

软件缺陷模式的研究

曾福萍 靳慧亮 陆民燕

(北京航空航天大学工程系统工程系 北京 100191)

摘要 软件缺陷是导致软件不可靠的根本原因,提高软件可靠性的关键在于减少软件缺陷,那么如何利用积累的缺陷数据提高软件可靠性?结合软件缺陷和模式的概念提出了软件缺陷模式的定义。通过分析积累的软件缺陷数据对缺陷模式的所属分类进行了划分,在此基础上进一步给出了软件需求分析、设计和编码各阶段的软件缺陷模式。最后阐述了在软件开发过程和测试过程中缺陷模式的应用,为如何利用缺陷数据来提高软件可靠性提供了思路。

关键词 软件缺陷,模式,缺陷模式,缺陷模式库

中图分类号 TP311 **文献标识码** A

Study on Software Defect Patterns

ZENG Fu-ping JIN Hui-liang LU Min-yan

(Department of System Engineering of Engineering Technology, Beijing University of Aeronautics and Astronautics, Beijing 100191, China)

Abstract Software defects are the root causes that lead software unreliable, thus the reduction of software defects is the key way to improve software reliability. How to improve the software reliability using the defect data accumulated? In this paper, the definition of software defect pattern was addressed on the basis of the concept of software defect and pattern. Through analyzing the defect data accumulated, the classification of defect patterns was proposed. The defect patterns in software requirement analysis, design and coding phase were advanced. How to apply the defect patterns is elaborated in the process of software development and testing, which provides idea on how to improve software reliability utilizing defect data.

Keywords Software defect, Pattern, Defect pattern, Defect pattern database

如何保证软件产品的可靠性,一直都是任何软件项目致力解决的问题^[1]。影响软件可靠性的因素很多,而软件缺陷是度量软件可靠性的一个重要的指标,减少软件缺陷才能保证软件可靠性^[2]。目前,减少软件缺陷主要有两种途径:一是在软件开发过程中尽量减少缺陷的引入;二是通过软件测试发现并改正缺陷。这两种活动必然要依赖于开发人员和测试人员的技术和经验。可是,现有的资深人数还远远不能满足需求。

随着软件开发和测试项目的不断深入,积累了越来越多的软件缺陷数据,如何利用这些积累的缺陷数据提高软件可靠性是人们一直在思考的问题。

软件缺陷的产生因人而异、因语言而异、因问题而异,总结已有软件缺陷数据可以发现,相当数量的软件缺陷是重复的,某些缺陷表现形式相同,某些缺陷产生原因相同,即软件缺陷本身及其产生遵循一定的规律。这种规律能够提供很多可以归入缺陷模式的具体范例。如果能结合已有的缺陷数据,总结这种缺陷模式,给软件开发人员、测试人员提供参考,减少经验的限制,在一定程度上就可以解决上述问题。也就是说,通过研究这些软件缺陷模式,测试人员可以在测试过程

中更快地识别缺陷并进行改正;开发人员也可以在开发过程中考虑采用什么样的开发技术预防这些缺陷模式的再次出现,从而提高软件开发和测试团队的整体水平。因此,对软件缺陷模式的研究越来越重要。目前已有一些人对软件缺陷模式开展了研究,并且已经取得了一定的进展,如文献[9]给出了软件缺陷模式的定义及一些C语言的缺陷模式;文献[10]介绍的“Bug Patterns in Java”针对java语言提出了Bug Pattern的定义及相应的缺陷模式;文献[11]提出了基于缺陷模式的测试方法及测试系统,但未显示地给出软件缺陷模式的概念。

上述对缺陷模式的研究主要集中在3个方面:软件缺陷模式的定义、软件缺陷模式的整理和基于缺陷模式测试自动系统的研究。但也存在一些不足:对缺陷模式的整理主要集中在代码级,没有延续到软件开发的其它阶段如需求分析和设计;对软件缺陷模式的定义多样,没有可以遵循的统一的概念;而且研究内容较为单一,对其进行系统的研究很少。基于此,本文欲从软件缺陷模式的概念、各阶段缺陷模式的整理以及缺陷模式的应用等方面进行较为系统的剖析,从而使人们对软件缺陷模式有一个较为全面的认识。

到稿日期:2010-03-09 返修日期:2010-06-23 本文受总装十一五预研(513190702)资助。

曾福萍(1977-),女,博士,讲师,主要研究方向为软件可靠性工程、软件测试,E-mail:zfp@buaa.edu.cn;靳慧亮(1984-),男,硕士,主要研究方向为软件测试、软件度量;陆民燕(1963-),女,教授,博士生导师,主要研究方向为软件可靠性测试技术、软件可靠性度量、软件可靠性设计分析、软件可靠性设计管理等。

