

一个软件测试过程管理工具

单 方¹, 陈 璞², 李美村¹, 王志明¹, 杨 斌¹, 贺 忠¹

(1. 广州大学科技干部学院 计算机系, 广东 广州 510091;

2. 广东工业大学 机械工程系, 广东 广州 510643)

摘要: 论述了软件测试过程管理的重要性, 并介绍了研制的一个软件测试过程管理工具—TestMonitor.

关键词: 软件工程; 测试过程; 软件缺陷

中图分类号: TP311.56 文献标识码: A 文章编号: 1007-7162 (1999) 02-0088-05

1 关于软件测试过程管理

软件测试在软件生存周期中占有重要的地位, 这不仅是因为测试阶段占用的时间、花费的人力和成本占软件开发的很大比重, 而且它直接影响着软件的质量, 是保证软件可靠性的主要方法之一, 其目的是保证发现软件缺陷 (错误)^[1]. 但是除了要测出软件缺陷, 还有必要对软件缺陷进行分析和处理, 从而对软件缺陷的消除、或者对软件的评价及决策提供依据. 这一问题已经显得越来越重要了.

目前的测试工具大多集中于软件功能测试和结构测试, 而缺乏对软件测试过程管理的支持, 也不能对测试出的软件缺陷进行分类和分析, 而软件缺陷数据是软件质量的直接反映, 而且还直接影响到软件项目的进度、费用和资源分配等问题. 在软件交付阶段, 项目管理人员应能够根据已经被发现的软件缺陷的信息来评估项目的进展状况, 例如: 根据在一定的时间内被发现的软件缺陷的数量, 应用线性回归的方法来预测软件缺陷数量的走势, 根据软件缺陷的分布来确定项目的进展情况等.

对已发现软件缺陷的处理是软件开发当中必不可少的活动, 合理的软件缺陷处理过程将会极大地提高工作的效率. 对软件缺陷处理过程的最基本的要求包括: 每一个被发现的问题都应该按类别和优先级进行分类; 整个处理过程应该是闭合的, 即确保每一个被发现的问题在报告之后, 及在过程中都能得到解决; 在整个过程中追踪问题的状态; 问题记录在整个合同期内都得到维护; 对问题的改正应该进行验证, 以确保问题确实被解决, 不利的影响已经被消除, 并且解决该问题所引起的变化不会带来新的问题.

根据以上的分析, 我们认为有必要为 PC 环境提供一个功能强大、使用简单灵活的软件测试过程管理工具, 为项目管理人员提供一个对软件缺陷分类、项目进展和软件质量评估的有效手段.

2 TestMonitor 系统概述

软件测试过程管理工具 TestMonitor 是一个基于 Windows95 和 WindowsNT 环境的实时地

收稿日期: 1999-01-21

作者简介: 单 方 (1956-), 男, 讲师; 主要研究方向: 专家系统.

©1994-2012 China Academic Journal Electronic Publishing House. All rights reserved. http://www.

记录和管理测试阶段信息的软件开发支持工具，也是一个基于软件缺陷的度量工具。这是作者与北京航空航天大学软件工程研究所联合研制和开发的。

在开展这一课题的研究过程中，我们首先研究了确定软件缺陷优先级的标准，根据在 1988 年公布的美国国防部软件开发标准 [DOD-STD-2167A]^[2] 和 1995 年公布的软件开发与文档标准 [MIL-STD-498]^[3]，制定了 TestMonitor 中软件缺陷的优先级。在此基础上我们建立了软件缺陷处理过程模型，根据有效性、完备性和正交性等原则，确定软件缺陷的状态及状态之间的转换关系，然后确定状态同角色、活动、信息之间的关系。软件缺陷处理过程模型的建立提高了我们对软件缺陷处理过程的理解，并为软件测试过程管理工具 TestMonitor 系统的实现打下了基础。

