

基于 ODC 的软件缺陷度量研究

陈爱真, 曾福萍, 陆民燕

(北京航空航天大学 工程系统工程系, 北京 100191)

摘要: 从正交缺陷分类(orthogonal defect classification, ODC)出发, 介绍在缺陷度量前需要收集的缺陷数据信息, 阐述了缺陷属性的具体分类, 然后从单维度和多维度两个角度介绍了如何利用 ODC 的缺陷属性进行度量分析, 并给出了软件组织应用 ODC 的流程, 最后提供了正交缺陷分类方法的应用实例, 为缺陷度量的应用研究提供了一种思路。

关键词: 软件度量; 缺陷度量; 正交缺陷分类

中图分类号: TP311 **文献标志码:** A **文章编号:** 1001-3695(2010)02-0563-03

doi:10.3969/j.issn.1001-3695.2010.02.044

Software defect metric based on orthogonal defect classification

CHEN Ai-zhen, ZENG Fu-ping, LU Min-yan

(Dept. of System Engineering of Engineering Technology, Beihang University, Beijing 100191, China)

Abstract: Firstly, according to the orthogonal defect classification, this paper described the defect data information which should be collected before defect measurement execution. Secondly, it described how to use the orthogonal defect classification for measurement analysis from the views of single dimension and multi-dimensional. Thirdly, presented the application process in the software organization. Finally, elaborated the application example. The study of this paper offered a new thought to the application of defect measurement.

Key words: software metric; defect metric; orthogonal defect classification(ODC)

如何评价软件产品的质量以及更好地对软件过程进行管理和改进是长期困扰软件业界的问题。近年来, 软件界许多专家提出了以软件缺陷数据为核心的观点^[1]。在软件开发和测试过程中贯穿着缺陷的引入、发现、修复和关闭的过程, 如何充分利用这些缺陷信息进行软件过程改进和产品质量的评估是当前研究的热点。软件缺陷度量是软件组织对软件的质量和过程进行评估和预测的常用手段之一。定量的统计缺陷模型和定性的缺陷根原因分析是两种传统的缺陷度量方法。前者用数学的方法抽象和统计缺陷的数量, 在开发过程的后期得到一些相关的分析报告。后者用自然语言描述单个缺陷的根本原因和表现形式, 是对单个缺陷的精确分析。显然, 定量和定性的方法是软件缺陷度量的两个极端, 它们无法实现对缺陷信息的全面分析, 以及将度量的结果及时反馈给开发人员。

正交缺陷分类(ODC)技术是 IBM TJ Watson 研究中心的 Chillarege^[2]在 1992 年提出的概念。该方法是一种介于定性和定量之间的分析方法, 它在统计缺陷模型和缺陷根原因分析之间架起了一座桥梁^[2]。

目前业界已有一些关于 ODC 的研究, 如文献[3,4]分别介绍了缺陷类型属性和缺陷触发属性的研究与应用; 文献[5]介绍了 ODC 在电信行业的应用; 文献[6]介绍了缺陷触发在测试过程中应用的一个案例。上述关于 ODC 的研究已有一些成果, 但也存在一些不足: 对 ODC 的研究更多地集中在理论上的研究, 针对软件组织如何应用 ODC 进行缺陷度量的研究较少, 这样就导致了软件组织在掌握 ODC 的基本知识后不知道如何应用于本企业。因此, 本文将从 ODC 缺陷分类、ODC 缺陷度量

以及 ODC 的应用流程进行较为系统的分析, 从而使软件组织更容易地应用 ODC 进行缺陷度量。

1 正交缺陷分类方法

1.1 ODC 缺陷分类

ODC 在高层次上, 是帮助获取缺陷信息的一种缺陷分类方法, 它所定义的缺陷属性是 IBM 公司在总结了大量的项目经验后抽象得到的。ODC 的特征属性^[7,8]包括发现缺陷的活动、缺陷的影响、缺陷触发、缺陷的修复对象、缺陷类型、缺陷限定词、缺陷来源、缺陷历史。在应用 ODC 进行缺陷数据收集时, 分为发现缺陷和修复缺陷两个阶段进行, 如图 1 所示。

a) 当测试人员或用户发现一个缺陷时需要定义以下三个属性:

(a) 活动属性, 指发现缺陷的活动, 即发现缺陷的测试类型, 如设计检查、代码审查、单元测试、功能测试和系统测试等。

(b) 触发属性, 指促使缺陷被发现的环境或条件。通常, 测试人员在发现缺陷时, 需要描述缺陷发现的条件和环境, 以利于缺陷的复现。Trigger 属性正是用来描述促使潜在缺陷显露出来成为一个可见的故障所必须具备的环境或条件。例如, 在软件代码审查中发现 C 语言中的相等判断符号“= =”, 被误写成了赋值符号“=”, 那么该缺陷的 trigger 属性应取值为 language dependency。其含义是在进行与编程语言相关的检查时发现缺陷。由于不同的测试活动发现软件缺陷的方法或手段是不相同的, 可以建立活动属性和触发属性之间的映射关

收稿日期: 2009-05-20; 修回日期: 2007-07-02

作者简介: 陈爱真(1983-), 女, 福建泉州人, 硕士研究生, 主要研究方向为软件缺陷度量、软件测试等(buaacaz@gmail.com); 曾福萍(1977-), 女, 江西人, 讲师, 硕士, 主要研究方向为软件度量、软件测试; 陆民燕(1963-), 北京人, 女, 教授, 博士, 主要研究方向为软件可靠性测试技术、度量和管理的。

系,如表 1 所示。

表 1 Trigger 与 activity 的映射关系

活动	设计检查和代码审查	单元测试	功能测试	系统测试
触发	设计一致性	简单路径	覆盖率	工作量
	逻辑/流程			压力
	向后兼容	复杂路径	变更	恢复
	横向兼容			例外启动
	并发性	交互作用	时序	重启
	内部文档			硬件配置
	语言依赖性	极少出现的情况		软件配置
边缘效应				

(c)影响属性,指假定或者实际的对用户的影响。如果缺陷是测试人员在测试过程中发现的,则由测试人员假定或推测判断该缺陷如果未被发现而逃逸到用户使用现场将对用户造成的影响;如果缺陷是用户在实际使用中发现的,则由用户定义缺陷对用户实际产生的影响。该属性可划分为 13 个子属性,包括安装性、完整性/安全性、文档、需求、服务性、标准、移植性、可靠性、性能、维护性、可用性、能力、易用性。

b)当开发人员接到一个缺陷,并且开始修复缺陷时,需要定义以下属性:

(a)修复对象属性,指修复缺陷的最高层的描述,即为了修改这个缺陷,需要追溯到哪个软件阶段、哪些软件产品。例如,缺陷描述为:数据库记录的数据太多将会影响到软件的性能。通过分析为了修复该缺陷需要更改设计,则该缺陷的 Target 属性为设计。target 属性可以划分为六个子属性,分别为需求、设计、代码、构造/打包、开发信息、国际语言支持。

(b)缺陷类型属性,指需要采取什么解决方案修复缺陷。例如,缺陷是由于没有对变量初始化而造成的,那么缺陷类型属性就应定义为初始化。由于缺陷修复的目标不同,其修改的方案也是不一样的。针对修复对象为设计和代码时,缺陷类型的划分包括七个子属性,分别为赋值/初始化、检查、算法/方法、功能/类/对象、时间/序列、接口/面向对象消息和关系。

