

软件过程度量的过程模型及其应用研究

徐如志¹ 钱乐秋¹ 张敬周^{1,2} 朱三元²

¹(复旦大学软件工程实验室 上海 200433) ²(上海计算机软件技术开发中心 上海 200233)

摘 要 软件度量是软件工程中最活跃的一个研究领域。本文给出了一个软件过程度量的过程模型,定义了实施软件过程度量的角色、内容、主要活动及相关的支持环境,重点阐述软件过程度量的数据采集、验证和分析活动的目标、任务和方法。通过一个软件项目的度量过程实例,研究并确认该模型在过程评估和改进方面的作用和意义。本文的研究对于规范和改善组织的软件过程度量,提高组织的过程能力成熟度,具有一定的指导意义和广泛的应用价值。

关键词 软件过程 过程度量 过程评估 过程改进

RESEARCH ON PROCESS MODEL AND ITS APPLICATION FOR SOFTWARE PROCESS METRICS

Xu Ruzhi¹ Qian Leqiu¹ Zhang Jingzhou^{1,2} Zhu Sanyuan²

¹(Laboratory of Software Engineering, Fudan University, Shanghai 200433, China)

²(Shanghai Development Center of Computer Software Technology, Shanghai 200233, China)

Abstract Software metrics is one of the active research areas of software engineering. In this paper, a process model for software process metrics is presented, which defines the roles, contents, activities and related supporting tools and methods for the implementation of software process metrics, and gives an emphasis on the description of the goals, tasks and methods for the activity of process data collection, validation and analysis. A practical application of this model in a large-scale software enterprise shows that the model is significant on software process assessment and improvement. The research results are greatly helpful and can be widely applied to effectively regulating and improving software process metrics, and to promoting the mature level of an organization's process capability.

Keywords Software process Process metrics Process assessment Process improvement

1 引 言

过程管理是现代软件质量管理的核心内容,自上个世纪七十年代以来,出现了 CMM^[1]和 ISO9001、ISO/IEC12007 以及 Bootstrap 和 Spice^[2]等众多的软件过程质量模型或软件质量保证标准。虽然上述标准的侧重点不同,但这些标准或模型都强调通过实施软件过程度量来驱动软件过程的持续改进。

在度量驱动的软件过程改进方面,CMM 定义了从 1 级 ~ 5 级的成熟度级别,每个级别包含若干个关键过程域(KPA),通过从过程度量中得到的反馈信息,不断识别和改进各 KPA 存在的不足,逐步提升软件过程能力的成熟度^[1]。CMM I^[6]为实施软件度量提供了一个基本框架和规约基础,但对如何实施软件过程度量,如何采集和分析软件过程数据,以及使用哪些工具和方法并没有给出明确规定,因而在实施软件过程度量时,各个企业没有可以共同遵循的规范和指导准则,而是根据各自的理解各行其是。

本文结合当前实施 CMM/CMM I 的软件工程实践,以软件过程评价和改进为目标,给出一个软件过程度量的过程模型,定义了实施软件过程度量的角色、内容、活动及相关的支持工具和方法。介绍了常用的软件过程数据的采集、验证和分析活动的目标、任务和方法,给出了基于该模型实施软件过程评估与改进

的一个实例。

2 过程度量的过程模型

过程度量是针对所指定的软件过程,以某种方式使其过程能力指标实现合理的量化,从而以一定的标准衡量该软件过程的质量,过程质量表现在静态和动态两个方面,过程能力是过程质量的动态表现,是通过过程运作而体现的^[2]。

过程度量是实施过程评价和过程改进的基础。过程评价是按照软件工程的一系列标准对软件过程的质量进行评定而使软件过程不断改进和优化的系列活动。通过过程度量,能够认知软件过程的现状,从而使已经或可能将要失控的项目得到合理的控制,同时还能评价过程改进的效果,达到过程评价的认知目的;过程改进是利用过程运作所获得的反馈信息,发现当前软件过程存在的问题和缺陷,提出改进的意见,进而实现软件过程的改进和完善。过程改进的关键是发现软件过程中所存在的问题和缺陷,而过程度量正是发现问题和缺陷的必备手段。

过程度量与过程运作紧密相关,是一个由多个角色在一定

收稿日期:2004-02-25。本项目得到“上海市科学技术委员会研究项目(No. 035115026)”的资助。徐如志,博士生,主研领域:软件过程工程,软件复用,软件项目管理。