软件缺陷数据对于软件质量保证有着重要的意义，因此我们根据软件缺陷处理模型设计和实现一个软件测试过程管理工具 TestMonitor，一方面实现在软件开发和维护过程中对软件缺陷处理的辅助支持，另一方面实现了软件缺陷数据的自动采集和基于软件缺陷数据的度量、分析功能。TestMonitor 能够根据软件缺陷的状态，确定将要采取的活动、负责实施的角色和所要传递的信息流，在活动转换时自动通知下一个角色，实现严格的角色权限控制；能够对软件缺陷数据进行采集、查询，并根据软件缺陷的数据进行统计、分析和输出报表。在软件缺陷的处理过程中，要涉及到的人员包括测试人员、开发人员、管理人员等等。他们一般处在不同的地理位置，而在解决问题的过程中又要交换大量的信息，因此，TestMonitor 还具有灵活而有效的通讯方式。

TestMonitor 是基于 Windows95 和 WindowsNT 操作系统的，运行于网络环境下的一个应用系统。TestMonitor 系统内部结构和外部支持环境如图 1 所示。

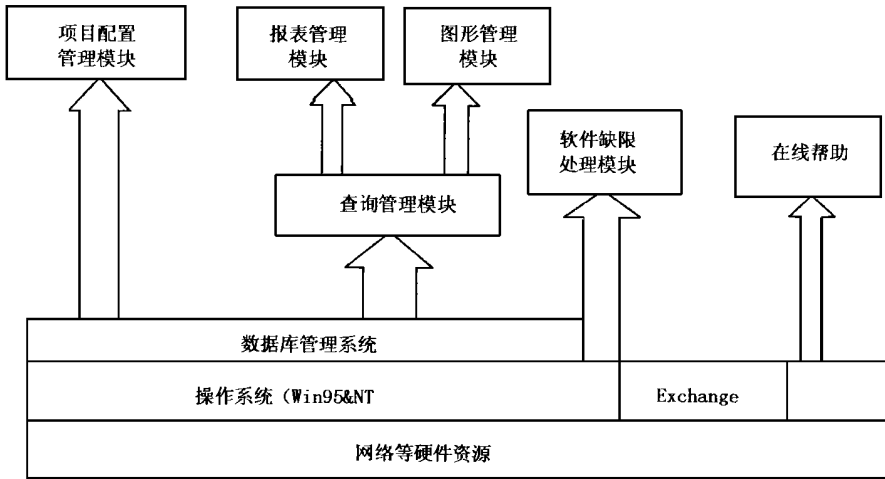


图 1 TestMonitor 系统功能模块及其支持环境

项目配置管理模块自身构成一个独立的应用程序，即 TestMonitor 服务器程序。它被用来配置和管理项目的数据库文件。在服务器程序中，用户可以创建新的项目或配置已有的项目文件；可以在项目中添加新的用户；可以将系统划分为若干模块，而每个模块指定专门的用户来负责；可以更新系统的版本；在项目配置结束时，检查项目的完整性。

软件缺陷处理模块是 TestMonitor 系统的核心部分，它一方面实现对软件缺陷处理过程的

全面支持, 另一方面为查询、报表、图形等模块提供所需的数据。软件缺陷处理模块提供对软件缺陷的追踪和处理功能。通过软件缺陷处理模块, 用户可以提交新发现的软件缺陷、修改软件缺陷记录的属性、为软件缺陷添加建议, 还可以删除冗余的缺陷记录。软件缺陷处理模块是系统中的主要模块, 其中对软件缺陷记录状态属性的修改完全遵循软件缺陷的状态转换关系图。

TestMonitor 支持查询条件用户定制化, 即支持用户根据自己的想法所确定的查询条件。对于一些常用的、用户所定制的查询条件, 具有存储的功能。通过查询管理模块, 用户可以按照缺陷记录的属性信息对数据库中的记录进行查询。查询的结果可以作为统计图形或数据报表的数据源。