本文首先在对软件缺陷和模式概念研究的基础上,提出了软件缺陷模式的概念;然后在收集的缺陷数据基础上,对缺陷模式的所属分类进行了划分,进一步给出了软件需求、设计和编码阶段的缺陷模式;最后说明了如何应用缺陷模式指导开发和测试工作,最终提高软件的可靠性。

1 软件缺陷模式概念

通过查找相关技术书籍,翻阅参考大量国内外参考文献,对目前软件缺陷和模式理论研究的发展情况和具体应用进行了深入了解:软件缺陷模式的研究始于近几年,尚没有权威组织提出软件缺陷模式的定义。目前,已有的软件缺陷模式的定义只针对代码级的缺陷,没有涉及到其它阶段如需求、设计阶段等。本文欲将软件缺陷模式的概念扩展到适用于软件生命周期各阶段。

为了更好地剖析软件缺陷模式的概念,本节首先对软件缺陷和模式的概念进行了探讨,在此基础上给出了软件缺陷模式的概念。

1.1 软件缺陷概念

软件缺陷的定义繁多,以下介绍几个比较常用的软件缺陷的定义。

(1) 国标 GB/T11457-89“软件工程术语”是等同采用 IEEE STD729-1983 制定的,IEEE 软件工程术语^[4]中对软件缺陷(software defect)的定义为:1)从产品内部看,软件缺陷是软件产品开发或维护过程中所存在的错误、偏差等各种问题;2)从外部看,软件缺陷是系统所需要实现的某种功能的失效或违背。

(2) SW-CMM 是这样定义软件缺陷的:“系统或系统成分中的能造成它们无法实现其被要求的功能的缺点。如果在执行过程中遇到缺陷,它可能导致系统的失效”^[5]。

(3) 在 IEEE 的软件可信性度量的标准词汇^[6](IEEE 982.1-2005 Standard Dictionary of Measures of the Software Aspects of Dependability)中对软件缺陷的定义为:一个通用术语,涉及软件故障原因和失效影响。

(4) 国内软件可靠性工程领域广泛使用的定义^[6]为:软件缺陷是存在于软件中的、不期望的或不可接受的偏差,其结果是当软件运行于某一特定条件时将出现软件故障(即,软件缺陷被激活),软件缺陷以一种静态的形式存在于软件的内部,是软件开发过程中人为错误的结果。软件缺陷的例子有:数组下标不对、循环变量初值设置有误、异常处理方法有误等。

本文采用软件可靠性工程领域广泛使用的定义,明确区分错误、软件缺陷、软件故障和软件失效的概念。这几个概念及联系如图 1 所示。

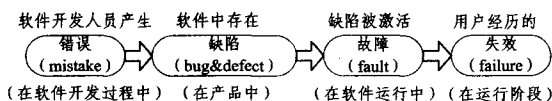


图 1 软件错误、缺陷、故障、失效的概念及联系

由此可以看出,软件作为思维的产物不可避免地受到开发人员自身、所使用的编程语言特点以及软件运行环境等多方面的影响。但是,由于人所具有的思维定势以及编程语言自身的特性,使得软件缺陷的出现具有一定的统计规律。

1.2 模式的概念

模式这一概念最早出现在城市建筑领域,Christopher

Alexander 的一本关于建筑的书明确给出了模式的概念^[12],他说:“每一个模式描述了一个在我们周围不断重复发生的问题,以及该问题的解决方案的核心,这样你就能一次又一次地使用该方案而不必做重复劳动”,他使用模式这一概念来解决建筑中的一些问题,随后,模式在构造复杂系统时的重要性逐渐在其他领域中得到认可。

模式是针对复杂系统中重复出现的问题而提出来的。有经验的专家在解决问题时,通常先考虑以前解决过的相似问题,并重用其解法的精华来解决问题,这个不断被引用的解法就是通常说的模式。

1.3 软件缺陷模式的概念

通过研究软件缺陷和模式的概念,可知:

(1) 缺陷存在于软件生命周期的各个阶段。各阶段由于执行者对上一阶段的认知不充分而导致本阶段的软件实现与上一层的设计意图不相符或各阶段使用方法有误,即产生了缺陷。由于人所具有的思维定势以及编程语言自身的特性,使得软件缺陷的出现具有一定的统计规律。

