

软件缺陷分类的研究

聂林波, 刘孟仁

(海军工程大学, 湖北 武汉 430033)

摘要: 软件缺陷分类是研究软件缺陷管理的基础。说明了软件缺陷的危害, 阐述了对软件缺陷分类的必要性, 考察了国内外关于软件缺陷错误分类的各种方法, 分析了各种分类法的优缺点, 提出了一个有利于提高软件质量和改进软件过程的分类方法, 指出了缺陷管理系统的基本功能要求并总结了对软件缺陷进行分类的意义。

关键词: 软件缺陷; 缺陷分类; 软件故障; 软件失效

中图法分类号: TP311.53

文献标识码: A

文章编号: 1001-3695(2004)06-0084-03

Research on Software Defects Classification

NIE Lin-bo, LIU Meng-ren

(Naval University of Engineering, Wuhan Hubei 430033, China)

Abstract: Software defects management is built on the base of classification of software defects. This paper explains the harm of software defects, expatiates the necessity of classifying software defects, reviews some classification of software defects home and abroad, analyses the merit and flaw of each classification, brings forward a means of classification which would benefit elevation of software and improvement of software process, finally indicates the basic functional requirements of management system of defects and the significance of classification of software defects.

Key words: Software Defect; Defect Classification; Software Fault; Software Failure

1 引言

软件缺陷是软件产品的固有成分。软件缺陷是软件“生来具有”的特征。不管是小程序还是大型软件系统, 无一例外地都存在缺陷, 这些软件缺陷有的容易表现出来, 有的隐藏很深难以发现; 有的对使用影响轻微, 有的会造成财产甚至生命的巨大损失。1991年海湾战争中, 美军使用爱国者导弹拦截伊拉克飞毛腿导弹中, 出现过几次拦截失败事件, 后经查明是软件计时系统的累积误差所致, 该软件缺陷使一个很小的系统时钟错误积累起来造成十几个小时的延迟, 致使跟踪系统准确度丧失, 最终导致了严重的后果^[3]。软件从最初的设计到最后的退出使用, 需要开发人员、用户和维护人员的大量智力劳动。为了保证软件正常运行, 必须对软件中存在的缺陷进行有效的管理。

2 软件缺陷概述

缺陷是软件产品期望属性的偏离。软件没有满足规范要求以及用户的期望, 造成用户使用的不便, 就是软件缺陷。SW-CMM对其定义是: “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^[4]”。

在软件领域还有几个与软件缺陷相关的术语: 软件故障和软件失效。软件故障是软件没有表现出人们所期待的正确的

结果。软件失效是指软件出现如下的状态: 功能部件执行其功能的能力的丧失; 系统或系统部件丧失了在规定限度内执行所要求功能的能力; 程序操作背离了程序需求^[8]。缺陷是这些故障和失效的源头。要想获得高质量的软件就要从消除软件缺陷着手。进行缺陷的预防、检测和消除工作。

不同的软件开发阶段会产生不同的缺陷。缺陷存在于软件生命周期的各个阶段。在问题定义阶段, 系统分析员对问题性质、问题规模和解决方案的考虑不周会引入缺陷, 这种缺陷在软件开发的前期不易察觉, 只有到了软件的接受测试阶段甚至投入使用后才暴露出来。对这类缺陷的预防是尤为关键的。在定义需求规范阶段, 规范定义不完善, 这是导致软件缺陷的最主要原因。在系统设计阶段, 错误的设计方案, 致使系统无法实现。在编码实现阶段, 错误地理解了算法, 导致代码错误。在测试阶段测试人员对缺陷出现条件判断错误, 修改了一个错误, 却引出了更多错误。在维护阶段, 修改了有缺陷的代码, 却又导致了先前正确的模块出现错误。