(c)限定词属性,指缺陷的性质,是遗漏的、错误的还是不相关的。该属性是对缺陷类型的补充说明。在度量应用中可利用该属性来判断软件产品代码是否完整。

(d)来源属性,指缺陷引入的最根本的来源,即缺陷是通过什么途径引入的。可能的引入来源包括内部开发、重用库、外购和移植。

(e)历史属性,指缺陷的历史,即该缺陷是以前版本中存在的新代码、修复缺陷时引入的,还是重写代码时引入的。

1.2 ODC 缺陷度量

缺陷分类的目的不仅是为了对缺陷进行更好的管理,更重要的在于可以利用缺陷信息进行缺陷度量分析。缺陷度量是为了定量地对缺陷进行分析,提供有关软件过程的反馈信息^[9,10]。因此,“只收集数据,不进行分析”是错误的做法,收集数据并不是最终的目的,只有通过缺陷数据进行度量分析,才能够掌握软件产品的质量情况,及早发现开发过程中存在的问题,以及度量测试的有效性等。

ODC 不仅仅是一个缺陷分类方法,它还是建立在其定义的缺陷语义信息基础上的度量方法。文献[11]对 ODC 在度量方面的应用进行了研究,并给出了 ODC 度量的一些常见模式,如表 2 所示,可以为企业在进行缺陷度量时提供参考。

表 2 正交缺陷分析的一些常见模式

度量目标	ODC 属性	非 ODC 属性
评估产品的稳定性	缺陷类型,影响,限定词,来源,缺陷历史	发现缺陷时间,严重程度,修复缺陷时间,模块
度量测试的有效性	活动,触发,限定词	发现缺陷时间,严重程度,模块,阶段
鉴别设计和代码中的强处和不足	缺陷类型,限定词	模块
度量客户的使用方法	影响,触发,缺陷类型,限定词	发现缺陷时间,严重程度,模块
度量进度	活动,触发	严重程度,模块,阶段

2 ODC 缺陷度量方法研究

本文将在充分理解 ODC 缺陷分类方法的基础上,从单维度和多维度两个角度对 ODC 缺陷度量方法进行研究。

1)单维度分析 即单独对一个缺陷属性进行度量分析。由于 ODC 定义了八大属性,可以分别对八大属性进行单维度分析。应用 ODC 进行单维度分析的步骤如下:

a)软件组织可以根据以往软件项目的历史数据确定软件过程中发现的缺陷属性分布情况,从而确定八大缺陷属性的期望分布。

b)针对每个缺陷属性进行分析时,把当前项目发现的缺陷属性分布情况与期望分布的缺陷特征分布进行比较,观察是否存在异常分布趋势或异常特征点。

c)针对每个缺陷属性存在的异常情况,分析该异常产生的原因,确定软件产品或软件过程存在的问题,并采取具体的改进措施。

2)多维度分析 即结合两个或两个以上的缺陷属性进行分析。例如,在评估产品的稳定性时,可以结合缺陷类型和限定词进行分析,如图 2 所示。

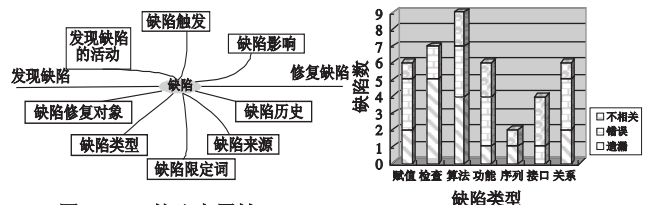


图 1 ODC 的八大属性

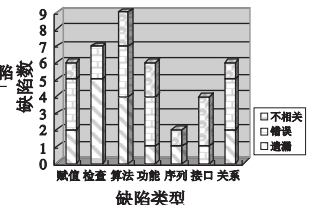


图 2 缺陷类型与缺陷限定词组合分析

在发现某一缺陷类型出现异常时,可以同时得到该缺陷类型的限定词信息,即是遗漏的、错误的还是不相关的缺陷占的比重较大。因此,可以提供更丰富的信息以确定软件产品和软件过程的问题。

一般而言,缺陷发现的活动可以与其他属性结合在一起分析,以掌握不同活动发现的缺陷其他属性分布情况。但是,其他的缺陷属性之间并不是都可以随便组合的,应该以企业自身度量的目的为驱动,有针对性地选取两个或两个以上的缺陷属性进行分析,这样得到的结果才会更有指导意义。如果软件组织不知道如何选取缺陷属性进行组合,那么可以参考表 2。

