

软件度量的操作模型

侯红 王强华 郝克刚

(西北大学计算机科学系 西安 710069)

摘要 软件企业在实施软件度量的过程中面临最主要的障碍是如何将软件度量转换成可实施的、体现组织过程特征的度量实施过程模型。为解决这一问题,本文基于“软件度量过程也是软件”的思想给出组织级实施软件度量的操作模型 OMMP,在此基础上,对基于度量的过程管理框架进行讨论,将度量集成在过程管理框架中。

关键词 软件度量,软件过程管理,软件过程改进

Operational Model of Measurement Process

HOU Hong WANG Qiang-hua HAO Ke-gang

(Department of Computer Science, Northwest University, Xi'an 710069, China)

Abstract The critical obstacle of implementing organizational software measurement is how to transfer metrics into an operational process model which characterizes the organizational software process. To solve the above problem, the operational model of measurement process(OMMP) is proposed, which is based on the idea of the measurement processes are software too. Then the process management framework based on the metrics is analyzed.

Keywords Software measurement, Software process management, Software process improvement

1 引言

1987年 Osterweil 提出“软件过程也是软件”,1999年 Morisio 将此引申为“软件度量过程本身也是软件”^[1],按照这样的观点用于软件过程的概念和工具可复用于度量过程中。文献[1,2]均给出了项目级软件度量过程的操作模型,但仅仅将度量看作一个独立的系统,未考虑度量过程与软件过程的集成问题。文献[3]给出了集成度量的过程管理模型,但就如何集成的问题没有展开深入的讨论,而且并没有将度量过程看作是软件过程系统中的一个独立系统。本文结合实践的经验教训在总结整理相关工作的基础上,给出组织级软件度量过程的操作模型 OMMP(Operational model of Measurement Process)和集成度量的软件过程管理环境。

2 软件度量的操作模型

通常度量被看作是软件过程的一项功能。事实上一个有效的软件度量需要资源和角色(项目经理或度量经理、工具、设备、数据收集的工作量、数据讨论和表示的工作量),产品(度量计划、度量规格说明书、度量数据库、度量报告)和阶段(测量元的定义、收集和分析)。所有这些元素定义了度量过程。

2.1 软件度量过程的各个阶段



图1 软件度量过程

如图1所示,如果软件度量看作是一个黑盒子,软件度量

过程整体的输入为组织的目标,度量的输出为管理层(高级经理、项目经理等)所需要的决策信息^[2]。

软件度量过程不仅是编码,也需要需求、设计等。图1中的方框部分的详细描述如图2所示^[1]。它们可分为计划度量、应用度量和运行阶段三大部分。计划度量包括需求定义、设计和实现。应用度量指度量的实际运行。支持阶段包括配置管理、项目管理和质量保证。

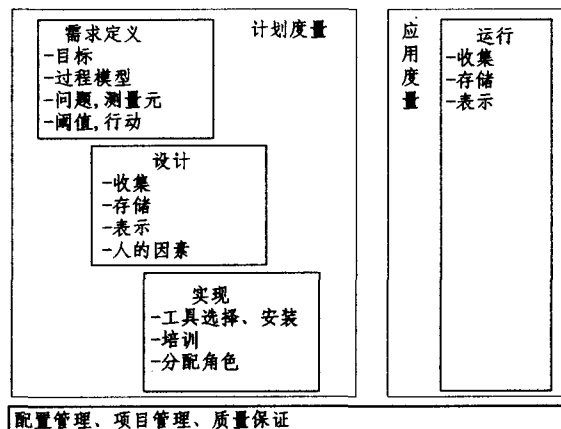


图2 软件度量过程的各个阶段

(1) 需求定义

需求定义阶段在度量过程中,就是定义度量什么?为了定义度量什么,必须陈述目标。在定义测量元之前,需定义一个描述性的过程模型。因此需求定义被分解为目标、过程模型化、问题和测量元的定义及相关的行动4个子阶段。初始目标的定义会影响过程模型和测量元的定义。组织决策时会用到测量元,因此通过某测量元限定的阈值来采取相关

行动。

目标的定义:在这个子活动中定义度量程序的目标。这些目标通常要与公司的和特定的项目质量目标一致。

过程模型的定义:首先需要给出一个描述性的过程模型。理想的方法是建立一个新的广义的模型作为当前过程模型的快照。

问题和测量元的定义:在这个子阶段的目标是详细定义问题和测量元。

定义阈值和行动:前面定义的测量元用于监视和控制软件过程。本阶段对定义的每一个测量元构成反馈控制。通过对每一个测量元定义一个上、下线和—个可接受的值的范围来实现。如果度量的实际结果超过了阈值,那么需对采取的行动达成—致意见。如若某模块的缺陷密度超过上限,模块需重新进行设计;而若该模块的缺陷密度低于下限,则可减少已计划的测试的工作量。