不同的软件缺陷会产生不同的后果。在时间、人力、物力允许的情况下, 应修复软件中存在的所有缺陷。但是, 这只是软件开发的一种理想情况。绝大多数软件项目都是要求按期完成的, 而且给予项目的资源有限(包括人力资源和软件生产资料), 只有对软件缺陷进行区别对待, 区分其严重等级和优先等级, 将有限的资源充分利用, 解决对软件产品质量最为关键的软件缺陷, 才能生产出用户满意的软件。另一方面, 软件缺陷的修复代价。在软件生命周期的各个阶段, 相同的软件缺陷的修复成本大不相同。随着时间的推移, 修复软件缺陷的费用

是呈几何级增加的。因此要尽可能早的发现缺陷并修复缺陷。

对软件缺陷进行分类,分析产生各类缺陷的软件过程原因,总结在开发软件过程中不同软件缺陷出现的频度,制定对应的软件过程管理和技术两方面的改进措施,是提高软件组织的生产能力和软件质量的重要手段^[4 5]。

3 传统的软件缺陷分类方法

软件缺陷的分类方法繁多。各种分类方法的目的不同,观察问题的角度和复杂程度也不一样。几个有代表性的软件分类方法如下:

(1)Putnam 等人^[9]提出的分类方法将软件缺陷分为六类:需求缺陷、设计缺陷、文档缺陷、算法缺陷、界面缺陷和性能缺陷。

国家军用标准 GJB-437 根据军用软件错误的来源将软件错误分为三类:①程序错误,运行程序与相应的文档不一致,而文档是正确的;②文档错误,运行程序与相应的文档不一致,而程序是正确的;③设计错误,虽然运行程序与相应的文档一致,但是存在设计缺陷,可能产生错误。

这两种分类方法可以分析软件缺陷的来源和出处,指明修复缺陷的努力方向,为软件开发过程各项活动的改进提供线索。分类简单是该分类方法的显著特点。因为分类方法简单,所以提供的缺陷相关信息对具体的缺陷修复工作的贡献有限。

(2)Thayer 软件错误分类方法^[3]是按错误性质分类,它利用测试人员在软件测试过程填写的问题报告 and 用户使用软件过程反馈的问题报告作为错误分类的信息。它包括 16 个类,在这 16 个类之下,还有 164 个子类。16 类有:①计算错误;②逻辑错误;③ I/O 错误;④数据加工错误;⑤操作系统和支持软件错误;⑥配置错误;⑦接口错误;⑧用户需求改变,这是指用户在使用软件后提出软件无法满足的新要求产生的错误;⑨预置数据库错误;⑩全局变量错误;⑪重复的错误;⑫文档错误;⑬需求实现错误,这是指软件偏离了需求说明产生的错误;⑭不明性质错误;⑮人员操作错误;⑯问题,这是指软件问题报告中提出的需要答复的问题。

该分类方法特别适用于指导开发人员的缺陷消除和软件改进工作。通过对错误进行分类统计,可以了解错误分布状况,对错误集中的位置重点加以改进。该方法分类详细,适用面广,当然分类也较为复杂。该分类方法没有考虑造成缺陷的过程原因,不适用于软件过程改进活动。

(3)缺陷正交分类 ODC(Orthogonal Defects Classification)^[10]是 IBM 公司提出的缺陷分类方法。该分类方法提供一个从缺陷中提取关键信息的测量范例,用于评价软件开发过程,提出正确的过程改进方案。

该分类方法用多个属性来描述缺陷特征。在 IBM ODC 最新版本里,缺陷特征包括以下八个属性:发现缺陷的活动、缺陷影响、缺陷引发事件、缺陷载体(Target)、缺陷年龄、缺陷来源、缺陷类型和缺陷限定词。ODC 对八个属性分别进行了分类。