(2) 从本质上讲,模式是一个抽象概念,是从不断重复出现的事件中发现和抽象出的规律,总结为解决问题的经验。重复出现的事物就可能存在某种模式。

据此,本文将软件缺陷模式定义为:

软件缺陷模式是对发生的不断重复的或类似的软件缺陷中发现和抽象出的规律描述。

此定义适用于软件生命周期中任何阶段引入的软件缺陷。

2 软件缺陷模式

明确了软件缺陷模式的定义之后,在软件缺陷数据的基础上就可以抽取归纳相应的缺陷模式。

2.1 软件缺陷模式的获取

软件缺陷模式的获取前提是要收集相应的缺陷数据。本文的软件缺陷数据来源及缺陷模式获取过程主要有 3 个方面:

(1) 多年来软件开发和测试所积累的丰富的缺陷数据。靠人工的方法收集和整理大量的缺陷数据,从中剔除重复的缺陷数据,提取出有代表性的数据作为研究软件缺陷模式的基础。

(2) 各种文献资料中归纳国内外项目实践积累的缺陷数据,用于补充缺陷模式。

(3) 领域专家的经验数据。这里领域专家主要泛指软件开发和测试人员等。将通过项目和资料收集整理出的软件缺陷模式交给专家辨别其正确性,同时专家根据自己的经验补充缺陷模式。

因此,软件缺陷模式的获取如图 2 所示。

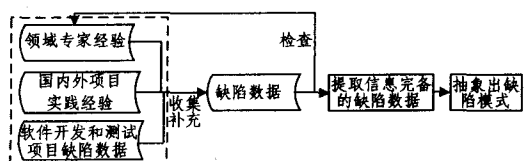


图 2 软件缺陷模式获取

2.2 软件缺陷模式所属分类

为了更好地组织软件缺陷模式,便于为开发和测试人员

提供参考,且具有更好的可操作性,对缺陷模式进行了所属分类划分。在缺陷模式所属分类中引入了“层次原理”,即分层思想。

一般将缺陷模式所属分类划分成3层:首先缺陷模式按开发阶段分为软件需求缺陷模式、设计缺陷模式和编码缺陷模式。然后软件需求设计缺陷模式按照质量要求及工作内容进一步细化,软件编码缺陷模式进一步细化成通用和C语言缺陷模式,C语言缺陷模式又从文件的结构分为头文件引用、顺序结构、if语句、switch语句、循环语句、函数定义等缺陷模式。

缺陷模式所属分类详如表1所列。

表1 缺陷模式所属分类

所属分类1	所属分类2	所属分类3
软件需求	完整性要求	功能需求
		性能需求
		接口需求
		数据需求
一致性要求	可验证性要求	异常需求
		其它要求
		其它要求
		其它要求
软件设计	容错设计	耦合和内聚设计
		模块的独立性
		模块的入口与出口
		模块的规模
软件设计	共享资源设计	模块的深度、宽度、扇出和扇入
		模块的接口
		McCabe指数
		数据的合理性检查
软件设计	异常设计	异常情况设计
		异常情况设计
		异常情况设计
		异常情况设计
软件设计	用户界面设计	结构容错
		时间容错
		信息容错
		信息容错
软件编码	C语言程序设计	中断使用的原则
		中断的初始化、打开和使能的注意事项
		中断的嵌套
		中断的优先级
软件编码	C语言程序设计	安全关键功能设计
		安全关键信息设计
		界面设计方法
		界面设计样式
软件编码	C语言程序设计	界面容错
		界面输出显示
		界面输出显示
		界面输出显示
软件编码	C语言程序设计	源程序文档化
		数据说明风格
		语句构造风格
		头文件结构
软件编码	C语言程序设计	顺序结构
		内存
		IF语句
		SWITCH语句
软件编码	C语言程序设计	循环语句
		函数
		函数
		函数

2.3 软件需求设计编码缺陷模式

通过对缺陷数据的抽象共整理出124条软件缺陷模式。限于篇幅,在此不能一一介绍这些缺陷模式,分阶段举例说明,软件需求阶段的缺陷模式示例如表2所列,软件设计阶段的缺陷模式示例如表3所列,软件编码阶段的缺陷模式示例如表4所列。

表2 软件需求阶段缺陷模式示例

所属分类2	所属分类3	软件缺陷模式
完整性要求	功能需求	1. 缺少对软件功能的输入信息的描述
		2. 软件功能的输入描述不完整
		3. 缺少对软件功能的输出信息的描述
		4. 软件功能的输出描述不完整
		5. 缺少对软件功能处理流程的描述
		6. 软件功能的处理流程描述不完整
异常需求	异常需求	1. 缺少对功能输入不符合要求时的处理流程
		2. 对功能输入不符合要求时的处理流程描述不完整
		3. 缺少对异常情况的判断