实际上本阶段为测量元的定义进行了早期的验证。如果很难在软件过程中识别针对该测量元采取的行动,说明此测量元是否被错误地定义。

需求的确认和验证(V&V):在需求阶段执行的第一次确认和验证活动是对目标的验证。

(2) 设计

到目前为止,测量元已经定义好了,但是可以有多种有关数据的获取和表示方式。设计阶段要解决的问题是测量元是如何得到的,也就是选择测量元的收集、存储和表示方式。在这一阶段中,要特别考虑人员的因素,因为人员因素会影响到度量过程。

在设计度量过程后,会改变软件过程本身。可以把度量过程的引入看作是软件过程的变更请求。

分清楚度量的需求和度量设计的不同是非常重要的。在度量设计过程中的变更请求可能是关于需求的也可能是关于设计的。有关需求(目标和测量元)的变更请求,如设计的测量元已经不需要了,需要删除掉。这类变更请求会影响到需求,而有关设计的变更不会影响到需求的变更。因此设计时需要考虑—定的可扩展性和灵活性。

收集:对于每一个基测量,度量的项目经理需要决定是手工收集还是自动地收集数据。数据的收集来自被执行的软件过程,来源于日常的报表,如项目周报、月报、个人日报等,需要确定数据收集的流程(谁来负责该测量元的数据收集、什么时候收集以及收集的频率等)并设计数据收集的方式(如屏幕显示格式、纸面报表的格式)。同时对于每一个基测量还需要确定验证技术,谁来验证,什么时间验证。

对于所有的派生测量要确定什么时间、由谁以及如何计算。基测量存储在数据库中,派生测量自动地通过查询语言实现。

存储:尽管基测量可以纸面的或文件形式存储,在工业界比较好的做法是放到数据库中。

表示:选择对测量结果最恰当的显示方式。可以用盒图、散点图、柱图等来表示。确定数据的结构、内容和显示方式并根据角色的不同确定数据的存取权限和访问方式。

设计验证:确认度量设计与需求是—致的,针对数据的收集检查其内部的完整性和—致性以及各种表格和报表的生成环境、培训的适合度、度量报告的易读性等。

分析和设计阶段结束之后形成度量计划书。软件度量计划书以结构化的方式,清晰地给出度量的目标、度量的进度、度量的角色、度量的指示器、基测量和派生测量等的描述。该

度量计划书成为实施度量的指导性的文件。

(3) 实现

购买工具、安装和本地化、培训员工等就位,并分配了组织级度量的角色,数据库以及数据的表示等全部准备就绪。在本阶段的最后,应对所有的产品按照需求和设计进行验证。

(4) 配置管理、项目管理

针对配置管理和项目管理,把软件度量看作是一个软件,可以按照其他管理软件项目的同样的方式来做。从项目管理的角度看,度量是有着计划、资源分配、跟踪和再计划的一个完整的项目。从配置管理的角度看,一系列的文档(度量计划书、数据库、表格、流程)应当进行配置管理。

(5) 质量保证

一个度量过程的质量保证意味着要确保所收集的测量元是有效的并且是有用的。这里有效意味着满足其表示条件,有用意味着它们被用来做决策。

在度量过程运行之前,各个阶段的验证活动应当纳入质量保证之下。在度量过程运行之中,质量保证活动验证收集到的数据并且定期地(如每月或每季)进行度量需求的确认和验证。另外,更进一步地对有关度量过程本身的测量元和缺陷数据进行收集和分析。

