

# 基于 CMMI 的软件缺陷度量<sup>①</sup>

裘婉卿 叶枫 (浙江工业大学 经贸管理学院 浙江 杭州 310023)

**摘要:** 软件缺陷度量是提高软件质量, 进行软件过程评估的重要手段。把软件度量的相关技术应用到软件缺陷管理中, 根据 CMMI 的度量与分析框架, 并结合 GQIM 度量方法, 提出基于 CMMI 的软件缺陷度量模型。该模型可以指导软件企业实施软件缺陷度量, 达到更高的软件能力成熟度等级。

**关键词:** 软件缺陷; 软件缺陷度量; 能力成熟度模型集成; 度量与分析

## Software Defects Measurement Based on CMMI

QIU Wan-Qing, YE Feng

(College of Business Administration, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** Software defects measurement is an important method to improve software quality and evaluate software process. This paper introduces technology of software measurement into software defects management, according to CMMI measurement and analysis framework. Combined with GQIM, it present a CMMI-based software defects measurement model. This model can direct Software Corporations to measure software defects to improve their production.

**Keywords:** software defect; software defect measurement; CMMI; measurement and analysis

## 1 引言

随着计算机技术和网络技术的迅猛发展, 计算机软件日益渗透到社会生活的各个领域。软件的功能越来越强大、复杂程度也越来越高, 同时人们对软件质量的要求也越来越高。而软件缺陷是影响软件质量的关键因素, 不可避免地成为软件企业关注的焦点。而在传统的缺陷管理模式的基础上引入软件度量技术, 量化软件缺陷的管理, 是把握软件质量的有效途径之一。

## 2 CMMI

### 2.1 CMMI 概述

能力成熟度模型集成 CMMI 是由美国卡内基·梅隆大学软件工程研究所(SEI)在美国国防部的支持下, 于 1998 年开始研究的项目。SEI 在 2002 年 1 月正式发布了 CMMI1.1 版本。同时宣布 2003 年后停止能力成熟度模型 CMM 体系的维护工作, 转为维护

CMMI 和推荐 CMMI 认证。CMMI 是一套软件过程评估与改进的模型, 其根本目的就是改进软件过程, 有效地控制软件产品开发质量。

CMMI 总共有 25 个过程域, 软件企业通过执行分配在不同成熟度等级中的过程域进行过程改进。每个过程域由一组目标构成, 用来判断企业是否达到相应级别, 每个目标得到一组实践的支持, 这些实践是实现目标必须做的工作。过程得到改进, 企业的成熟度等级就得到提高。

### 2.2 CMMI 的度量与分析过程域

CMMI 与之前提出的 CMM 的明显区别就是, 在 CMMI 模型中“度量和分析(MA)”已经成为一个独立的过程域, 并且 CMMI 第四级的名称改为定量管理级。可见在 CMMI 中, 非常强调对软件开发过程的量化管理。某种程度上说, 软件过程度量的结果是软件过程改进的依据。

度量与分析过程域有两个特定目标, 一个是按信

① 基金项目:国家自然科学基金项目(70671095)

收稿时间:2009-07-30

息需要调整度量与分析活动，另一个目标是提供度量结果，如表 1 所示。从表中可以看出，两个 MA 目标的实践，定义了一个具有八个步骤的度量过程。各有四个特定实践映射到两个目标。针对第一个目标需要实现的活动有，确定满足信息需要的度量目的，指定满足目的所需要的度量，指示如何获得和存储数据，然后确定数据如何被分析和报告。而第二个目标，主要是收集数据，分析和解释收集到的信息，然后存储数据和分析结果，并就结果与项目相关人员汇报。

表 1 CMMI 的度量与分析过程域的特定目标和特定实践

过程域	特定目标(SG)	特定实践(SP)
度量与分析 (MA)	SG1 调整度量与分析活动: 根据指定的信息需要和目标对度量目标和活动进行调整。	SP1.1 建立度量目标
		SP1.2 指定度量
		SP1.3 指定数据收集和存储规程
		SP1.4 指定分析规程
	SG2 提供度量结果: 提供满足确定信息需求和目标的度量结果。	SP2.1 收集度量数据
		SP2.2 分析度量数据
		SP2.3 存储数据和结果
		SP2.4 沟通结果

CMMI 的基本度量范畴包括进度量与分析、规模度量与分析、成本度量与分析、工作量度量与分析、变更度量与分析和缺陷度量与分析。本文重点研究基于 CMMI 的软件缺陷的度量与分析。对于实施 CMMI 的企业来说，软件缺陷的度量必不可少。但是，CMMI 并没有详细地说明如何来建立和实施度量与分析活动，企业在开展度量活动时就会面临不知如何收集数据并进行分析等很多问题。

因此，本文提出一种基于 CMMI 的软件缺陷度量模型，利用 GQIM<sup>[1]</sup>建立度量目标和度量属性的关联，然后给出缺陷数据的分析方法，为企业实施软件缺陷度量提供指导。

