

缺陷密度法的软件测试质量度量分析技术

尤 艺,段洪栋,王瀚超

(北京航天自动控制研究所,北京 100854)

摘要:阐述了航天型号软件测试质量度量的现状,通过构建缺陷度量测试工具平台,提供了一种软件确认测试项目的质量数据统计方法和分析预测方法,指导嵌入式、非嵌入式及 FPGA 等各类软件第三方确认测试过程中软件缺陷的发现,提高软件测试的充分性和有效性。分别从测试度量平台搭建的重要性以及该平台应用于测试过程中软件缺陷度量、分析及预测等方面进行深入剖析,为提高软件测试的效率,发现软件开发设计缺陷提供有效真实的依据和方法。

关键词:软件测试;质量度量;缺陷密度

中图分类号:TP311.5 **文献标识码:**B

Quality Metrics and Analysis System Based on Defect Density Method of Software Testing

YOU Yi, DUAN Hong - dong, WANG Han - chao

(Beijing Aerospace Automatic Control Institute, Beijing 100854, China)

ABSTRACT: This paper expounded the present situation of aerospace software test quality metric. By constructing a defect metric platform, we provided the statistical method and the forecasting analysis method of software testing quality data. The method guides to find software defect in software testing process, and improves the adequacy and effectiveness of software testing.

KEYWORDS: Software testing; Quality metrics; Defect density

1 引言

随着软件在航天产品的广泛应用,以及软件质量对航天发射成败重大影响越来越突显,总装对航天型号软件测试的要求越来越高,要求进行独立第三方确认测试的软件范围也越来越大(以前仅局限于对 A、B 级软件进行测试,现在扩大到 C、D 级软件),对于第三方确认测试项目的质量度量就显得越发重要和迫切。传统的测试质量度量方法已不能适应测试发展需求,通过构建工具化的缺陷度量统计平台,并在大量实际软件测试项目的质量数据基础上,对各项目缺陷进行自动化统计分析和预测分析,掌握第三方确认测试项目中软件缺陷分布、分类情况,提高软件测试的针对性,研究各类软件规模、测试粒度、用例强度等因素在测试过程中对软件缺陷分布的影响,是一件十分必要、有价值的工作。没有产品的质量度量,就等于没有产品的质量标^[4]。

2 测试度量平台搭建的重要性

软件质量度量的意义主要在于:它是有效产品质量管理的基础,是改进用户满意度和产品质量的数据来源和依据;是减少产品开发和售后服务支持费用的有效手段。

软件测试作为软件需求、设计以及实现的最终审定,仍然是软件开发周期中清除已产生的缺陷、降低缺陷密度,保证软件质量的主要手段。软件测试质量度量的意义在于:它能有效实行测试质量分析和管理的;可及时检查测试进度和质量;帮助发现测试漏洞;比较测试质量变化趋势,风险分析;帮助找出最佳实践。

传统的软件测试质量度量方法是通过测试人员的纸质问题记录单进行统计和分析,其缺点是:测试人员主要关注的是对发现的软件问题的分析,对问题的分类考虑的比较少,且没有统一的细化的分类度量标准,统计人员无法提取到有效地分类度量信息,同时,由于这些纸质问题记录单中的数据信息无法被自动提取、分类和整理,单靠人工去统计分析记录中的信息,当测试项目日益繁多时,其统计分析的工作量会成倍增加,其统计数据的可靠性、可信性和客观性

也会下降,即质量问题度量分析缺乏有效的数据提取手段,在此基础上分析数据的可靠性和有效性得到质疑。而最佳实践实际上是经验积累总结的结果,由于没有一个有效可靠的度量管理平台,涉及的不定因素和人为因素太多,片面性和不准确性的分析结果不但不能帮助发现漏洞和指导工作,还会造成一些负面的影响以及人力资的浪费和质量成本的提高。因此,一个工具化的软件缺陷质量度量平台建立具有十分突出的现实意义。

3 测试过程软件缺陷度量、分析及预测

软件质量度量可以分为三大类:产品度量、过程度量和项目度量。产品度量用来描述软件产品的特征,例如软件规模大小、软件设计特征、软件性能以及质量水平。过程度量是用来提高和改善软件开发维护过程的度量,例如缺陷移除效率,测试阶段中的缺陷到达模式,以及缺陷修复过程的反应时间。项目度量主要是用来描述项目的特性和执行,例如开发人员数量,软件周期中的人员使用模式、费用、进度等。本文从测试的角度只对产品度量进行分析和建模,对后两种不做讨论^[2]。