图形管理模块从数据库中提取数据, 做简单的统计之后以图形的方式进行输出。图形的风格有两种: 柱状图和饼图。图形的类型包括模块分布图、颜色分布图、状态分布图、类别分布图等等。

报表管理模块接收用户指定的软件缺陷记录, 把它们按预定的报表格式输出到屏幕或打印机。报表的数据源可以是查询模块输出的结果, 也可以是用户精确指定的软件缺陷记录。报表的格式由系统提供。

3 软件缺陷处理过程

软件缺陷处理是 TestMonitor 系统的核心部分, 本节主要对软件缺陷处理过程作进一步的论述。

对软件缺陷处理过程最基本的要求包括以下几个方面:

①整个处理过程应该是闭合的, 即确保每一个被发现的问题在过程中都能得到解决; 在整个过程中追踪缺陷的状态; 问题记录在整个合同期内都得到维护。

②每一个被发现的软件缺陷都应该按类别和优先级进行分类。

③对软件缺陷的改正应该进行验证, 以确保问题确实被解决、不利的影响已经被消除, 并且解决该问题所引起的变化不会带来新的问题。

3.1 软件缺陷处理过程模型—SPDM

在实际的情况当中, 我们采用非形式化的方法来建立软件缺陷处理过程模型 SPDM。更具体地说, 是采用基于状态的建模方法。这主要是基于以下几个方面的考虑:

建模目的: 我们建立过程模型的最终目的是要实现对软件缺陷处理过程的支持。

基于实际情况的考虑: 软件缺陷的状态是整个软件缺陷处理过程的核心。软件缺陷的状态反映出软件缺陷处理过程所处的阶段; 软件缺陷的状态决定了过程中的下一个活动; 软件缺陷的状态决定了过程中应负责的角色。

基于状态的建模方法是基于控制流的建模方法的一种变形, 这种方法首先要确定过程中所需的状态的转换关系, 然后确定状态同角色、活动、信息之间的关系。下面将以软件缺陷的状态为核心, 从角色、活动、信息流等几个方面来描述软件缺陷处理过程。

3.2 软件缺陷状态

软件缺陷状态是描述软件缺陷处理过程所处阶段的一个引发属性。对应于不同的状态, 我们能确定对该问题的处理已经进展到什么阶段, 还需要进行哪些工作, 需要哪些人员的参与等信息。在 SPDM 模型中, 我们确定了 5 种状态, 来描述软件缺陷的生命周期, 即“活

跃”、“已修订”、“关闭”、“已延迟”和“无法再现”。对于 SPDM 中的 5 种状态, 根据它们在实际的软件缺陷处理过程中所反映出来的任务进展状况的不同, 可以分为 3 种稳定程度, 即稳定的、半稳定的和不稳定的状态。其中“关闭”为稳定状态, “已修订”、“已延迟”和“无法再现”为半稳定状态, “活跃”为不稳定状态。稳定状态反映的是对问题的处理过程已结束, 不再需要进一步的工作。半稳定状态反映的是对问题的处理已经进展到了一定的程度, 虽然还没有得到最后的解决, 但该问题不会影响整个系统的一次新的 BUILD。不稳定状态反映的是问题的解决仍处于初期阶段、对问题的调查刚刚开始, 或负责解决的人员需要更进一步的问题描述信息等等。

3.3 角色

角色是与某一功能目标有关的权利与义务的一个集合。角色是人们描述软件过程时对过程实施者的一种抽象, 而在过程的具体实施时, 角色要有合适的人来扮演。某一实施者一旦与一个角色连接起来, 他便拥有了角色所包含的权利与义务。在 SPDM 模型中, 我们确定了 3 种角色: 项目负责人、开发人员和测试人员。