(6) 运行

度量过程按照需求和设计的定义在组织中的多个项目中被执行。来自项目和生产的数据被收集、存储和表示出来,同时定期地给出开发人员和管理人员所需要使用的各类度量报告。

2.2 产品

不同的阶段会产生不同的度量产品,主要有下列产品。

(1) 需求文档:目标、过程模型、测量元的定义。

(2) 设计文档:数据收集的表格的设计、数据库范式、可供选择的工具和使用情况、度量报告的结构的表现方式、培训课程的内容和进度安排。

(3) 实现:安装的工具、度量的工作数据库、数据收集的工作流程和工具、受过培训的开发人员。

(4) 运行:测量元数据库的定期更新、定期发布的度量报告。

(5) 质量保证:度量运行前的质量保证报告、运行中的度量需求的确认和验证报告、度量过程的缺陷报告。

2.3 角色

度量的角色主要有度量经理、管理人员(高级经理和项目经理)、开发人员和质量经理。各角色的职责如表1所示。

重要的是,各个角色之间相互制约。质量经理控制度量经理的所有活动,度量经理控制由高级经理和项目经理陈述的目标,度量经理控制由开发人员收集的数据。

表1 度量过程的角色责任图

角色	责任
度量经理	需求定义、设计和实现,
项目经理	度量目标的定义
高级经理	度量目标的定义
开发人员	数据收集、数据的精度和准确度
质量经理	质量保证

2.4 生命周期

图2所描述的各个阶段,没有时间关系。理想的度量过程按照需求、设计、实现和运行顺序执行,同时并行执行的过程

程为项目管理、配置管理和质量保证。

当缺陷报告和变更请求单获得批准后,度量过程进入了一个新的循环。因此度量的生命周期为演进式的开发周期,可采用增量或迭代式的模型,在每一次增量和迭代的周期中,可以看作是一个简化的 V 模型。

3 基于度量的软件过程管理框架

3.1 过程管理的职责

20 世纪中期, W. Edwards Deming 在 Walter A. Shewhart 工作的影响下,引进了过程管理和持续改进的概念。他认为^[4]:为了提高竞争力,需要通过关注产品生产和服务的过程来提高质量和生产率;要确保这些过程得到严格的管理,对性能不佳的过程的管理是改进其过程而不是责备工作人员;要认识到所有的过程都在不断的变化,变化是改进的机会,改进是为了适应变化;在决策的过程中要考虑变化,并从过程取得数据以指导决策。因此,要使这种思想被实践所接受和使用,就要求对过程数据进行收集和分析,使用定量的方法来监控软件过程。如果对过程不能进行度量,就不能用数字表达它,那么对它的了解也就很贫乏,它可能只是知识的开始,在思想上还远远没有进入科学的阶段^[3,6]。

为了对过程管理有共同的理解,首先介绍过程管理的四个核心职责,即定义过程、度量过程、控制过程和改进过程,如图 3 所示^[3]。

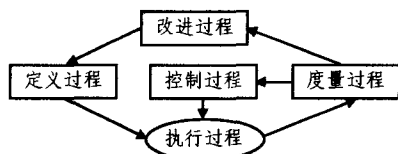


图 3 过程管理的 4 个职责

(1) 定义过程

定义过程是指对一个软件过程进行定义,为控制和改进这个过程创造严格的管理规范和结构化的环境。定义、实现和支持一个过程是为了达到以下四个目标:设计的过程应满足商业和技术目标;确定和定义与过程性能有关的问题、模型和度量指标;提供用来支持软件活动的基础设施;确保软件组织有执行这些过程的能力。

(2) 度量过程

度量过程是指观测一个组织在推行过程中的性能数据,其测量值是检测实际性能与可接受性能之间偏差的基础,同时也是获得过程改进机会的基础。过程度量的三个主要目标是:收集过程性能的数据;保留数据,用以分析过程性能和评价过程的稳定性和能力;为预测未来的过程性能提供基准,以描述趋势和确定改进的机会。

(3) 控制过程

控制过程是指使过程的各项性能处在其正常的界限之内,使过程的行为始终如一。因此控制过程包括以下四方面的任务:度量(获得有关过程性能的信息);检测(分析信息,确定过程中由可确定的因素造成的异变);纠正(采取行动排除过程中的这些异变及其带给产品的影响);建立软件过程改进的基础设施。