### 3 基于CMMI的软件缺陷度量过程与分析

#### 3.1 基于 CMMI 的软件缺陷度量模型

软件缺陷度量框架大体可分为 4 个主要阶段：计划度量、实施度量、评价度量和改进过程。每一阶段都有自己不同的要求和实践，如图 1 所示。从建立度

量目标开始到最后的过程改进，整个度量活动是一个循环迭代的过程。

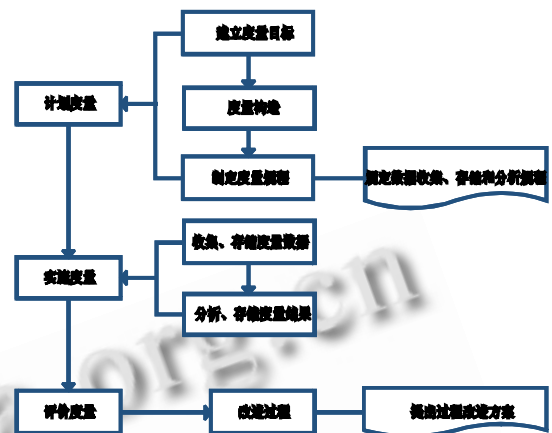


图 1 基于 CMMI 的软件缺陷度量模型

#### 3.2 计划度量

计划度量是对缺陷度量过程进行策划，为收集、存储和分析度量数据提供依据，制定在组织目标下的度量规程。具体包括一下步骤：

##### 1) 建立度量目标

进行软件缺陷度量的首要任务就是确定度量的目标，明确组织的信息需要，然后根据不同的目标来定义度量。度量要为组织的需要服务，软件缺陷度量作为软件过程改进的基础，度量的目标就是要提高软件开发质量。然后把目标细化，例如为了解产品缺陷发展趋势，了解项目进展情况，开发人员的效率，测试人员的效率等等。

##### 2) 度量构造

度量构造是计划度量的核心部分，度量构造就是将指定的信息需要和度量实体联系起来，是度量具体实施的依据。度量构造包括对度量元的设计，就是明确要度量的对象。度量元分为基本度量元和派生度量元。基本度量元是对单一缺陷属性的直接度量，派生度量元是根据两个或者多个基本度量元的数据计算出来。本文采用 GQIM 模型的基本思想，GQIM 是指“目标-问题-指标-度量” (Goal - Question - Indication - Measure)，该模型通过回答问题的形式分解度量目标，确定指示器，然后建立问题和度量的对应关系，如表 2 所示。

##### 3) 制定度量规程

在进行数据收集之前，还必须规定数据收集和存

储的规程以及数据分析和汇报规程,这样便于规范化的管理数据,避免数据量大的时候引起混乱,浪费组织资源。度量规程制定的是否完善关系到缺陷数据的收集,收集到的数据还要建立相应的数据库进行管理。缺陷管理中的一个关键因素是缺陷数据库<sup>[2]</sup>。度量规程的制定具体可从“4W1H”五个方面来考虑:

**WHO:** 数据收集、存储、分析和报告的相关责任人;

**WHEN:** 数据收集时间、分析时间、汇报时间;

**WHERE:** 数据收集位置、存储位置、分析位置;

**WHAT:** 收集什么、存储什么、分析什么;

**HOW:** 怎样收集、怎样存储、怎样分析(度量方法/分析工具)。

表2 问题-指示器-度量元分析

问题	指示器	度量元
各个模块的缺陷分布情况如何?	缺陷的模块分布(直方图)	各个模块的缺陷数
软件的缺陷密度?	缺陷密度(控制图)	软件的缺陷总数/软件规模(可以按模块度量)
缺陷的严重情况如何?	缺陷的严重程度分布(直方图)	缺陷的严重程度(致命、严重、一般和建议)
各阶段分别注入和清除了多少缺陷?	缺陷阶段分布(注入和清除矩阵)	各阶段注入的缺陷数 各阶段发现的缺陷数 (需求评审+设计评审,代码评审+单元测试,集成测试,系统那个测试,验收测试)
缺陷的类型情况如何?	缺陷类型分布(Pareto图)	各类型的缺陷数
发现的缺陷是否被及时修复?	缺陷的修复效率(折线图)	缺陷发现时间 缺陷消除时间
缺陷的发展趋势怎样?	缺陷趋势(折线图)	每周打开的缺陷数量 每周关闭的缺陷数量

### 3.3 实施度量

#### 1) 收集、存储度量数据

软件缺陷数据是软件过程改进决策最重要的管理信息来源<sup>[3]</sup>。实施度量的第一步就是根据度量计划,收集相关数据,可以通过一些自动化工具来收集数据,例如软件缺陷管理工具 Clear Quest 等。不论是采用自动收集数据还是人工收集,关键的一点是要保证数据的正确性,然后建立相应的数据库来保存数据以备

分析。需要收集的缺陷信息主要由缺陷属性、生命周期以及项目开发实践中的需要组成:

① 标题: 简要描述缺陷,方便检索;

② 标识号 ID: 唯一标识一个缺陷;

③ 状态: 缺陷当前的状态,可分提交、打开、验证、关闭等状态;

④ 时间: 缺陷发现的时间和关闭时间,用来计算缺陷的修复效率;

⑤ 严重度: 缺陷的严重程度,可分为致命、严重、一般、建议 4 个等级;

⑥ 模块: 缺陷所在的模块,用来统计缺陷的模块分布;

⑦ 来源: 缺陷是由软件开发过程中的哪个阶段引起的,可分为由于需求、设计、编码和测试的问题引起等;

⑧ 缺陷类型: 对缺陷的简单分类;

⑨ 提交者: 提交缺陷的测试人员;

⑩ 附件: 其他描述,例如与缺陷相关的文件的地址,可包括与缺陷有关的图表和文档等。

#### 2) 分析、存储度量结果

对缺陷数据的分析,可以帮助我们发现隐藏在数据背后的信息。按照度量构造中已定义的分析方法进行数据分析,得出分析报告,通过图表的形式直观的向项目相关负责人反馈信息。

缺陷度量过程中,了解缺陷在各个模块的分布情况是检测软件模块质量的重要手段,如图 2 所示。从图中可以看出 M4 和 M5 的缺陷数量明显比其他模块多,在测试过程中就需要多关注这两个模块,增加测试用例。另外, M7、M8 和 M9 的缺陷数量较少,说明这三个模块已经相对稳定,可以少关注一点。接下来还要利用 Pareto 图等其他的分析工具对 M4 和 M5 模块中的缺陷类型,严重程度等进行详细分析,便于更好的提出改进方案,这里不再赘述。

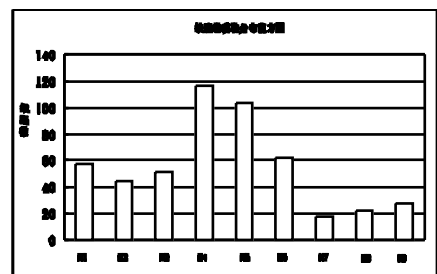


图2 缺陷的模块分布

### 3.4 评价度量

从计划度量到实施度量, 得出缺陷度量的结果, 在进行过程改进之前还应该对度量过程进行评价, 确保整个过程的正确性, 得出的结果是否和既定的信息需要相符, 以及是否科学合理的。利用 CMMI 模型所提供的 SCAMPI 评估标准, 对实施度量所产生的结果进行过程评估, 并将最终评估结果提交给项目组负责人, 从而便于管理层进行软件开发过程中的关键决策。

每一次度量都会积累很多经验教训, 经验将产生更好的度量与分析的方法。度量评价将确保项目度量方法的持续更新, 以满足当前的信息需要并促进企业度量过程的成熟度的提高。评价度量活动包括三个任务:

评价度量——考虑度量构造、基本度量、派生度量和指示器是否满足项目的信息需要<sup>[4]</sup>;

评价度量实施过程——检查项目度量过程的实施, 包括度量过程的能力, 度量过程与度量计划的符合性, 相对于预期的偏差;

评价结果——得出对度量过程综合评价的结果, 并积累在度量及其评价过程中所获得的经验教训。

### 3.5 改进过程

根据度量的评估结果制定过程改进方案, 在制定方案时要充分考虑方案的可行性, 以及实施的成本和效益的平衡问题, 然后选择一个最优的解决方案。软件过程的改进, 是一个持续不断的过程, 过程改进既是度量的结果又是度量的动因, 改进后的效果如何还

要通过度量进行评价。这就回到了度量计划阶段, 重新制定度量目标开始新的度量与分析活动。

## 4 结论

软件缺陷度量是为了定量地对软件缺陷进行分析, 为软件过程改进提供反馈信息。CMMI 中非常重视量化管理, 企业要想达到更高的软件能力成熟度等级, 就必须有完善的软件缺陷管理。本文提出基于 CMMI 的软件缺陷度量模型, 从计划度量、实施度量、评价度量和改进过程四个方面论述缺陷度量, 明确度量与分析的具体步骤, 从而帮助企业解决在实施软件缺陷度量时面临的问题。接下来的研究当中还要进一步完善度量指标体系, 使度量与分析的角度更加全面, 从而便于更好的进行缺陷管理与预防。

### 参考文献

- 1 Park RE, Goelert WB, Florac WA. Goal-Driven Software Measurement-A Guidebook. CMU/SEI-96-HB-002, 1996.
- 2 Pandian CR. Software Metrics: A Guide to Planning, Analysis, and Application. Auerbach Publications, 2004.45-46.
- 3 Fredericks M, Basili V. Using Defect Tracking and Analysis to Improve Software Quality. IBM Journal of Research and Development, 1998,19 (10):23-26.
- 4 任发科.软件开发过程中实用软件度量的研究与支持环境的实现.北京:北京航空航天大学, 2004.