的条件和约束下,按计划进行的一系列活动^[5]。因此过程度量本身也是一个过程。图 1 为本文给出的一个软件度量的过程模型,它标明过程度量各项活动的输入、输出、约束条件以及所需要的支持工具和方法。其中每个活动的输入来源于上一个活动在当前时间的数值及相应的外部输入,使得当前时间得到的度量分析结果总能反映截至当前时间为止过程的运作情况,进而达到过程评价、过程控制及过程改进的目的。

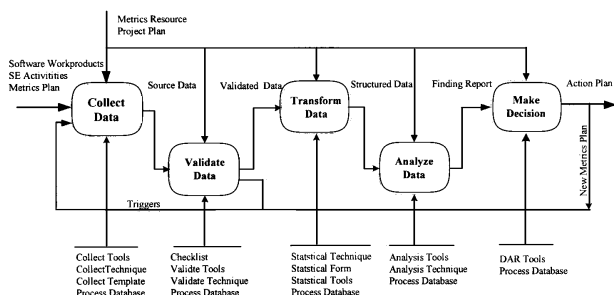


图 1 软件度量的过程模型

2.1 过程度量的角色及其职责

软件过程度量的约束体现为实施过程度量所需要的资源,其中最主要的资源是人力资源,下面给出实施软件过程度量的相关角色及其职责定义:

- 项目经理 (PM) 或软件工程组负责人 负责组织在项目计划和实施过程中,对软件过程数据的收集和审核,与高层经理分析项目过程的执行效率,必要时实施纠正措施或预防措施。并负责在每个阶段结束时,将软件质量保证组和软件工程过程组验证通过的项目过程数据录入过程数据库。

- 软件质量保证组 (SQA) 在项目中负责对项目度量的过程及度量数据进行审核,评审其有效性,并将异常情况汇报给项目经理和高层经理。

- 软件工程过程组 (SEPG) 定期汇总和分析软件项目的度量情况,在组织层面上负责统计软件过程的度量数据,分析和发布组织的度量标准,建议过程的改进,并提出对过程基线的修订意见。

- 高层经理 (SM) 负责为项目组实施过程度量提供人力资源以及必要的工具和培训;负责总结所有当月项目执行数据;分析每个项目的效率,审批项目负责人提出的度量报告和度量计划,必要时提出纠正意见或建议。

2.2 过程度量的输入及输出

软件过程度量的对象是软件过程活动及软件工作产品,因此软件过程度量的输入是软件项目组每天的活动和每个阶段的工作产品,过程度量的另一个输入项是软件过程度量计划,过程度量计划是根据组织和项目的目标按照 GQM (Goal-Question-Metrics) 度量模型,由项目组结合项目实际情况制定的。度量计划中包含实施度量的目标和范围,所选择的度量元的定义,度量过程、工具、方法及人力资源等,由于软件过程度量无法孤立于它所度量的过程而独立运行,通常把要执行的度量活动集成到软件过程中,因而过程度量计划一般作为项目计划的一个组成部分。

过程度量的目标是对所度量过程的评价和改进,因此过程度量的输出是指导下一步过程活动的行动计划,这一计划往往触发新的度量过程的产生,度量的阶段工作产品包括项目状态报告、项目状态汇总报告,以及项目过程评价报告等。

2.3 过程度量的主要活动

软件过程度量由数据获取和度量分析组成,其中数据获取包含数据采集、数据验证两个活动,而度量分析包括数值转换、数据分析和度量决策三个活动。

- 数据采集 是过程度量的基础,用于收集与所要求的度量值相关的基础数据,鉴于过程运作及过程度量的复杂性和时效性,为避免客观数据的主观化,保证数据的有效性、一致性和准确性,合理选取数据来源,开发各种各样的标准化的报表,使用有效的数据收集工具是实现数据获取的重要手段。图 2 为本文给出的一个用于工作量数据采集的员工工作日志填报格式,在员工工作日志里记录了员工编号、项目编号、阶段编号、任务编号、工作内容、工作结果、工作类型及为完成每项任务所花费的时间等。