3 ODC 缺陷度量流程

虽然 IBM 公司已经给出一套完备的缺陷分类属性,包括八大关键属性和 164 个子属性,但是不同的软件组织的缺陷有着不同的特征。软件组织在应用 ODC 时,需要根据软件组织的实际情况借鉴其相似的缺陷属性,而对于差异较大的缺陷属性需要根据实际情况在深刻理解 ODC 八大属性的基础上对其

进行改造。在缺陷度量时,可以结合软件组织的度量目的,参考表 2 和软件组织实际情况综合分析选择缺陷属性数据。基于 ODC 的缺陷度量应用流程(图 3)可以归纳为下面的几个步骤:

- a) 准备条件:深刻理解 ODC 的八大属性,了解软件组织的缺陷特征及缺陷分类;
- b) ODC 改造:基于对 ODC 八大属性的深刻理解,依据软件组织自身的缺陷特点对 ODC 的子属性进行改造;
- c) 收集缺陷数据:在度量目的的指导下,选择适当的度量,应用改造后的 ODC 进行缺陷数据的收集;
- d) 实施缺陷度量:应用改造后的 ODC 模型,根据度量的目的选择度量,并收集缺陷数据,进行缺陷度量。
- e) 分析数据:对度量中的异常数据进行分析。
- f) 制定改进措施:根据异常数据分析的结果结合具体的实际情况制定改进措施。

4 ODC 缺陷度量应用

4.1 ODC 改造

上面介绍的 ODC 缺陷度量应用流程,为软件组织应用 ODC 进行缺陷度量提供了一种思路。本文将 ODC 应用于某软件测试组织收集的缺陷数据,并根据软件测试组织的自身特点对 ODC 进行改造,得到的改造结果如表 3 所示。

表 3 某软件组织改造 ODC 结果

ODC 属性		ODC 子属性		
活动	文档审查	代码审查	单元测试	系统测试
触发	种类、标志、内容、格式	逻辑、数据、接口、注释、文档、可追溯性、例外情况处理、内存	复杂路径、简单路径	功能、性能、接口、边界、强度、余量、安全性、恢复性、人机交互界面、可靠性、安装性、容量、互操作性、敏感性、数据处理
影响类型	功能、安装性、可移植性、效率、可维护性、可靠性			
限定词	需求、设计、代码、其他			
修复对象	文档、代码			
来源	开发、测试			
历史	基础、修复			

4.2 缺陷度量

本节将在上述缺陷分类的基础上,从单维度和多维度两方面对缺陷信息进行度量分析。通过收集六个软件项目的缺陷数据来确定其缺陷属性的期望分布。

1) 单维度分析 通过对六个软件项目在不同测试活动中发现的缺陷数进行统计分析,可得到如图 4 所示的期望分布情况。

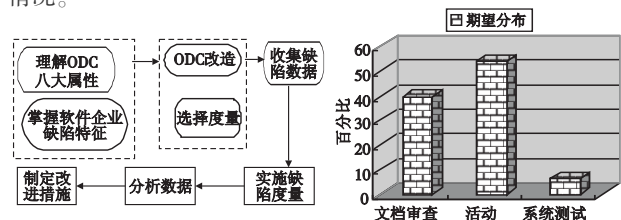


图 3 基于 ODC 的缺陷度量应用流程

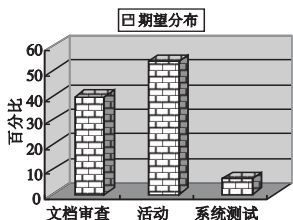


图 4 测试活动缺陷期望分布图

依据期望分布情况,对某直升机的昼夜观瞄装置管理软件进行度量分析。通过采集其不同测试活动发现的缺陷数,并将其与该期望分布情况进行比较,如图 5 所示。

由图 5 可以看出,该软件项目在代码审查和系统测试活

动中发现的缺陷数较多,即该软件在代码审查和系统测试活动的测试较为充分。文档审查活动发现的缺陷较少的原因可能是软件组织的文档化管理比较完善。通过了解软件组织的情况,发现该软件组织的文档都已经按照标准制定了各种文档的模板,包括文档需要介绍的内容、格式等都进行了明确的规定。

