构件可测试性模型研究与应用

傅　程[1]　宫云战[2]　洪　慧[1]
(1. 中国电子系统设备工程公司研究所, 北京　100039; 2. 北京邮电大学计算机科学与技术学院, 北京　100876)

摘　要: 本文分析了目前军队构件化软件开发的实际情况,在已有的工作基础上,提出了一个对构件可测试性进行度量的五边形法则,该方法能够直观地体现被测构件的可测试性指标数值,能够让构件开发者清楚地了解构件存在的缺陷,指导开发人员设计出高可测试性的构件。针对笔者的实际测试项目,通过先后几次的实验数据表明五边形法则可以有效地提高构件可测试性,并得到了开发方及测试人员的认可。

关键词: 软件构件; 可测试性; 软件测试
中图分类号: TP311.5

引　言

随着软件行业的迅猛发展,软件逐步成为构件组装的集合体,就像组装汽车一样,软件的各项功能,可以被不同的构件模块所负责,不同需求的软件可以通过不同的构件组装来实现。这样,构件自身的质量对软件的整体质量至关重要。然而根据近几年来的测试工作证明,组装后的软件成品在提交给测试方后,顺利通过测试的寥寥无几,往往是构件自身的问题影响了整个软件的质量[1]。这就对构件开发提出了严格的规范要求,本文所要引入的构件可测试性模型就是用以提高构件自身质量的一种度量方法。构件具有良好的架构设计,完备的接口参数,才能够使得软件测试更加高效和可靠,同时产品维护也更加便利,从而提高整个软件的质量,达到测试部门的测试要求。

1　现有的可测试性模型

目前业界有许多专家学者都在研究构件可测试性模型,最有实际应用意义的是 Gao 等人提出的一个在透视图法的角度来分析构件的可测试性模型。他们定义构件可测试性为五条特性:可理解性、可观察性、可控制性、可跟踪性和测试支撑能力[2]。

本文在 Gao 的模型基础之上,针对军队构件化软件开发实际,将 Gao 的可测试性模型加以修改,根据军队构件测试的实际需要进行删增,建立了一套新的可测试性模型。

收稿日期: 2007-04-20
第一作者: 男,1980 年生,硕士生
E-mail: fuchengzh @hotmail.com

2　构件可测试性模型

本文通过一个实际的测试项目来说明构件可测试性模型的应用意义。作战编成构件是某军用软件共用平台系统的重要组成部分,是由地方工业部门负责开发并组装的,其构件特征符合本文所提出的可测试性模型要求,因此在对该构件的整个测试工作中,课题组使用了模型,并得到了良好的测试效果。

模型各分支的打分是以软件工程开发规范和经典代码规则为准则,尺度是以达到和未达到为基准,按符合要求的单位量增减 分值,同一分支按实际得分占总分数的百分比来确定该分支的最终分数。

2.1　构件可理解性

构件可理解性是指开发方以文档形式对构件物信息表述的优劣程度[3]。通过实验,得出以下结论如表 1 所示。

表 1　作战编成可理解性结果

Table 1　Campaign organize understandability result

构件可理解性	分值结果
文档有效性	10
文档可测试性	8
需求可测试性	7
需求可测量性	9
构件可用性	10

①文档有效性:本构件的文档提供是非常完善的,充分地解释了构件物,包括需求分析,设计规范和用户参考手册。对该分支打分为 10 分。

②文档可读性:本构件的文档虽然非常全面,但

部分文档需要跟开发方人员沟通才能顺利解读。对该分支打分为8分。

③需求可测试性:在非功能需求方面,该构件基本能够做到对软件需求的完善,可以进行测试。在功能需求方面,该文档对功能结构的描述非常清楚,可测试性良好。但部分控制接口的参数设置没有说明清楚,对测试工作的进展产生了些许影响,对该分支打分为7分。