(4) 改进过程

改进过程是指通过改变过程或者使用其它更有效的子过程替代现有的子过程等手段来提高过程的能力。一个组织进

行过程改进有三个主要目标:理解现存过程的特点和影响过程能力的因素;计划、论证和实现那些改变过程的措施;评定风险和效益,优选过程改进方案。

总的来说,其研究目标包括三个方面:对于项目开发过程性能的控制和持续改进;从项目开发过程中进行组织学习;以过程管理系统为框架,实现对协同的支持。

3.2 过程管理与项目的关系

图 4 对图 3 进行了扩展,显示了过程管理和项目的关系并描述了度量活动与这些职责的关系^[3]。

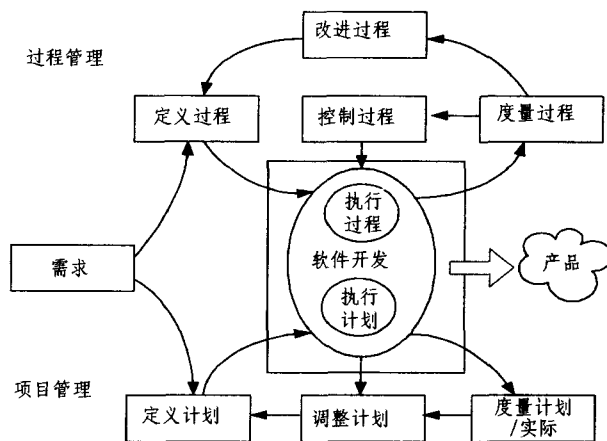


图 4 过程管理与项目的关系

如图 4 所示,软件项目团队基于产品需求、项目计划和已定义的软件过程来生产软件产品。项目经理使用度量数据指导其识别和特征化需求;准备一个可执行的项目计划;实现计划;监控在计划中描述的工作进展和状态。与此同时,过程管理使用相同的数据和其它相关的度量来控制和改进软件过程本身。这意味着组织可以采用一个公共的框架建立和维护度量活动,为两者的管理提供数据。

3.3 基于度量的软件过程管理框架

图 5 对图 4 再次以不同的方式进行了扩展,它显示了过程管理和计划和应用度量过程间的关系。

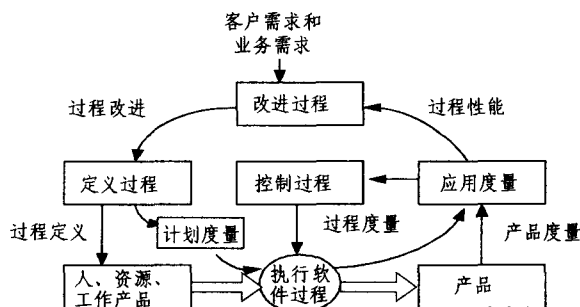


图 5 度量过程与过程管理的关系

图 5 中度和过程管理之间的交互总结如下。

(1) 定义过程

过程是为产生特定的最终结果(通常是一种产品或服务)而参与的人、材料、能力、设备和流程的有组织的集合。在选择和实现度量之前,首先应识别每一个有贡献的过程元素,对过程实施和管理所要达到的目标中,这些过程元素的作用要有透彻的理解。这期间,可使用数据流图和控制流图帮助我们描述。

(2) 计划度量(图 2 中的需求、设计和实现三个阶段)

度量的计划建立在对已定义(明确)的过程的理解基础上。在这里,与产品、过程和资源相关的问题和属性被识别出来;产品和过程质量的测量元被选择和定义了;跟踪和评估过程性能的度量数据的收集和使用集成到了软件过程中。

(3) 执行软件过程

过程被软件组织执行。被识别出的产品、过程和资源的属性在每一个软件过程的执行和结束的时候进行了度量。

(4) 应用度量(图 2 的运行过程)

将测量元应用到软件过程执行中所进行的软件度量中。软件过程的数据和过程所产生的软件产品的数据被收集、保留和分析以便它们被用于控制和改进过程中。

(5) 控制过程

如果产品的度量或属性的性能显示过程的偏差处于不期望或不可预测之中,必须采取行动排除造成这些问题的原因,将偏差稳定下来,同时如果可能的话,恢复到过程的正常性能状态。

(6) 改进过程

一旦度量结果表明所有的过程偏差来源于过程本身(也就是说,偏差是过程所固有的),过程性能数据可被依赖和用于指导我们的行动以改变性能的水平。改进行动可用来更新和演进过程的定义,其收益亦需通过度量来验证。

结束语 从基于度量的软件过程管理框架,我们可以体会到,软件度量是软件过程的一部分,不能将其割裂开来。如果将软件过程看作完整的一个系统,度量过程就是软件过程的子系统。因此在实施度量时,需要用完整的系统的观点来

看待,同时要关注软件度量过程与软件过程的接口,将软件度量与软件过程集成起来,使软件过程度量成为软件过程执行中的副产品。成功的过程管理需要将度量过程、产品和资源建立在一个规范的、受控的度量过程的基础之上。一个成功的度量过程不仅仅包括清晰定义的数据同时也包括定义、收集、分析相关测量元的操作过程。软件过程的控制、预测和改进需要可信的和准确的数据,这意味着度量数据必须真实地反映过程本身,同时用与数据相匹配的技术进行分析^[6]。

