科学院软件研究所 http://www.jos.org.cn

算法 2. 指标到过程特征的映射算法:*MapI*to*T*.

输入:任意指标 *I*.

输出:关联的特征集合 T_I.

步骤:

(1) 初始化:$T_I=\varnothing$,*T*=过程特征全集

(2) Loop 1:

Break If 特征集合 $T=\varnothing$

取过程 $\tau\in T$,$T=T-\tau$

根据定义 5 得到:$\tau=\langle Fid,cls,I\tau\rangle$

If $I\in\tau.I$ Then

$T_I=T_I+\tau$

(3) End Loop 1

算法结束.

从算法 2,我们可以得到指标 *I* 对应的特征 τ 的集合:$T_I=MapI\text{to}T(I)$.

2 主动软件过程度量模型

在第 1 节我们讨论了软件过程度量的基本元素,基于这些基本元素,我们可以定义软件过程的度量模型,定义如下:

定义 8. 软件过程主动度量模型 AMM(active measurement model for software process).

设 AMM_Model 表示软件过程度量模型,$AMM\_Model=\langle T\text{set},I\text{set},\text{Metric-set}\rangle$,其中:$T\text{set}=\{\tau_1,\tau_2,\ldots,\tau_n\}$表示所有软件过程特征的全集.$I\text{set}=\{I_1,I_2,\ldots,I_m\}$表示所有软件过程指标的全集.$\text{Metric-set}=\{\text{Metric}_1,\text{Metric}_2,\ldots,\text{Metric}_L\}$.

定义 8 定义了 AMM 度量模型.由该模型,我们可以以组织当前关注的过程目标为输入,自动导出满足需求的度量过程.如图 5 所示.

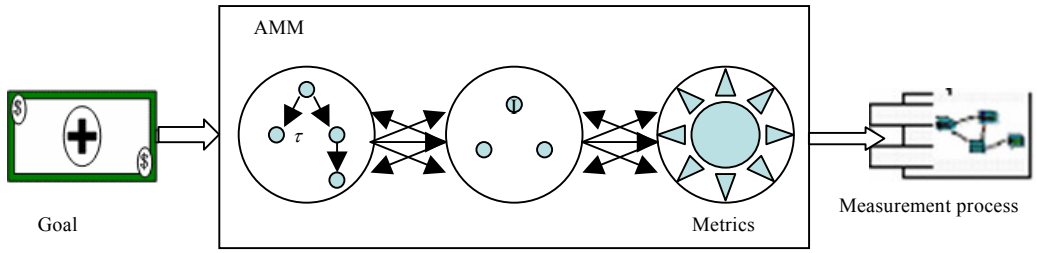

Fig.5 Generating the measurement processes based on AMM

图 5 基于 AMM 模型的度量过程生成

主动度量模型 AMM 具有以下特征:

(1) 基于组织在当前时间关注的过程目标,使得度量的目标明确,度量工作有的放矢.可以很好地解决为什么而度量的问题.

(2) 以过程目标为输入,基于 AMM 度量模型,可以主动导出适宜的度量过程,解决度量什么以及如何度量的问题.

(3) 度量结果具有 3 方面的作用:

- 对度量的对象过程进行控制
- 为组织过程的改进提供依据
- 对度量模型本身进行自我优化和改进

(4) 基于度量结果,可以主动导出过程改进的目标和方向.

3 基于 AMM 的软件过程度量方法

我们多次讨论过,度量是有成本的.度量的目的是为了提高过程管理的效能,以量化的数据分析和评价来度量对象的某些目标是否实现.度量是过程管理的一种手段,而不是管理的目标.

软件过程可度量的方面很多.然而,过度的度量会造成成本非必须性增长,过少的度量又容易使管理者看不到度量的意义,并且由于数据支持不足,容易导致决策失误.所以组织需要根据其能力、项目的具体情况来选择合适的度量.