3.1 产品度量模型的建立与应用

产品度量包括三个层面:产品本质质量度量、用户满意度度量和综合指标度量。

产品本质质量度量主要通过缺陷密度度量来实现,即采用软件中“BUG”的数目来测量。这里提到的缺陷并非是指区别于错误和故障的产品异常,而是统指测试过程中发现的软件产品问题。

缺陷密度度量方法可按照三种方法对问题进行分类度量:

按严重等级进行度量。即根据发现的软件问题对任务、整个系统、人员安全等的影响程度不同,分为致命、严重、一般、轻微 4 个级别的缺陷问题。

按问题分布情况进行度量。

按问题分类情况进行度量。

用户满意度度量方法可通过委托方满意度情况主要从测试产品满意度和服务满意度两个层面,量化到 5 个方面进行度量进行度量。

产品综合指标度量主要是指与软件问题相关的综合质量数据:如软件规模,千行代码缺陷率,被测件类型,开发方行业分类,芯片配置合理情况分析统计等,以及测试人员的工作量和进度度量^[2]。

软件测试质量度量平台运行在测试专用局域网络,其系统总体功能架构如图 1 所示^[5]。

应用层以软件产品度量类型为主线,从三个方面实现了对软件测试项目全过程中发现的软件缺陷及软件综合特性数据等进行管理。由于软件类型不同,其产品本质质量度量要求也不同,因此,在度量平台流程设计中,分两大类进行设计:一类是嵌入、非嵌入式软件,另一类是 FPGA 类软件。其

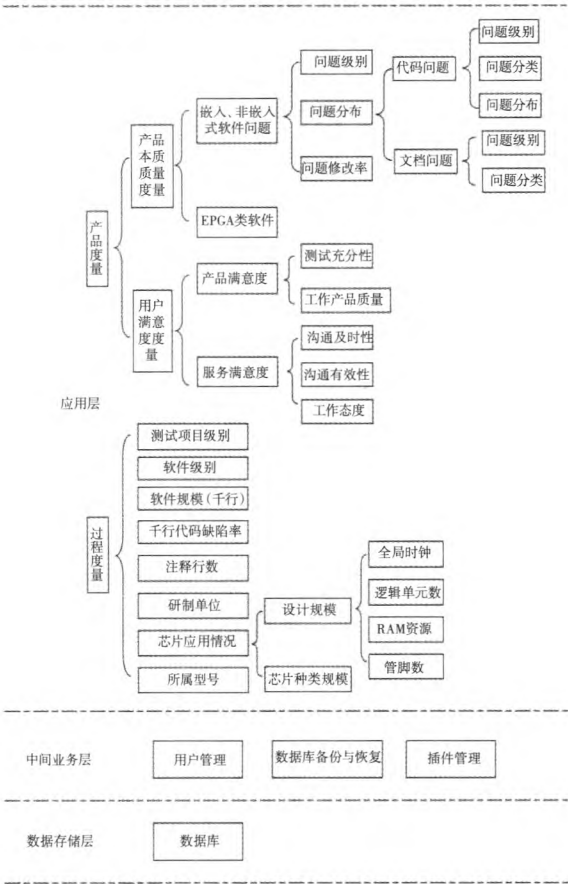


图 1 软件测试质量度量平台的总体架构

设计后应用实现情况见图 2、图 3、图 4、图 5、图 6 所示。

嵌入、非嵌入式软件测试项目综合数据							
序号	项目名称	被测软件类型	被测软件级别	开发单位	项目组长	项目截止日期	项目截止日期
1	软件配置管理: 非嵌入式软件	C 类			2013-07-15	2013-10-12	
2	软件配置管理: 非嵌入式软件	C 类			2013-07-15	2013-10-12	
3	控制软件配置管理: 非嵌入式软件	C 类			2013-07-15	2013-10-12	
4	控制软件配置: 嵌入式软件	A 类			2013-10-11	2013-10-24	
5	软件配置管理: 嵌入式软件	A 类			2013-08-01	2013-10-11	
6	软件配置管理: 嵌入式软件	B 类			2013-06-20	2013-09-08	

图 2 嵌入、非嵌入式软件测试项目综合数据

文档问题汇总统计表											
序号	测试项目名称	文档问题统计				文档问题分类					
		设计	需求	开发	测试	文档不一致	文档不一致	文档不一致	文档不一致	文档不一致	文档不一致
1	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
2	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
3	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
4	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
5	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
6	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
7	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
8	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
9	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
10	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0
11	软件配置管理: 非嵌入式软件	0	0	0	0	0	0	0	0	0	0