表3 软件设计阶段缺陷模式示例

所属分类2	所属分类3	软件缺陷模式
简化设计	模块的入口与出口	模块有多个入口和多个出口
	模块的规模	1. 小型软件的模块语句数大于60 2. 中型软件的模块语句数大于150 3. 大型软件的模块语句数大于250
简化设计	模块的深度、宽度、扇出和扇入	1. 模块的扇入大于7 2. 模块的扇出大于7 3. 底层模块有较高的扇出,高层模块有较高的扇入
	模块的接口	模块间传递的参数个数太多
简化设计	McCabe指数	McCabe大于10

表4 软件编码阶段缺陷模式示例

所属分类2	所属分类3	软件缺陷模式
程序设计风格	语句构造风格	1. 多个语句写在同一行
		2. 对条件“非”进行判断
		3. 未使用括号表明表达式的运算次序
		4. 使用GOTO语句
C语言程序设计	IF语句	1. 将布尔变量直接与TRUE, FALSE 或者1,0进行比较
		2. 将浮点变量用“==”或“!=”与数字比较
		3. 在判断语句中,对其他变量进行计算或赋值
		4. 遗漏某些判断条件

3 软件缺陷模式的应用

将软件缺陷数据抽象成缺陷模式的目的是为了能够更好地提高软件的可靠性,所以可以从如何避免引入类似缺陷和如何发现类似缺陷两个角度考虑缺陷模式的应用,即软件开发过程和软件测试过程中如何应用缺陷模式。

3.1 软件测试过程的缺陷模式应用

表5 软件需求测试检查单示例

项目名称		审查组长		
审查人		审查时间		
序号	审查内容	是	否	NA
1	功能完整性要求	对软件功能的描述是否缺少输入信息的描述,或缺少对输入信息的完整描述,如来源、意义、格式、接收方法、数量、输入范围及换算方法等		
		对软件功能的描述是否缺少对输出信息的完整描述,如信息的传送方法、意义、格式、数量、输出范围等		
2	异常处理的完整性要求	软件功能的处理流程描述是否完整		
		是否缺少对功能输入不符合要求时的处理流程		
检查情况说明		是否缺少对异常情况的判断		

缺陷模式可以指导测试人员在测试过程中更快地发现和识别缺陷。针对每条缺陷模式都制定了相应的测试检查单,

这些检查单可以用于指导软件不同阶段不同性质的测试工作,提高了发现缺陷的效率。表5给出了根据需求缺陷模式制定的测试检查单示例。

根据缺陷模式所制定的测试检查单已成功地指导了多个测评项目的文档审查、代码审查及系统测试的用例设计,受到了测试人员的青睐和肯定。

3.2 软件开发过程的缺陷模式应用

缺陷模式可以指导开发人员在软件开发过程中避免引入类似缺陷,即开发人员可以在开发过程中考虑采用什么样的开发技术预防这些缺陷模式的再次出现,将软件缺陷模式转换为软件设计准则。如何将软件缺陷模式转化为软件设计准则呢?通过对软件缺陷的因果分析及软件缺陷预防措施的研究可知:软件设计准则在某种程度上就是软件缺陷的预防措施。为此,需对缺陷模式作进一步的分析,本文引入软件缺陷模式库的概念,将缺陷模式库定义为四元组,由软件缺陷模式属性的集合构成,属性包括软件缺陷模式的名称、软件缺陷引入的原因及后果、软件缺陷预防措施,缺陷模式库定义如下:

$$\psi = (N, I, O, P)$$

式中, N 是软件缺陷模式的名称; I 是该缺陷模式引入原因的集合; O 是该缺陷模式引发后果的集合; P 是该缺陷模式预防措施集合,均不能为空值。

建立软件缺陷模式库之后,分析缺陷模式预防措施就可以制定相应的软件设计准则。

表6和表7给出了缺陷模式库及相应设计准则的示例。

表6 需求缺陷模式库示例

缺陷模式	引入原因	引发后果	预防措施
缺少对异常情况的判断	1. 需求分析人员与用户沟通不充分;2. 需求分析人员对“应该做什么、不能做什么”考虑不全;3. 粗心导致漏掉某些需求	软件功能无法满足用户需求,软件需求存在缺陷	与用户充分沟通,既要考虑软件要做什么,也应考虑软件不能做什么,应该明确输入、输出存在哪些异常情况,对其处理措施应详细说明