图 2 员工工作日志填报格式

- 数据验证 包括两个方面的活动内容:一方面是检查数据采集的过程是否按计划执行;另一方面要验证每个度量活动的输入和输出是否是正确的、满足要求的,除验证所采集数据的真实性外,还要确保所采集的数据是一致的和有效的,经过验证的过程数据存储存储在过程数据库或以文件的形式存放在文件系统中。

- 数值转换 数据获取阶段得到的数据是杂乱无章的,必须经过加工处理即进行适当的数值转换才能用于度量分析,数值转换的任务之一是进行数据标度,实现从观察得到的状态到一个数值范围的映射,其次是生成用于总结所有观察数据的统计数据,亦即对所采集的过程数据按照一定的数据模型进行转换。

- 度量分析 是利用适当的工具和方法,将经过数值转换汇总或计算的度量结果与相同格式的基线数据进行比较,判断其能否反映度量的目标,并发现所度量的过程存在的问题,显然,过程分析的结果是进行过程评价和过程改进的依据。表 1 为本文给出的软件过程度量的统计汇总数据模型。项目结束时,项目经理、SEPG 和 SQA 人员,根据对项目估算和实测的软件过程数据。按照表 1 所示的过程度量数据模型进行数据转换和汇总。

- 度量决策 管理者在清晰地了解到当前过程存在的问题或需要改进的内容后,需要按照一定的决策分析处理 (DAR) 方法做出下一阶段过程度量的目标和内容,从而引发在新的软件过程周期中实施新的过程度量。

获取真实有效的过程数据是实施过程度量的基础,合理地设计度量数据模型是实施过程度量进而完成过程评价、预测和过程改进的关键,正确、恰当的分析结论带给度量活动的将是新的问题着眼点。

2.4 过程度量的基础设施或支持工具

实施软件过程度量需要软硬件两个方面的支持。软件部分包括:过程度量应遵循的规程、方法、指南、模板以及各种各样标准化的报表等,这些报表包括产品描述、日志的使用、在线报告以及出错记录表格等。硬件部分主要是过程数据库和各类数据

表 1 软件项目过程度量数据模型

度量项 Metric	计算公式 Equation
进度偏差 (Schedule Variance)	(实际工期 - 计划工期) / 计划工期
工作量偏差 (Effort Variance)	(实际工作量 - 计划工作量) / 计划工作量
规模偏差 (Size Variance)	(计划规模 - 实际规模) / 计划规模
需求稳定指数 (Requirement stability Index)	变更的、增加的、删除的需求数 / 需求总数
缺陷密度 (Defect density)	发现的缺陷总数 / 规模
残存缺陷密度 (Residual defect density)	系统测试后发现的缺陷数 / 规模 (KLOC)
效率 (Productivity)	每人天的编码行数 (软件的代码行数 / 总工作量)
工作量分布 Effort distribution (%)	
软件需求规约 (SRS)	SRS的工作量 * 100 / 整个项目的工作量
设计 (Design)	Design的工作量 * 100 / 整个项目的工作量
编码 (Code)	Code的工作量 * 100 / 整个项目的工作量
测试 (Testing)	Test的工作量 * 100 / 整个项目的工作量
项目管理 (PM)	PM的工作量 * 100 / 整个项目的工作量
质量保证 (QA)	QA的工作量 * 100 / 整个项目的工作量
培训 (Training)	Training的工作量 * 100 / 整个项目的工作量
配置管理 (CM)	CM活动的工作量 * 100 / 整个项目的工作量
支持活动 (Support)	Support活动的工作量 * 100 / 整个项目的工作量
其它 (Others)	其他所有项目活动的工作量 * 100 / 整个项目的工作量
缺陷分析 Defect distribution (%)	
分析 (Analysis)	需求类别的缺陷数 / 总缺陷数
设计 (Design)	设计类别的缺陷数 / 总缺陷数
编码 (Code)	编码类别的缺陷数 / 总缺陷数
编写文档 (Document)	文档类别的缺陷数 / 总缺陷数
其它 (Others)	其他类别的缺陷数 / 总缺陷数
过程效率 Process efficiency (%)	
SRS评审效率	需求评审发现的需求类别的缺陷数 / 总的需求类别的缺陷数
设计评审效率	设计评审发现的设计类别的缺陷数 / 总的设计类别的缺陷数
代码评审和单元测试效率	代码评审发现的编码类别的缺陷数 / 总的编码类别的缺陷数
测试效率	包括系统测试在内的测试阶段找出的错误数 / 在测试阶段内和测试阶段以后发现的错误数
缺陷排除效率	直到系统测试包括系统测试发现的缺陷数 / 总的缺陷数