3.4 活动

活动是一个基本的过程步。活动在一定的条件下被激活, 接受输入信息, 进行一定的处理, 在满足出口条件时退出活动, 并输出一定的结果信息。在任务执行过程中, 可能还要接受反馈信息, 或传递反馈信息到其他的活动。在 SPDM 模型中的活动主要是更改软件缺陷属性信息的活动。活动的入口条件和出口条件主要都是由软件缺陷所处的状态决定的, 活动的实施也是和软件缺陷的状态相对应的。

3.5 信息流

在软件缺陷处理过程中所流动的信息, 除状态转换时的通知信息外, 其余的都是软件缺陷的属性信息, 包括软件缺陷的自然属性和引发属性。如果仅仅从软件缺陷属性变化的角度来看, 我们对软件缺陷的处理过程也可以这样理解: 每个软件缺陷的所有的属性信息组成一个大的表格。这样的一张表格在问题处理过程中流动, 过程中的每一次活动, 都是由相应的角色参考表格中已有的信息, 执行一定的动作之后, 添加或改写表格中的某些单元, 然后将其转入下一个活动。这样一直进行下去, 直到对软件缺陷的处理结束。此后这张表格中的数据将不再发生变化。在每个活动进行过程中, 角色所能改变的软件缺陷的属性是由软件缺陷当前所处的状态和用户的权限所共同决定的。

3.6 软件缺陷处理工具

软件缺陷处理工具是 TestMonitor 系统的核心部分, 它一方面实现对软件缺陷处理过程的全面的支持, 另一方面为查询、报表、图形等模块提供所需的数据。为实现对 SPDM 模型的支持, 软件缺陷处理模块在设计时体现了以下的思想:

①软件缺陷状态在整个软件缺陷处理过程中的核心地位。软件缺陷当前所处的状态决定了目前应负责任的角色、下一步应该进行的活动。在软件缺陷状态发生变化时自动通知下一个应负责的角色。

②角色是一定权利和义务的体现。不同级别的用户对同一个软件缺陷拥有不同的处理权限。在具体实现时, 对于 SPDM 模型中的每个活动, 都提供相应的命令来支持; 对于每个级别的用户, 只能实施同相应角色符合的命令; 状态转换时, 以 MAIL 的形式通知下一个应负责的角色。

4 结束语

软件测试过程管理工具 TestMonitor 课题的技术要点是: (1) 提出了“软件测试过程管理”的思想, 强调对软件测试过程应该通过更为严格的、量化的方法进行管理; (2) 建立了软件缺陷处理过程模型. 根据有效性、完备性和正交性原则确定了软件缺陷的状态及其转换关系, 并进一步确定状态同角色、活动、信息之间的关系; (3) 提出了软件缺陷优先级的分类标准; (4) 提出了对软件测试数据进行统计和分析的方法; (5) 设计和实现了一个基于 Windows95 和 NT 平台的软件测试过程管理工具, 一方面实现了对软件测试过程管理的自动化的支持, 另一方面为基于软件测试数据的软件质量的度量提供了基础数据.

参考文献:

- [1] 周苏, 陈敏玲, 陈根才等. 软件工程及其应用 [M]. 天津: 科学技术出版社, 1992. 234~ 236.
- [2] Military Standard. Defense System Software Development (DOD- STD- 2167A) [S]. Washington D C: United States Department of Defense, 1989. 146~ 148.
- [3] Military Standard. Software Development and Documentation [S]. MIL- STD- 498, 1994. 112~ 114.

A Software Test Process Management Tool

Shan Fang¹, Chen Pu², Li Meicun¹, Wang Zhiming¹, Yang Bin¹, He Zhong¹

(1. Dept. of Computer, College of Technocrats, Guangzhou University, Guangzhou 510090, China;

2. Dept. of Mechanical engineering, GDUT, Guangzhou 510643, China)

Abstract: It discusses the importance of the software test process management, and introduces a software test process management tool, TestMonitor.

Key words: software engineering; Test process; software defect

(责任编辑: 利 民)