创建度量的过程模型是软件度量过程的难点之一,且通过实践证明度量的过程模型不仅需要有一个清晰的概念模型,更需要有一个可实施的操作模型。度量过程模型的建立有助于减少对度量过程的误解,保证度量的有用性,避免陷入度量的困境,提高度量的有效性,为组织度量的实施奠定基础。

参考文献

[1] Morisio M. Measurement processes are software, too[J]. The Journal of Systems and Software, 1999,49:17-31
 [2] Perkins T, Peterson R, Smith L. Back to the Basics: Measurement and Metrics. CROSSTALK The Journal of Defense Software Engineering[J], December 2003
 [3] Florac W A, Park R E, Carleton A D. Practical Software measurement, Measuring for Process Management and Improvement [J]. GUIDEBOOK CMU/SEI-97-HB-003
 [4] Deming W E. Out of the Crisis, Cambridge[M]; MIT Press, 1982
 [5] 吴超英. 关于软件过程改进中测量问题的研究与实践[J]. 计算机应用与软件, 2001(11)
 [6] 侯红, 郝克刚. 软件测量刻度及选择方法[J]. 计算机科学, 2005,32(5)

(上接第 259 页)



图 6 初始的综合检索界面 图 7 经三次反馈后的检索结果

图 2、图 5 分别是从小图、动物这两类图像库中随意选择一幅图像作为查询图。对图 2 的检索结果可见,初次检索后经反馈得到和原查询图从不同视角相关的实际图像;对图 5 的查询结果可看出系统对底层特征特性具有一定的稳定性,可检索出经各种变换的相似图像。从实验结果可以看出,对比较复杂的、在各种视角和不同实际场景环境下的实际图像具有较好的检索结果。

为了较全面地评价本文提出的综合多特征和相关反馈 ROI 图像检索技术,设计了 5 类查询:花、食物、动物、竞技运动、自然景物。从每一类中随机抽取 10 幅图作为查询图像,构成 50 次查询,采用查准率作为系统性能的评价,综合 50 次查询,得到相应的平均查准率。

表 1 五类图像检索的平均检索率

图像类别	初次检索	第一次反馈	第二次反馈	第三次反馈
花	0.67	0.76	0.82	0.89
食物	0.58	0.70	0.79	0.84
自然景物	0.53	0.68	0.74	0.80
动物	0.45	0.53	0.66	0.78
竞技运动	0.42	0.62	0.75	0.86

实验评价结果如表 1 所示,其中初始查询中不同特征的权重相同,经三次反馈后平均检索率得到很大提高。实验结果表明,综合多种特征进行检索本文的综合多特征和相关反馈 ROI 检索算法能显著提高图像检索性能。

结束语 本文对感兴趣区域检索的研究简单、有效且符合人类的视觉概念,同时相关反馈技术实现了人机交互,更进一步接近用户意图。将这两种技术再融合进多特征的图像检索中,进一步减小了底层特征和高层语义之间的语义鸿沟。

参考文献

[1] Smeulders A W M, Worring M, Santini S, et al. Content-based image retrieval at the end of the early years[J]. IEEE transactions on pattern analysis and machine intelligence, 2000, 22 (12):1349-1379
 [2] Moghaddam B, Biermann H, Margaritis D. Defining image content with multiple regions-of-interest//IEEE Workshop on Content-Based Access of Image and Video Libraries. 1999,6
 [3] Rui Y, Huang T S. Optimizing learning in image retrieval // IEEE Conf. on CVPR. South Carolina, USA, 2000
 [4] Beis J S, Lowe D G. Indexing Without Invariants in 3D Object Recognition. IEEE Trans. PAMI, 1999,21(10):1000-1015
 [5] 王海霞, 覃团发. 综合 MPEG-7 中颜色特征的图像检索方法[J]. 计算机应用研究, 2005,22(3):164-164
 [6] Gotlieb C C, Kreyszig H E. Texture descriptors based on co-occurrence matrices. Comput. Vis., Graphics, and Image Proc., 1990, 51:70-86
 [7] 金莲芳, 覃团发, 帅勤. 一种基于 MPEG-7 形状特征描述符算法[J]. 计算技术与自动化, 2005,24(4):56-56
 [8] 朱虹, 等. 数字图像处理基础[M]. 北京: 科学出版社, 2005:109-120
 [9] 王海霞, 覃团发. 一种基于 MPEG-7 中主颜色提取的改进方法[J]. 计算机工程, 2005,31(16):185