2) 多维度分析 在进行多维度分析时,可以同时考虑两个及两个以上的缺陷属性。本文结合测试活动和缺陷触发两个属性进行分析,以期了解某一测试活动的缺陷触发分布情况。如图 6 所示,给出了某一测试项目文档审查阶段的缺陷触发与期望分布的比较情况。

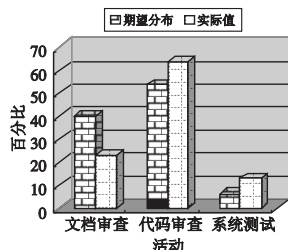


图 5 某一软件项目活动属性的单维度分析

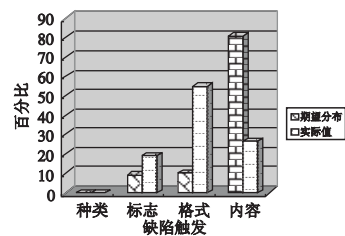


图 6 文档审查活动的缺陷触发属性分析

由图 6 可以看出,该软件项目在文档审查阶段,格式触发发现的缺陷比期望分布多,因此,需要与开发人员确认软件的文档是否按照软件工程化的格式要求进行编写,并及时提醒开发人员应注意文档的格式,以提高文档的可读性和可维护性。同样,该软件项目中内容触发发现的缺陷比期望分布少,因此,需要了解测试过程中对文档内容的审查是否充分。在资源允许的情况下,测试人员可以对该软件项目的内容进行更全面的审查。

5 结束语

软件缺陷度量的目的在于充分利用软件开发和测试过程中发现的缺陷数据,挖掘其有用的信息,为评价软件产品质量,以及制定测试和开发过程的改进决策提供数据基础。本文阐述了基于 ODC 进行缺陷度量的应用。但是,如何利用缺陷度量的结果制定更为合理的改进措施,将作为以后研究工作的重点。

参考文献:

- [1] 陈莉. 正交缺陷分类方法在软件缺陷管理及分析中的应用[D]. 长沙:湖南大学,2005.
- [2] CHILLAREGE R. Orthogonal defect classification: a concept for in-process measurements [J]. IEEE Trans on Software Engineering, 1992, 18(11): 943-956.
- [3] 袁东林. 基于正交缺陷分类的软件过程测量方法[J]. 计算机应用与软件, 2007, 24(3): 65-38, 73.
- [4] 袁东林. 利用正交缺陷分类技术测量软件验证过程的有效性[J]. 计算机应用研究, 2006, 23(4): 81-84.
- [5] 潘铁军, 郑蕾娜. 基于正交分类的软件缺陷跟踪系统[J]. 计算机工程与应用, 2007, 43(13): 76-79, 101.
- [6] CHILAREGE R. Test and development process retrospective: a case study using ODC triggers [C]//Proc of the International Conference on Dependable Systems and Networks. 2002.
- [7] CHILLAREGE R. ODC for process measurement, analysis and control [C]//Proc of the 4th International Conference on Software Quality. 1994: 10-15.
- [8] ODC official website [EB/OL]. (2002). <http://www.research.ibm.com/softeng/ODC/ODC.HTM>.
- [9] United Kingdom Software Metrics Association. Quality standards defect measurement manual [S]. 2000
- [10] 胡冠林, 汪厚祥. 软件缺陷分类及其度量技术研究[J]. 舰船电子工程, 2005, 25(3): 55-58.
- [11] BUTCHER M, MUNRO H, KRATSCHEMER T. Improving software testing via ODC: three case studies [J]. IBM Systems Journal, 2002, 41(1): 31.