表7 需求设计准则示例

缺陷模式	需求阶段软件设计准则
缺少对异常情况的判断	准则:必须仔细分析软件运行过程中各种可能的异常情况,处理过程应考虑相应的保护措施。特别当采用现成软件时,必须仔细分析原有的异常保护措施对于现有的软件需求是否足够且完全使用

根据软件缺陷模式制定的设计准则已应用于某光学系统控制软件和某测试过程管理软件的开发中。软件开发人员认为,软件设计准则正是他们开展软件开发实践工作所迫切需要的,便于开发人员间吸取经验、避免已经发生缺陷的再发生,同时能更快地提高开发人员的熟练程度。

结束语 本文收集了大量软件缺陷数据,对整个软件生命周期中引入缺陷数据进行了深入研究,发现软件缺陷本身及其产生遵循一定的规律,同时结合模式的概念,提出软件缺陷模式的概念。软件缺陷模式是对软件缺陷的抽象描述,可以清晰描述某类具有共同特征的软件缺陷。之后分别对软件需求阶段、设计及编码阶段的缺陷模式进行了所属分类划分,同时整理归纳了相应的缺陷模式。最后,从软件开发和测试两个方面阐述了如何应用缺陷模式,将缺陷模式的预防措施转换为设计准则指导软件开发,避免相似缺陷的再发生;同时也可以根据缺陷模式制定测试检查单,更快更准确地识别相似缺陷并改正,为如何利用缺陷数据提高软件可靠性提供了思路。

后续工作一方面应随着软件缺陷数据的不断积累从不同类型软件、不同编程语言等角度扩充并完善缺陷模式。另一方面需加强基于缺陷模式自动测试和验证系统的研究,切实有效地指导软件开发和测试工作。

参考文献

- [1] 阮廉,陆民燕,韩峰岩. 装备软件质量和可靠性管理[M]. 北京:国防工业出版社,2006
- [2] 韩卫岗,周红建,赵禄丰. 软件缺陷信息分析研究[J]. 计算机工程与设计,2008,7:3381-3383,3447
- [3] 聂林波,刘孟仁. 软件缺陷分类的研究[J]. 计算机应用研究,2004,7:84-86,98
- [4] IEEE Std 729-1983. Standard Glossary of Software Engineering Terminology[S]. IEEE,1990
- [5] Paulk M C. Capability Maturity-model SM for software [R]. Pittsburgh, Pennsylvania; Carnegie Mellon University,1993
- [6] IEEE 982. 1-2005 Standard Dictionary of Measures of the Software Aspects of Dependability[S]. IEEE,2005
- [7] 石柱,何新贵,武庄. 软件可靠性及其评估[J]. 计算机应用,2000,20(11):1-5
- [8] Frederieks M, Basili V. Using Defect Tracking and Analysis to Improve Software Quality[J]. IBM Journal of Research and Development,1998,19(10):23-26
- [9] 唐为明. 嵌入式软件缺陷模式知识库研究[D]. 北京:北京航空航天大学,2007
- [10] Allen, Eric. Bug Patterns in Java [M]. Springer-Verlag: New York Inc,2005
- [11] 宫云战,赵瑞莲,等. 软件测试教程[M]. 北京:机械工业出版社,2008
- [12] 熊节. 模式的乐趣[M]. 北京:清华大学出版社,2003
- [13] 刘海,郝克刚. 软件缺陷原因分析方法[J]. 计算机科学,2009,36(1):242-243,251

(上接第109页)

- [12] Hervé C, Duong H P, Vid P. Public traceability in traitor tracing scheme[C]//EUROCRYPT 2005. Berlin; Springer Press, 2005: 542-558
- [13] Tzeng W G, Tzeng Z J. A public-key traitor tracing scheme with revocation using dynamic shares[J]. Designs, Codes and Cryptography, 2005, 35(1): 47-61
- [14] Yevgeniy D, Nelly F. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack[C]//PKC2002. Ber-

lin; Springer Press, 2003: 100-115

- [15] Kim C H, Hwang Y H, Lee P J. An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack [C]//ASIACRYPT2003. Berlin; Springer Press, 2003: 359-373
- [16] Kim C H, Hwang Y H, Lee P J. TTS without revocation capability secure against CCA2[C]//ACISP2004. Berlin: Springer Press, 2004: 36-49
- [17] Naor M, Pinkas B. Oblivious transfer and polynomial evaluation [C]//Proc. of STOC'99. Atlanta; ACM Press, 1999: 245-254