其中缺陷类型被分为七大类:赋值、检验(Checking)、算法、时序、接口、功能、关联(Relationship)。

分类过程分两步进行。第一步,缺陷打开时,导致缺陷暴露的环境和缺陷对用户可能的影响是易见的,此时可以确定缺陷的三个属性:发现缺陷的活动、缺陷引发事件和缺陷影响。第二步,缺陷修复关闭时,可以确定缺陷的其余五个属性:缺陷载体、缺陷类型、缺陷限定词、缺陷年龄和缺陷来源。这八个属性对于缺陷的消除和预防起到关键作用。

该分类方法分类细致,适用于缺陷的定位、排除、缺陷原因分析和缺陷预防活动。缺陷特征提供的丰富信息为缺陷的消除、预防和软件过程的改进创造了条件。IBM 的研究机构制定出该分类方法已经十多年。IBM 一直在改进 ODC。近几年,业界开始研究并使用 ODC。ODC 的缺点在于分类复杂,难以把握分类标准,缺陷分析人员的主观意见会影响属性的确定。

(4)电气和电子工程师学会制定的软件异常分类标准^[11](IEEE Standard Classification for Anomalies 1044-1993)对软件异常进行了全面的分类。该标准描述了软件生命周期各个阶段发现的软件异常的处理过程。

分类过程由识别、调查、行动计划和实施处理四个步骤组成,每一步骤包括三项活动:记录、分类和确定影响。异常的描述数据称为支持数据项。分类编码由两个字母和三个数字组成。如果需要进一步的分类,可以添加小数。例如 RR324, IV321.1。RR 表示识别步骤,IV 表示调查步骤,AC 表示行动计划步骤,IM 表示确定影响活动,DP 表示实施处理步骤。分类过程的四个步骤都需要支持数据项。由于每个项目都有各自的支持数据项,该标准不强制规定支持数据项,但提供了各个步骤相关的建议支持数据项。强制分类建立通用的定义术语和概念,便于项目之间、商业环境之间、人员之间的交流沟通。可选分类提供对于特殊情况有用的额外的细节。在调查步骤,对实际原因、来源和类型进行了强制分类。其中调查步骤将异常类型分为逻辑问题、计算问题、接口/时序问题、数据处理问题、数据问题、文档问题、文档质量问题 and 强化问题(Enhancement),共八大类,下面又分为数量不等的小类。分类细致深入,准确说明了异常的类型。

该分类方法提供一个统一的方法对软件和文档中发现的异常进行详细的分类,并提供异常的相关数据项帮助异常的识别和异常的跟踪活动。IEEE 软件异常分类标准具有较高的权威性,可针对实际的软件项目进行裁剪,灵活度高,应用面广。不足之处是没有考虑软件工程的过程缺陷,并且分类过程复杂。但是该方法提供了丰富的缺陷信息。缺陷原因分析活动可以充分利用这些信息进行原因分析。

4 基于过程的缺陷分类法

高质量高可靠性大型实时软件系统的按期按预算开发存在很大的困难。承担这类软件开发的组织需要较高级别的软件能力成熟度。为了满足软件开发组织实施缺陷预防、改进软

件过程和提高软件能力成熟度的需要,必要的工作就是软件缺陷管理。软件缺陷管理又需要缺陷分类作为基础。因此需要制定一个针对软件缺陷管理的缺陷分类方法。分类的目的是分析软件缺陷产生的过程原因,改进软件过程,预防软件缺陷,改进软件质量,提高组织的软件能力成熟度。该分类方法应满足以下要求:准确地对发现的缺陷进行分类;分类之间应无重叠,分类体系应覆盖所有的缺陷类型;分类要与软件生命周期^[1,2]有机结合,从软件过程的角度对软件缺陷进行分类。

传统的软件缺陷分类方法主要是为了消除软件缺陷,提高软件质量或是为了评价软件的性能和可靠性。这些分类方法不完全适用软件缺陷管理的需要。因此需要设计一个新的缺陷分类方法,在软件生命周期各个阶段中按照缺陷产生的所在过程来分类。