图 3 嵌入、非嵌入式软件文档问题缺陷度量

3.2 度量数据的分析方法及预测方法

为了提高研究成果的实用性,本文对大量实际测试项目

项目/问题名称	问题类别										问题子类									
	1级	2级	3级	4级	5级	6级	7级	8级	9级	10级	编程/语言使用	初始化和复位	时序性能	时钟信号和跨时钟域	通信接口	管脚配置	存储器控制	容错与防错	其他	其他
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

图4 嵌入、非嵌入式软件代码问题缺陷度量情况

项目/问题名称	问题类别										问题子类									
	1级	2级	3级	4级	5级	6级	7级	8级	9级	10级	编程/语言使用	初始化和复位	时序性能	时钟信号和跨时钟域	通信接口	管脚配置	存储器控制	容错与防错	其他	其他
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

图5 FPGA类软件代码问题缺陷度量统计情况

项目/问题名称	问题类别										问题子类									
	1级	2级	3级	4级	5级	6级	7级	8级	9级	10级	编程/语言使用	初始化和复位	时序性能	时钟信号和跨时钟域	通信接口	管脚配置	存储器控制	容错与防错	其他	其他
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10
软件配置问题	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10

图6 FPGA类软件芯片应用情况

数据进行统计分析,建立软件缺陷度量数据集作为本文研究软件缺陷预测模型的基础。以代码问题为例,对近两年平均每年一百多个被测软件(包括嵌入式软件类、非嵌入式软件类、FPGA类三个类型)测试项目所发现的缺陷分类关系及时间比例趋势进行统计^[3],结果见图7、图8所示^[1]。

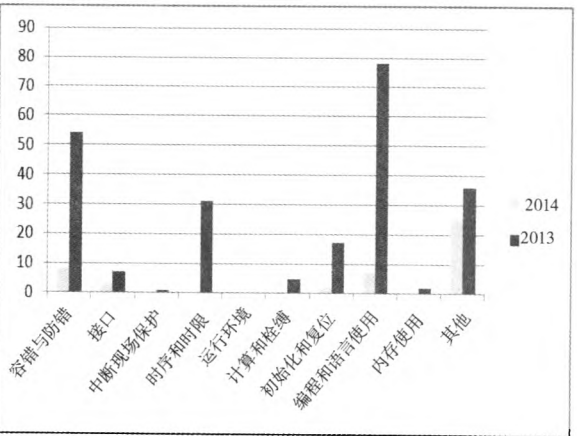


图7 2014、2013年嵌入及非嵌软件代码问题分类对比

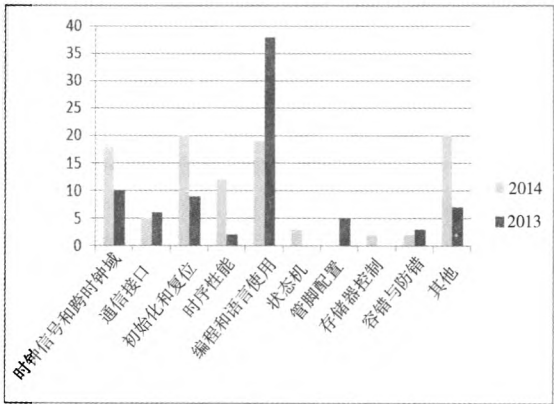


图8 2014年与2013年FPGA软件代码问题分类对比

以FPGA类软件为例,对其代码分类情况进行分析(见图9)^[1],得出以下结论:“编程和语言使用”、“初始化和复位”、“时钟信号和跨时钟域”、“其它”是FPGA软件代码问题的主体,占10类问题比例的67%,且“时钟信号和跨时钟域”、“其它”和“初始化和复位”这三类比上一年有明显上升趋势。经分析发现上述三类软件问题主要集中在部分开发单位的FPGA软件方面,反映了同行业中不同单位的FPGA软件开发水平。此外,“时序性能”问题有明显上升趋势,经分析主要故障原因有两种:一是异步逻辑电路产生的竞争冒险问题,可采用代数法和选通法两种方法避免此类问题的发生;二是接口时序与芯片手册时序要求不一致的问题,这需要对设计人员进行FPGA时序设计和时序分析知识培训,让设计人员意识到FPGA设计不仅要满足功能要求,还要满足时序要求。在完成FPGA接口模块设计后,需要开展模块级仿真测试,保证接口时序与芯片手册时序要求一致。另外,针对设计中使用的时钟和时序关键路径,需要编写时序约束文件,保证FPGA内部电路的时序性能都处于设计者控制范围内。