第 2 节提出的一种主动的度量模型正是遵循了这样的基本原则,以解决度量的有效性和适宜性问题.图 6 是基于 AMM 模型的软件过程度量方法.该方法旨在针对组织确定关注的过程目标,基于 AMM 度量模型,自动导出软件过程的度量过程,并支持度量过程和其他软件过程的融合,以实现度量数据的自动采集、度量的可重复性和度量过程的自动执行.

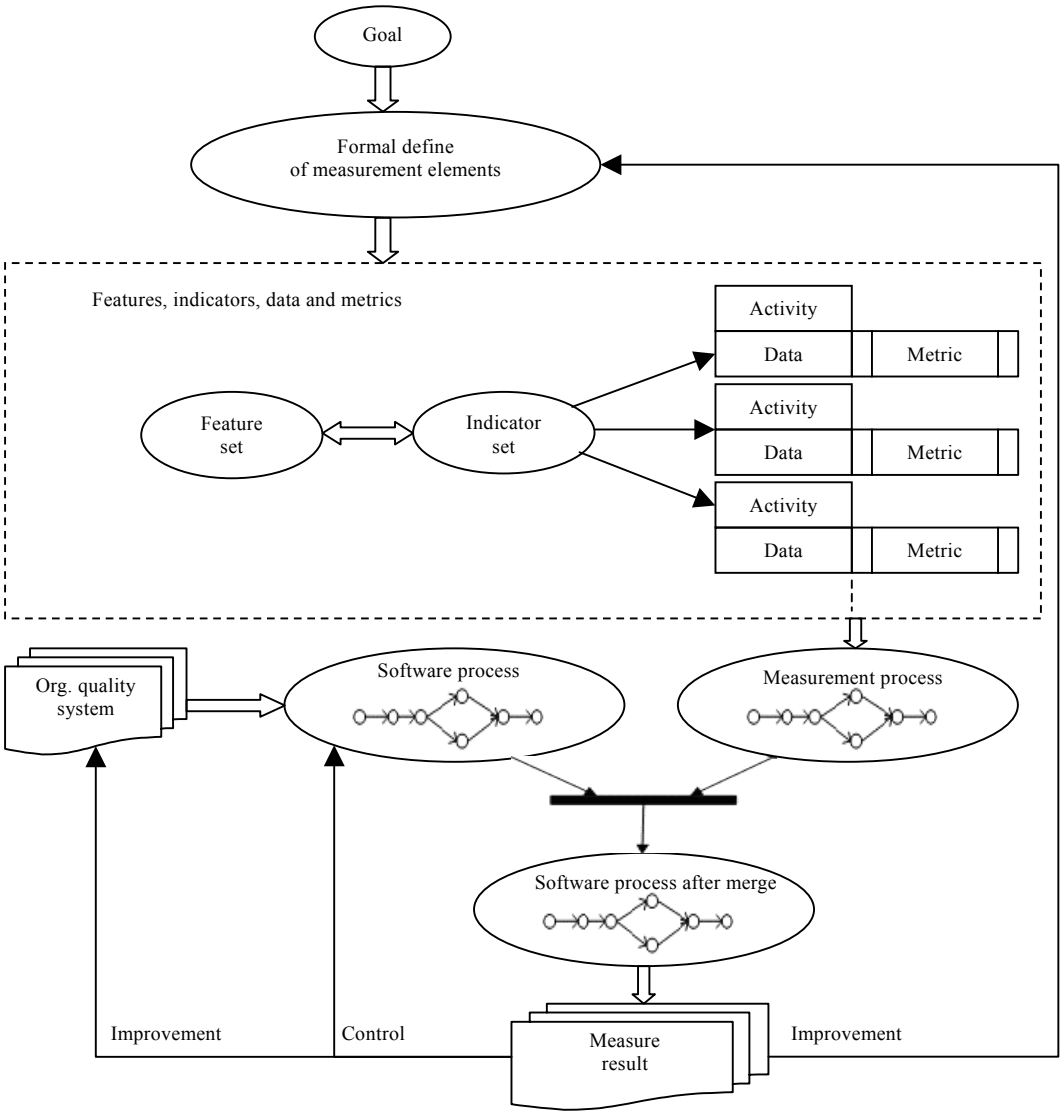

Fig.6 Measurement methods for software process based on AMM

图 6 基于 AMM 的软件过程度量方法

由图 6 我们可以看到,AMM 是一个开放的模型,既指导组织当前阶段度量过程的生成,还可以在度量结果的支持下不断地自我改进和优化.度量过程也可以基于组织关注的过程目标,在 AMM 模型的支持下,主动、动态地增加或减少.