软件生产是以过程为主线的,各种活动都围绕过程进行。各种工具和方法的使用都和过程紧密联系。过程由一系列的活动组成。这些活动由开发者使用工具、方法和技术完成。过程之间是相互联系的。过程结果会影响到相关的以该过程结果为基础的过程。将分类方法建立在过程基础上可以更好地理解缺陷形成的过程,把握缺陷的本质,从根本上预防缺陷。

4.1 分类过程

分类过程分为三个步骤:①缺陷确认。根据测试人员提交的缺陷报告重现缺陷,确认测试对象存在报告宣称的缺陷。②缺陷测量。重现缺陷,观察缺陷现象,记录缺陷数据。缺陷数据一般包括发现缺陷的检测活动、引发缺陷的事件、缺陷来源、缺陷症状、缺陷的影响、硬件环境、软件环境、缺陷类型和缺陷的严重性等。③缺陷分类。根据缺陷数据将缺陷划归某个产生该缺陷的软件过程。

缺陷数据是缺陷分析活动必需的信息。通过分析缺陷产生的原因,就可能消除软件中尚未被发现的缺陷,尽可能地改善软件质量。改进有缺陷的过程,能够预防同类软件缺陷重复发生。当组织的过程得到改进时,软件组织的软件产品质量就得以提高。

不同开发组织、不同项目制定的软件过程不完全相同,应根据自身的软件过程制定具体的缺陷分类方案。基于过程的缺陷分类法的主要目的是分离、检测并消除缺陷,并为预防缺陷积累经验,寻找缺陷产生的过程原因,改进软件工程过程。

4.2 缺陷管理

确定了缺陷分类方法,缺陷的分类和管理就有章可循。由于在软件生产过程的各种验证、检查、测试和分类过程中产生了大量的缺陷数据,为妥善管理缺陷数据,必须建立缺陷数据库,实现缺陷数据管理自动化。此外,为了对缺陷进行统计和分析,需要开发相应的软件工具辅助缺陷分析人员的工作。缺陷分析人员定期报告缺陷分析结果。分析报告应包括缺陷产生的原因、缺陷的危害性、修复的优先级、缺陷出现的频率、各类缺陷的比例、对应的预防措施、过程的缺陷和对软件过程改进的建议等内容。

缺陷管理需要结合缺陷生命周期。缺陷生命周期是指从发现缺陷直到完成缺陷处理的过程。这个过程有六个状态:缺陷打开(Open);缺陷分析(Analyze);缺陷修改(Correct);缺陷搁置(Defer);缺陷验证(Verify);缺陷关闭(Close)。图1是缺陷状态转换图。

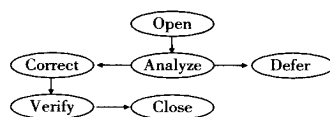


图1 缺陷状态转换图

缺陷打开状态是指缺陷被发现并被确定。缺陷分析是分析缺陷原因寻求解决方案,需要及时修复的要确定修复办法,需要搁置延期修复的要说明理由。缺陷验证是对修改的缺陷进行检查,确认修改成功。Open、Analyze、Correct、Verify 是活动状态;Defer 和 Close 是终结状态。缺陷最终要停留在终结状态。缺陷信息数据库要记录每个缺陷的生命状态并及时更新缺陷状态,这是缺陷管理的基本要求。软件开发人员要确保缺陷状态能够顺利转换到终结状态。

软件缺陷分类方案不同,采用的缺陷数据集可能不同,但最小数据集应该相同。最小数据集包括发现缺陷的检测活动、引发缺陷的事件、缺陷来源、缺陷症状和缺陷类型。缺陷数据应包括这个最小数据集。一个供选择参考的缺陷数据表表头设计如下:

缺陷编号	发现活动	发现位置	引发事件	症状	类型	状态
------	------	------	------	----	----	----

4.3 缺陷管理系统的要求

为满足大型的分布的软件开发环境的需要,缺陷管理系统必须基于计算机网络,采用浏览器/服务器结构,使缺陷管理成员通过 Web 浏览器获取缺陷数据,处理缺陷管理任务;利用自动电子邮件通知功能简化、加速和改善团队通信,加快缺陷处理的速度,提高缺陷管理的效率;建立与项目一致的规则集,使系统自动处理简单重复性任务,减轻人员的负担,将精力集中到缺陷原因分析上;分配给不同角色的用户相应的权限,保证管理工作的安全;支持结构化查询语言 SQL,具有强大的数据搜索功能;丰富的报表功能,便于组织高级管理人员了解缺陷管理工作的成效,制定合理的软件过程改进方案。

5 结论

为了生产出高质量的软件产品,传统方法是通过软件测试探测软件中的错误,包括单元测试、集成测试等各个级别测试。更有潜力的方法是缺陷预防。通过缺陷原因分析,确定缺陷原因并改正缺陷,确定并改正缺陷的过程原因,改进遗漏缺陷的缺陷检测活动,进行复查确保类似缺陷不再出现。通过如此严格缺陷预防过程必定能实现预期的质量目标。缺陷分类作为缺陷预防的有力工具,对它的研究将促进软件组织的软件过程改进和软件产品质量的提高,提升软件企业的竞争力。

(下转第 98 页)

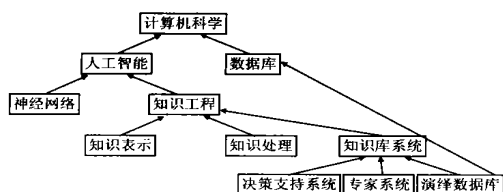


图2 已建立的一个关于学科的本体

那么上面的简单查询就涉及到概念之间复杂的逻辑、语义和语法关系:

专家是指在某领域具备深厚专业知识的人,如参加或完成相关领域研究和应用项目、发表过相关著作或论文、教授相关课程等人员都可能成为该领域的专家。假设在 Web 上目前有如下人员的资料:

```

< person>
  < name>      </ name>
  < xlink ref= http:// exp. edu. cn/ computers/ Mary. html/>
  < Books>
    < Book>
      < title> Design and Realization of Expert System </ title>
      < topic> Software Design</ topic>
      < topic> Expert System</ topic>
      < publisher> Beijing Publish</ publisher>
      < publish time> 2001</ publish time>
    </ Book>
    ...
  </ Books>
  ...
</ person>
  
```

一般认为,是某领域的子领域的专家同样也是这个领域的专家,所以应该是符合条件的结果输出。

可以看出这种基于本体和语义的信息检索更接近于实际情况,更能获取到我们所需要的信息。

(2) 电子商务

在电子商务中,不同的商家可能采用不同的内容标准和目录描述标准对其产品进行描述,这就为他们之间的信息交互带来了困难,因此,在 B2B 的电子商务中,需要解决:①联合不同的内容和产品描述标准;②能链接到不同的目录表示标准;③能联合不同的描述商业文档交互的标准,如订单等。

本体在 B2B 的通信中能起到信息集成的作用。如图 3 所示。

在 Ontology 层定义不同产品和文档标准提供的各种信息的术语。

另外,Ontology 在 Web 上完成其信息定义后,还能有效地帮助其完成 Web 数据管理,突破以前仅仅只是符号和数据管理,还能进行语义上的处理和管理。

4 实现

在具体的应用和实现过程中,应该注意以下问题:

(上接第 86 页)

参考文献:

- [1] 张海藩. 软件工程导论[M]. 北京:清华大学出版社, 1998.
- [2] 王立福, 张世琨, 朱冰. 软件工程——技术、方法与环境[M]. 北京:北京大学出版社, 1997.
- [3] 黄锡滋. 软件可靠性、安全性与质量保证[M]. 北京:电子工业出版社, 2002.
- [4] James S Collofello, Bakul P Gosalia. An Application of Causal Analysis to the Software Modification Process[J]. Software- practice and Experience, 1993, 23(10): 1095- 1105.
- [5] John W Hordh. Practical Guide to Software Quality Management [M]. Norwood: Artech House, 1996.
- [6] Mark C Paulk, et al. Capability Maturity ModelSM for Software [R]. Pitts-