* 除编码外还包括单元测试 (Unit testing)

表 2 实际项目过程度量数据与组织的过程能力基线比较

度量类型	组织的基线数据	开发项目的实际数据	偏差 %
进度偏差 Schedule Variance	10.0%	16.8%	68.0(+)
工作量偏差 Effort Variance	12.5%	22.5%	80.0(+)
生产率 Productivity	42.5代码行 / 人天	41代码行 / 人天	3.5(-)
需求稳定指数 RSI	0.55	0.52	5.1(-)
工作量分布 (%)			
a SRS	10	12	20.0(+)
b Design	25	18	28.0(-)
c Code	35	32	8.6(-)
d Testing	30	38	26.7(+)

表 3 实际缺陷度量分布

项	通过评审和单元测试发现的缺陷数目	通过测试发现的缺陷数目	本阶段总的缺陷数目	每个阶段的缺陷分布
SRS	23	35	58	13.62%
Design	60	96	156	36.62%
Code	75	137	212	49.77%
Total	158	268	426	100%

采集及测试工具等。图 1 列出软件过程度量各主要活动所需要的规程、模板及支持工具,其中过程数据库已成为软件过程度量最重要的基础设施。

按照 SE 的定义,过程数据库中保存有过程的定义、过程实施情况的度量数据以及其它所有与过程相关的数据。在过程数

据库中,除保存组织的过程能力基线数据外,还包含关于软件项目的估算数据及经过 SQA 和 SEPG 确认的过程实测数据。此外,过程数据库中还包含软件过程组总结出来的与过程改进有关的经验和知识。这些数据可用于项目计划、过程控制和生产率分析等方面^[3,4]。(下转第 91 页)

4 结 论

VxWorks 产品映像设计是设备研发过程的最后关键环节。本文提出常规设计和软件可升级两种设计方案,均在设备开发中得到验证。前者将应用模块静态链接到操作系统模块,设备启动快,设计相对简单,属常规设计方法;后者将应用模块存放在文件系统中,动态链接到操作系统模块,设备启动慢,设计复杂,同时需要考虑软件升级的安全控制问题。但后者充分利用了网络处理器的软件可编程能力,提供了软件在线升级、功能重组的可能性。

参 考 文 献

- [1] Intel® IXP1200 Network Processor Family Microcode Software Reference Manual Intel, Inc
- [2] Tomado 2.0 Online Manulas—BSP Reference WindRiver, Inc
- [3] Tomado 2.0 Online Manulas—VxWorks Network Programmer's Guide, WindRiver, Inc
- [4] Tomado 2.0 Online Manulas—VxWorks Programmer's Guide WindRiver, Inc
- [5] 郑更生、郑炜煜,“基于 VxWorks 的产品映像设计”,《电子设计应用》,2003,4.
- [6] 成灵安,“一种板级支撑软件包设计”,《电子产品世界》,2002,10.
- [7] 孙祥营、柏桂枝,嵌入式实时操作系统 VxWorks 及其开发环境 Tor-nado,中国电力出版社,2002.
- [8] 蔡一兵、石晶林,“下一代网络设备核心单元—网络处理器应用研究”,《电子技术应用》,2004,1.

(上接第 16 页)

通过上述数据模型,将实际的软件过程度量结果与组织的基线数据进行比较,可以识别出所度量过程与组织当前的过程能力基线之间存在的差距,作为过程改进的依据和目标。

3 应用示例:一个金融软件开发过程评估

基于上述度量过程定义及数据模型,我们对一个中等规模的金融软件开发项目进行了度量和分析。

过程数据来源于员工工作日志、评审缺陷列表、软件测试记录单、用户反馈意见记录单等。上述数据由项目经理确认,QA 人员进行审核验证后,实时地录入公司的数据采集系统(DNS),根据项目编号、任务编号以及所处阶段的标注,项目经理利用 DNS 可统计出每个任务、每个阶段及整个项目的实际工作量、工作时间、费用、缺陷数据。表 2 和表 3 为从该项目中获得的度量数据统计结果,为了对项目的过程进行评估和分析,在表 2 中同时给出了相关度量数据的组织基线数据。图 3 为该项目工作量分布的直方图。