基于 AMM 度量模型,我们可以主动地导出过程的度量过程.度量过程的导出算法见算法 3.

算法 3. 基于 AMM 的度量过程自动生成算法.

输入:要度量的过程的目标集合 $InPut=\{\langle p,MPg\rangle|p\in P,MPg\subseteq Pgoa\}$.

输出:度量过程 $OutPut=MeasrementProcess=\{Mp_1,Mp_2,\ldots,Mp_m,\ldots\}$

$$Mp_i=\langle p,\{\langle Pg,\{\langle \tau,\{\langle D(\tau),A(\tau),f(I)\rangle,\ldots\}\rangle,\ldots\}\rangle,\ldots\}\rangle$$

步骤:

(1) 初始化:MeasurementProcess=∅

(2) Loop 1:

Break If 要度量过程为空;

从 InPut 中选取要度量的过程及其目标$\langle p,MPg\rangle$;

$InPut=InPut-\langle p,MPg\rangle$

(3) Loop 2:

Break If 过程 p 的目标集合 MPg 为空;

选取过程的 p 的目标 $Pg,Pg\in MPg$;

根据定义 2,$Pg=\langle Gid,p,G\_cls,T\rangle$

Pg 对应的特征集合 $T_{Pg},T_{Pg}=Pg.T$

(4) Loop 3:

Break If T_{pg} 为空

取特征 $\tau\in T_{pg}$

根据定义 5,$\tau=\langle Fid,\tau\ cls,I\tau\rangle$

τ对应的度量指标集合 $I\tau$,

$I\tau=\tau.I\tau=\{I_1,I_2,\ldots,I_m\}|$

根据定义 6,$I_i=\langle \tau,D_{Ii}(\tau),A_{Ii}(\tau)\rangle$

(5) Loop 4:

Break If $I\tau$为空;

取指标 $I_i,I_i\in I\tau$,

根据定义 7,可以得到 I_i 对应的度量程序

$Mgram_{Ii}=\langle I_i,f(I_i)\rangle$

构造元组$\langle D_{Ii}(\tau),A_{Ii}(\tau),f(I_i)\rangle$

(6) End Loop 4

构造元组$\langle \tau,\{\langle D_{I1}(\tau),A_{I1}(\tau),f(I_1)\rangle,\langle D_{I2}(\tau),A_{I2}(\tau),f(I_2)\rangle,\ldots\}\rangle$

继续 Loop 3(取集合 T_{pg} 中的下一个特征τ)

(7) end Loop 3

构造元组

$\langle Pg,\{\langle \tau_1,\{\langle D_{Ii1}(\tau_1),A_{Ii1}(\tau_1),f(I_{11})\rangle,\langle D_{I12}(\tau_1),A_{I12}(\tau_1),f(I_{12})\rangle,\ldots\}\rangle,$

$\langle \tau_2,\{\langle D_{I21}(\tau_2),A_{I21}(\tau_2),f(I_{21})\rangle,\langle D_{I22}(\tau_2),A_{I22}(\tau_2),f(I_{22})\rangle,\ldots\}\rangle,\ldots\}\rangle$

继续 Loop 2(取集合 MPg 中的下一个目标 Pg)

(8) end Loop 2

构造元组 $Mp=\langle p,\{Pg_1,\{\langle \tau_1,\{\langle D_{Ii1}(\tau_1),A_{Ii1}(\tau_1),f(I_{11})\rangle,\langle D_{I12}(\tau_1),A_{I12}(\tau_1),f(I_{12})\rangle,\ldots\}\rangle,$