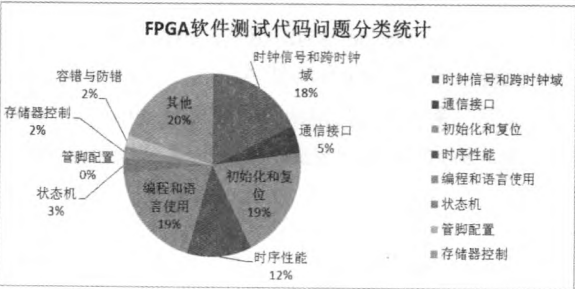


图9 FPGA软件测试问题统计

4 结论

通过设计一种第三方软件确认测试质量度量自动化统计分析工具,有效解决了软件缺陷问题的趋势分析及预测技术,为提高软件测试的效率,发现软件开发设计缺陷提供有

效真实的依据和方法。

参考文献:

[1] 李建. 软件过程质量度量与控制(第一版)[M]. 北京:清华大学出版社, 2006.

[2] Stephen H Kan 著, 吴明晖, 等译. 软件质量工程——度量与模型(第二版)[M]. 北京:电子工业出版社, 2004.

[3] 涂亚明, 毛军鹏, 余静, 尹磊. 系统测试阶段的软件缺陷预测模型分析[C]. 中国合肥第六届中国测试学术会议论文集(CTC2010), 2010-7:164-167.

[4] 袁玉宇. 软件测试与质量保证[M]. 北京:北京邮电大学出版社, 2008-4.

[5] 王俊伟, 等. SQL Server 2000 中文版数据库管理与应用标准教程[M]. 北京:清华大学出版社, 2006-11.

[作者简介]



尤 艺(1967-), 女(汉族), 四川内江人, 硕士, 高级工程师, 主要研究方向为嵌入式软件编程, 软件测试质量控制。

段洪栋(1981-), 男(汉族), 河北冀州人, 硕士, 工程师, 主要研究方向为嵌入式软件测试。

王瀚超(1986-), 男(汉族), 北京人, 硕士, 助理工程师, 主要研究方向为嵌入式软件测试。

(上接第 450 页)

将支持向量机 SVM 方法的惩罚因子设置成 $C = 3000$ 。

表 2 两种方法的比较结果

检测方法	敏感性	准确性	特异性	假阳性
方法	0.6481	0.7364	0.6416	0.4806
传统方法	0.3559	0.5793	0.7992	0.3462

针对同组样本集, 分别采用方法和传统 AULPCNN 赋时矩阵方法进行比较实验, 分析表 2 可以看出, 针对同组样本集, 与传统 AULPCNN 赋时矩阵方法相比, 方法具有较高的敏感性及准确性, 这是因为方法针对各种晚期肺结节图像特征对晚期肺结节进行识别, 得到的结果更加准确。

5 结论

提出了一种基于分类识别技术的晚期肺结节图像特征识别方法, 通过近似体积特征对圆形肿块特征进行描述, 采用傅里叶描述子和类圆度对分叶特征进行体现, 依据紧凑度与边界不规则度对棘状突起特征进行判断, 通过统计矩向量和不变矩向量对结节特征和空泡特征进行描述。通过 Mahalanobis 距离对肺结节图像特征进行识别, 依据距离与阈值之间的大小关系, 判断是否为肺结节。对经过识别确定为肺结节的图像区域, 依据晚期肺结节图像的一般表现及特征, 通过拉格朗日乘法对目标函数的最优解进行求解, 有效的完成了晚期肺结节图像特征的识别。仿真结果表明, 所提方

法具有很高的检测精度, 明显优于传统方法。

参考文献:

[1] 李玲, 等. 基于纹理特征和支持向量机的 ALOS 图像土地覆被分类[J]. 国土资源遥感, 2011, 4(6):58-63.

[2] 郭冬梅, 等. 基于灰度共生矩阵的大鼠肝细胞癌、肝硬化结节超顺磁性氧化铁 MR 增强图像纹理特征分析[J]. 中国医学影像技术, 2010, 26(3):563-566.

[3] 龚家强, 李小宁. 基于共生矩阵纹理特征提取的改进算法[J]. 计算机工程与设计, 2011, 32(6):2068-2071.

[4] 张弘, 范九伦. 灰度-梯度共生矩阵模型的加权条件熵阈值法[J]. 计算机工程与应用, 2010, 46(6):10-13.

[5] 梁小霞, 封筠. 基于 Gabor 变化和灰度梯度共生矩阵的人耳识别研究[J]. 石家庄铁道大学学报(自然科学版), 2011, 24(1):78-84.

[6] 廖璠, 孙季丰. 基于视觉注意力模型的医学图像目标检测[J]. 计算机仿真, 2010, 27(9):239-241.

[作者简介]



阮曙芬(1980-), 女(汉族), 湖南娄底人, 博士在读, 讲师, 主要研究方向:信号与信息处理。