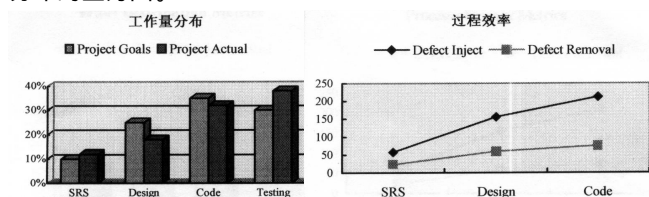


图 3 软件项目工作量分布度量

图 4 软件项目过程缺陷分布

由于组织级的缺陷基线数据尚未形成,在表 3 中列出该项目的实际度量数据。图 4 反映了表 3 统计的过程缺陷注入与缺陷清除数量的对比情况,从中呈现出软件过程的质量或评审效

率。

通过对上述度量数据的统计和比较,可以对该项目的软件过程做出如下分析和评价:

进度偏差和工作量偏差分别是基线数据的 68% 和 80%,而需求稳定指数和编码效率与基线数据基本持平,说明工作量的增加和进度的延迟并不是由于增加需求或需求不稳定引起的,从需求稳定指数和在需求分析阶段投入的工作量也反映出该项目在需求分析阶段的过程质量是比较高的。

设计和实现阶段工作量(投入)明显不足,从图 3 可以看出,除了在分析阶段实际工作量满足项目计划要求外,设计和实现阶段的实际工作量投入都少于项目计划的工作量,表 3 和图 4 的统计结果表明:由于设计阶段投入不足,软件设计不够精细,导致编码缺陷增加;而实现阶段的投入不足,必然简化单元测试及代码走查的过程,使得本应在实现阶段通过单元测试及代码走查发现和清除的缺陷被带入到最后的测试阶段,增加了后期测试阶段的工作压力,导致测试阶段返工工作量的大量增加和整个项目的延期。

图 4 和表 3 也反映出实现阶段同行评审和代码走查等过程质量或评审效率不高。在表 3 中统计的全部 426 个缺陷中仅有 158 个缺陷是通过过程评审发现和排除的,余下的 268 个缺陷是到了最后的测试阶段发现和排除的。而 Infosys 的过程统计数据恰恰相反^[3],软件过程缺陷总数中约有 65% 是通过同行评审、代码走查和单元测试发现和清除的,单元测试和代码走查阶段的工作量在通常情况下超过单纯编码的工作量。

对该项目软件过程改进建议为:加大设计阶段和实现阶段的工作量投入,加强对过程工作产品的同行评审,强化对单元测试和代码走查的执行力度,将更多的缺陷发现和清除在项目的早期阶段。

4 结 论

本文给出了一个软件度量过程模型,定义了实施软件过程度量的角色、内容、主要活动及相关的支持工具和方法;重点介绍了软件过程数据的收集和分析过程。

本文介绍了一家已通过 CMM3 级认证的软件企业进行软件过程度量的一些做法,阐述了本文给出的软件度量过程模型在实际中的应用,并以一个开发项目度量实例说明过程度量对软件过程评估和改进的意义。

参 考 文 献

- [1] Mark C. Paulk, Charles V. Weber, etc. Key Practices of the Capability Maturity Model, SM, Version 1.1. Technical Report, CMU/SEI-93-TR-025, ESC-TR-93-178, 1993.
- [2] 朱三元、钱乐秋、宿为民,软件工程技术概论,北京:科学出版社, pp. 117 ~ 134, 2002.
- [3] Pankaj Jajte, CMM in Practice: Process for Executing Software Projects at Infosys Addison-Wesley, 2000.
- [4] Xu Ru-zhi, Qian Le-qiu, CMM-based Software Risk Control Optimization Accepted for publication in Proceeding of the 2003 IEEE International Conference on Information Reuse and Integration (IRI2003), Las Vegas, October 27 ~ 29, 2003.
- [5] 任爱华、刘又诚译,度量软件过程:用于软件过程改进的统计过程控制,北京航空航天大学出版社,2002.
- [6] Dennis M. Ahen, etc. CMMI Distilled: A Practical Introduction to Integrated Process Improvement, Addison Wesley, 2002.