$\langle \tau_2,\{\langle D_{I21}(\tau_2),A_{I21}(\tau_2),f(I_{21})\rangle,\langle D_{I22}(\tau_2),A_{I22}(\tau_2),f(I_{22})\rangle,\ldots\}\rangle,\ldots\}\rangle,$

$\langle Pg_2,\{\langle \tau_1,\{\langle D_{Ii1}(\tau_1),A_{Ii1}(\tau_1),f(I_{11})\rangle,\langle D_{I12}(\tau_1),A_{I12}(\tau_1),f(I_{12})\rangle,\ldots\}\rangle,$

$\langle \tau_2,\{\langle D_{I21}(\tau_2),A_{I21}(\tau_2),f(I_{21})\rangle,\langle D_{I22}(\tau_2),A_{I22}(\tau_2),f(I_{22})\rangle,\ldots\}\rangle,\ldots\}\rangle,\ldots\}\rangle$

注:为简单起见,τ_1 在每个元组分别表示对于目标 Pg 的第 1 个特征,其余 D,A,I 的下标编号类似.

MeasurementProcess=MeasurementProcess+Mp;

继续 Loop 1(取输入集合中的下一个过程)

(9) end Loop 1

算法结束.

随着软件组织过程的改进,对过程的度量要求也是不断变化的.利用算法 3,我们可以根据组织所关注的质量目标,基于 AMM 度量模型,主动地导出相应的度量过程.从算法 3 的描述可知,算法 3 的输出结果是一个实例化度量模型后得到的度量过程实体.

在这样的模型支持下,软件组织总是可以根据当时组织的过程环境和过程控制与改进的目标主动地建立合适的度量过程,该过程不仅关注度量本身的目标,还与被度量的过程对象之间,自动建立关联关系.度量程序中的每一个度量活动都包含了〈对象过程,度量数据,数据的采集方法〉等信息,这样,度量程序建立后,就已经产生相对的测试或度量数据采集计划,该计划可以和软件项目的开发计划、质量保证计划相对应,为度量程序的执行和管理提供有力的支持.

4 基于主动度量模型的过程改进决策支持算法

AMM 模型的主动性一方面体现在可以根据组织关注的过程目标,主动导出有效、适宜的度量过程;另一方面,AMM 的主动性还体现在根据度量分析的结果,还可以主动导出过程改进的目标和方向.

基于第 2 节中 AMM 的定义和算法 3,算法 4 给出基于 AMM、依据度量结果对过程改进主动进行决策支持的算法.过程改进的前提是过程的目标没有很好地实现,或者提出更高的过程目标.依据 AMM 的度量结果,我们可以通过其中所指出的过程缺陷、异常状态或期望值,逆向地主动导出对应的过程目标,并进一步指导相关的过程实践的改进.

算法 4. 基于 AMM 的度量结果对过程改进的决策支持.

输入:有缺陷、异常、或更高期望的度量结果,Result;

输出:受影响的过程目标集合:

$$AffGoal=\{PnG\tau_1,PnG\tau_2,\ldots,PnG\tau_n\}.$$

步骤:

(1) 初始化:$AffGoal=\varnothing$;

$AMM\_Model=\langle T\text{set},I\text{set},\text{Mgram-set}\rangle$,由 AMM 导出的度量过程是 AMM 模型的一个实例化.所以从度量结果 Result 中,我们可以提取有问题、有改进机会和期望改进的结果所对应的度量指标集合:

$$SPI\_I\text{set}=\{I_1,I_2,\ldots,I_n\}$$

(2) Loop 1:

Break If $SPI\_I\text{set}=\varnothing$

任取指标 $I,I\in I\text{set}$,$SPI\_I\text{set}=SPI\_I\text{set}-I$

根据算法 2,可以得到对应的特征集合:$T_I=MapI\text{to}T(I)$

(3) Loop 2:

Break If T_I 为空

取 $\tau\in T_I$,$T_I=T_I-\tau$

根据第 4.1.1 节定义:$\tau=\langle Tid,cls,I\rangle$

IF $\tau.cls=Gener$ Then 所有的过程都需改进.