(1) 选择合适的描述方法和语言

描述 Ontology 的方法有很多种,如形式化、半形式化、非形式化等,根据 Web 信息的特点,可以选择半形式化或形式化的方法。

目前描述 Ontology 的语言和框架有很多,有很多是基于—阶谓词的,如 Ontolingua, Cycl, Loom 等。对于 Web 上的应用而言,最好使用通用的语言来实现,避免之间的转换。XML 语言和 RDF 框架被认为是 Web 上数据交换的标准,有很多基于它们的描述语言被定义出来: OIL, SHOE, RDFS 等。

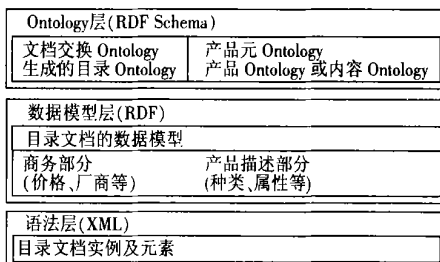


图3 B2B 电子商务信息组织分层

(2) 选择合适的创建 Ontology 方法

目前 Ontology 的开发还没有统一的标准,不同的应用和工程所遵循的创建过程和方法不同。例如现在出现的方法有: Mike Uscholdde & King 的骨架法; TOVE 评价法; KACTUS 工程法等,在具体的应用中,应该根据应用的特点选择和借鉴具体的创建和开发方法。

5 总结

语义 Web 是今后 Web 建立的方向,而 Ontology 对其实现起着重要的作用。Ontology 基础上的语义 Web 在很多应用方面有着广阔的前景,本文只讨论了其中的几个方面。要真正实现语义 Web 并完成 Ontology 在它上面的应用,还有很多问题需要研究,包括它在 Web 上的合适的创建方法和框架,以及成功的实例。

参考文献:

- [1] T B Lee. Semantic Web Architecture [EB/OL]. http://www. w3. org/ 2000/ Talks/ l206- xml2k- tb/ slide10- 0. html, 2001- 07- 20.
- [2] Borys Omelayenko. Preliminary Ontology Modeling for B2B Content Integration[EB/OL]. http://www. cs. vu. nl/~ borys, 2001- 01.
- [3] 杨秋芬, 陈跃新. Ontology 方法学综述[J]. 计算机应用研究, 2002, 19(4): 5- 7.
- [4] 邓志鸿, 唐世渭, 张铭, 等. Ontology 研究总述[J]. 北京大学学报, 2002, 38(5): 730- 738.

作者简介:

邓芳(1972-), 副教授, 研究方向为知识工程、决策支持、数据挖掘及数据仓库。

burgh, Pennsylvania: Carnegie Mellon University, 1993.

- [7] GJB 437- 88 军用软件开发规范[S].
- [8] GB/T11457, 软件工程技术语[S].
- [9] Putnam Lawrence H, Myers Ware. Measures for Excellence: Reliable Software on Time within Budget[M]. Prentice Hall, 1992.
- [10] Ram Chillarego et al. Orthogonal Defect Classification: A Concept for Inprocess Measurements[J]. IEEE Transactions on Software Engineering, 1992, 18(11): 943- 956.
- [11] IEEE Std 1044-1993. IEEE Standard Classification for Anomalies [S].

作者简介:

聂碧波(1978-), 男, 湖北孝感人, 硕士研究生, 主要研究方向为软件工程与软件可靠性; 刘孟仁(1940-), 男, 湖南汨罗人, 教授, 主要研究方向为计算机软件工程、计算机可视化技术。