Else

根据算法 1,可以得到该特征对应的过程目标集合:$PnG\tau=MapT\text{to}G(\tau)$

End If

$$AffGoal=AffGoal\cup PnG\tau$$

(4) End Loop 2

重复 Loop 1(取 *I*set 中的下一个指标)

(5) End Loop 1

算法 4 提供了根据度量结果来分析、判断受影响的过程目标集合的方法.给出了应该改进的⟨过程,过程目标集合⟩.该算法可以指导组织进行有效的过程改进.

5 研究实例

AMM 模型的基本元素是过程的目标、特征和度量指标.以需求分析过程为例,基于 AMM 模型的定义,可以得到需求分析过程的度量元素及其关系,见表 1.

Table 1 The goals, features and indicators of requirement analysis process

表 1 需求分析过程的目标、特征与指标

Goal	Feature name/ID	Feature type	Indicator name/ID
Generic goal			
Institutionalize a managed process	Conformability to related quality system /ReqQsys_Conf	Gener	Ratio of conformability for SQA audit /R_SQAConf
	Conformability to plan /ReqPlan_conf	Gener	Ratio of plan conform Org. procedure /R_PlanOrgPro
	Identifiability of process product /ReqProd_Iden	Speci-Prod	Ratio of products identify/R_ProdIden Ratio of effective identification /R_IdenEffe
	Traceability of process product /ReqProd_Trac	Speci-Prod	Ratio of products identify/R_ProdIden Ratio of effective identification /R_IdenEffe
	Controllability of process /ReqProc_Cont	Gener	Ratio of cost be control/R_CostContr Ratio of schedule be control /R_ScheContr
Institutionalize a quantitatively managed process	Stability of process /ReqP_Stab	Gener	Ratio of cost conform its plan /R_CostConfPlan Ratio of schedule conform its plan /R_ScheConfPlan Ratio of process changing/R_ProcChan
Institutionalize an optmizing process	Effectivity of SPI /ReqPImp_Effe	Gener	Decrease trend of process cost /Run_CostDecrea Decrease trend of process duration time /Run _TimeDecrea
Specific goal			
Effective requirement management	Requirement stability /Req_Stab	Speci-Prod	Ratio of requirement changing /R_ReqChan Ratio of requirement non-consistent /R_ReqUncons
	Requirement conformability /Req_Conf	Speci-Prod	Ratio of requirement conform Org. standard /R_ReqConStd
	Effectivity of requirement management/ReqM_Effe	Speci-Proc	Ratio of cost return /R_CostReqCon

从表 1 可以看出,需求过程有 3 个通用目标,1 个特殊目标.对应于 4 个目标有一组特征和指标.

假定软件组织有以下 4 个软件过程 *P*={需求、设计、编码、测试}.选取过程“需求分析”为目标过程,即 *p*=*ReqPro*.以过程 *p* 的目标 *MPg*={制度化对需求过程的量化管理,有效的需求管理}为度量关注的目标,基于前面讨论的算法,可以导出相应的度量过程.步骤如下:

输入:度量过程的目标集合.

根据表 1 列示的需求过程的度量元素.作为输入的过程目标集合为:*Input*={⟨*p*,*MPg*⟩}=

{⟨*ReqPro*,制度化对需求过程的管理⟩,⟨*ReqPro*,有效的需求管理⟩}

根据算法 3:

(1) 选取第 1 个度量的过程及其目标:

〈*ReqPro*,制度化对需求过程的量化管理〉

根据定义 2,制度化对需求过程的量化管理=〈*Gid*,*p*,*G_cls*,*T*〉=〈G_1,*ReqPro*,*GG*,T_{G_1}〉

(2) 根据表 1,由定义 2 可得该目标对应的特征集合 T_{G_1}:

T_{G_1} =*Pg*.*T*={过程相关质量体系的适应性,过程计划的符合性,过程产品的可标识性,过程产品的可追踪性,过程的可控性}

(3) 根据定义 5,τ=〈*Fid*,*cls*,*I*〉

过程相关质量体系的适应性=〈*ReqQsys_Conf*,*Gener*,{*R_SQAConf*}〉

过程计划的符合性=〈*ReqPlan_conf*,*Gener*,{*R_PlanOrgPro*}〉

过程产品的可标识性=〈*ReqProd_Iden*,*Speci-prod*,{*R_ProdIden*,*R_IdenEffe*}〉

过程产品的可追踪性=〈*ReqProd_Trac*,*Speci-prod*,{*R_ProdIden*,*R_IdenEffe*}〉

过程的可控性=〈*ReqProc_Cont*,*Gener*,{*R_CostContr*,*R_ScheContr*}〉

根据定义 5 可以得到该对应的度量指标集合:

$$G\text{–}I\text{set}=\cup\ \tau.I=\{R\_SQAConf\}\cup\{R\_PlanOrgPro\}\cup\{R\_ProdIden,R\_IdenEffe\}\cup\{R\_ProdIden,R\_IdenEffe\}\cup\{R\_CostContr,R\_ScheContr\}.$$

(4) 根据定义 6 和定义 7,可以得到 I_i 对应的度量程序

I_i=〈τ,$D_{Ii}(\tau)$,$A_{Ii}(\tau)$,Metric$_{Ii}$=〈I_i,$f(I_i)$〉〉

构造元组〈$D_{Ii}(\tau)$,$A_{Ii}(\tau)$,$f(I_i)$〉

对于指标 *I*=*R_SQAConf*

构造元组 *M_R_SQAConf*=〈{*audit_defect*},{SQA *audit*},*ConfControlCharts*〉

对于指标 *I*=*R_PlanOrgPro*

构造元组 *M_R_PlanOrgPro*=〈{*plandata*,*realdata*},{*process audit*},*StabControlCharts*〉

同样,对于其他指标,分别构造元组:*M_R_ProdIden*,*M_R_IdenEffe*,*M_R_ProdIden*,*M_R_IdenEffe*,*M_R_CostContr*,*M_R_ScheContr*

(5) 于是可以得到过程目标〈*ReqPro*,制度化对需求过程的量化管理〉的度量元组 Mp_1

Mp_1=〈*ReqPro*,{〈制度化对需求过程的量化管理,{〈*ReqQsys_Conf*,{*M_R_SQAConf*}〉,〈*ReqPlan_conf*,{*M_R_PlanOrgPro*}〉,〈*ReqProd_Iden*,{*M_R_ProdIden*,*M_R_IdenEffe*}〉,〈*ReqProd_Trac*,{*M_R_ProdIden*,*M_R_IdenEffe*}〉,〈*ReqProc_Cont*,{*M_R_CostContr*,*M_R_ScheContr*}〉}〉}〉

(6) 对输入集合中的第 2 个〈过程、目标〉对:〈*ReqPro*,有效的需求管理〉

重复前面的(1)~(5),得到度量元组 Mp_2

Mp_2=〈*ReqPro*,{〈有效的需求管理,{〈*Req_Stab*,{*M_R_ReqChan*,*M_R_ReqUncons*}〉,〈*Req_Conf*,{*M_R_ReqConStd*}〉,〈*ReqM_Effe*,{*M_R_CostReqCon*}〉}〉}〉

输出结果:*MeasurementProcess*={Mp_1,Mp_2}就是相应的度量程序.

反之,我们根据度量结果,还可以主动导出应该改进的过程和对应的过程目标.假设指标“需求变更率/*R_ReqChan*”和“投入产出比率/*R_CostReqCon*”的度量结果表明有缺陷,可以根据算法 1,2 和 4 导出受影响的过程目标集合.步骤如下:

输入:*Result*=有缺陷的度量结果集,亦即 *SPI_I*set={需求变更率,投入产出比率}

根据算法 4:

(1) 选取 *SPI_I*set 中的第 1 个指标:I_1=需求变更率

(2) 根据算法 2,该指标对应的特征集合 T_1=*MapI*to*T*(*R_ReqChan*)={需求稳定性/*Req_Stab*}

从表 1 可见:需求稳定性=〈*Req_Stab*,*Speci-prod*,{*R_ReqChan*,*R_ReqUncons*}〉

(3) 需求稳定性.*cls*=*Speci-prod*,是特殊特征

所以继续根据算法 1 可以得到该特征对应的过程及过程目标集合:

$PnG\tau$=*MapT*to*G*(*Req_Stab*)={〈需求分析过程/*ReqPro*,有效的需求管理〉}

(4) *AffGoal*={〈需求分析过程/*ReqPro*,有效的需求管理〉}

(5) 选取 *SPI_I*set 中的第 2 个指标:I_2=投入产出比率,重复(2)~(3),可得

*MapI*to*T*(*R_CostReqCon*)={需求管理的有效性/*ReqM_Effe*}

$PnG\tau$=*MapT*to*G*(*ReqM_Effe*)={〈需求分析过程/*ReqPro*,有效的需求管理〉}

(6) *AffGoal*=*AffGoal*+$PnG\tau$={〈需求分析过程/*ReqPro*,有效的需求管理〉}

(7) *SPI_I*set 中的所有指标都已选取

输出:*AffGoal*={〈需求分析过程/*ReqPro*,有效的需求管理〉}

输出结果表明,这两个指标缺陷指征应该关注的过程目标集合是需求分析过程的特殊目标"有效的需求管理".亦即表明,该目标还没有达到预期的期望,需要进行改进.

6 小 结

过程度量的目的是进行科学的过程管理,所以度量只是组织经营管理的手段而不是目标.度量不可避免地需要成本,无序无效的度量不能解决组织的问题,过度的度量由于不能让组织的管理者看到合理的回报,而对管理失去信心,只有合适的度量才能给组织带来有效的利润空间和合理的管理成本投资回报,形成良性的经营管理循环.本文提出的 AMM 模型旨在解决软件组织合适度量的问题,提出了一种根据过程控制和改进的需要,进行主动度量的思想和方法.AMM 模型是一个开放的模型,不仅支持度量过程在组织关注的过程目标驱动和引导下,增加或裁减度量活动,形成适当的度量过程,还提供对过程改进的决策支持和模型本身的自我优化和改进.AMM 模型为软件组织开展合适而有效的过程度量提供了方法和技术上的有力支持.

References:

[1] Basili VR, Caldiera G, Rombach HD. The goal question metric approach. In: Encyclopedia of Software Engineering. John Wiley & Sons, Inc., 1994,2:528−532.

[2] Park RE, Goethert WB, Florac WA. Goal-Driven software measurement—A guidebook. Technical Report, CMU/SEI-96-HB-002, Software Engineering Institute, Carnegie Mellon University, 1996.

[3] Jacob AL, Pillai SK. Statistical process control to improve coding and code review. IEEE Software, 2003,20(3):50−55.

[4] Jalote P. Optimum control limits for employing statistical process control in software process. IEEE Trans. on Software Engineering, 2002,28(12):1126−1134.

[5] Florac WA, Carleton AD. Measuring the Software Process—Statistical Process Control for Software Process Improvement. Massachusetts: Addison Welsley, 1999.

[6] Murugappan M, Keeni G. Blending CMM and six Sigma to meet business goals. IEEE Software, 2003,20(2):42−48.

[7] Lawler J, Kitchenham B. Measurement modeling technology. IEEE Software, 2003,20(3):68−75.

[8] CMMI Product Team. Capability maturity model integration (CMMI^{SM})—CMMI^{SM} for systems engineering, software engineering, integrated, product and process development, and supplier sourcing (CMMI-SE/SW/IPPD/SS, V1.1). Continuous Representation, Technical Report, CMU/SEI-2002-TR-011, ESC-TR-2002-011, 2002.

[9] CMMI Product Team. Capability maturity model integration (CMMI^{SM}), CMMI^{SM} for systems engineering, software engineering, integrated, product and process development, and supplier sourcing (CMMI-SE/SW/IPPD/SS, V1.1). Staged Representation, Technical Report, CMU/SEI-2002-TR-012, ESC-TR-2002-012, 2002.