④需求可测量性:通过外部接口测试可以对功能需求进行测试。覆盖的需求点比较全面,对该分支打分为9分。

⑤构件可用性:通过查看《软件需求规格说明-导航工具》、《软件需求规格说明-编组工具》两份文档,用户可以容易地使用。对该分支打分为10分。

2.2 构件可观察性

构件可观察性是指根据一个程序的操作行为、输入参数及输出结果,来检验观察这个程序的简单程度[4]。通过实验,得出以下结论如表2所示。

表2 作战编成可观察性结果

Table 2 Campaign organize observability result

构件可观察性	分值结果
GUI可观察性	29/3
功能可观察性	10
外部交互可观察性	10
信息流转的可观察性	10

①GUI可观察性:该构件带有GUI接口,全部功能均有交互界面,使用者简单易学。

通过对"用户输入"、"GUI事件"、"输出给用户"三条分支因素来分析:在这三个方面,该构件的得分很高,分别为10分、10分和9分。这一分支的平均得分为29/3。

②功能可观察性:本构件所定义的功能清晰、明了。在通过阅读用户手册之后,一般的软件操作人员均能够轻易使用本构件。对该分支打分为10分。

③外部交互可观察性:由于这一分支的评价没有实质性内容,对该分支打分为10分。

④信息流转的可观察性:作战编成的信息流转是一个用户输入编成信息,最后导出特定含编成信息格式的文件的过程,这种能力是一个构件内部的基本功能,该构件的流转可观察性很好。对该分支打分为10分。

2.3 构件可控制性

构件的可控制性是指用户控制一个程序(或构件)的输入/输出、操作和行为的难易程度,通过实验,得出以下结论如表3所示。

表3 作战编成可控制性结果

Table 3 Campaign organize controllability result

构件可控制性	分值结果
行为控制	6
特性定制	1
构件执行控制	2
构件内部功能控制	1

①行为控制:是指控制构件的行为和输出结果的能力。被测构件在函数参数设置方面存在问题,在编写启动函数时没有将席位名、读写方式这些参数纳入函数设计。在首次的测试问题报告中可以总结出5个此类错误的现象。对该分支打分为6分。

②特性定制:是指支持构件内部特性的定制和配置的嵌入能力。被测构件的局部函数设置无效,说明在进行代码测试过程中程序员未对返回值进行全面地验证。在测试问题报告中可以总结出13个此类错误的现象。根据其错误类型的数量占评分标准的百分比来对该分支进行打分,得分为1分。

③构件执行控制:是指内嵌构件在测试模型、普通功能模型、控制模型等不同模式下执行的控制能力。由于被测构件是以通用软件支撑环境为基础,它在非软件框架的环境下只能执行部分功能。在测试问题报告中可以总结出8个此类错误的现象。根据其错误类型的数量占评分标准的百分比来对该分支进行打分,对该分支打分为2分。

④构件内部的功能控制:是指易被构件功能控制的内嵌能力。被测构件的内部功能存在使用冲突的现象,可以定位此错误为构件内部的功能控制较混乱。在测试问题报告中可以总结出10个此类错误的现象。根据其错误类型的数量占评分标准的百分比来对该分支进行打分,对该分支打分为1分。

2.4 构件可移植性

构件可移植性是笔者针对军队构件软件开发实际情况而提出的新的衡量分支,它的衡量准则包括4个方面,即:安装布置、构件环境控制、GUI控制、输入输出控制。通过实验,得出以下结论如表4所示。

①安装布置是指构件安装和配置的控制能力。

该软件的安装与配置,在首次测试中的表现还是不错的,开发方直接提出了安装包可以直接正常

表 4　作战编成可移植性结果

Table 4　Campaign organize replantability result

构件可移植性	分值结果
安装布置	10
构件环境控制	0
GUI 控制	3
输入输出控制	10

安装,且界面友好。对该分支打分为 10 分。

②构件环境控制是构件与外部环境之间的配置协调的能力。

作战编成构件的开发特点为机动性的通用支撑构件,它的使用应该可以脱离开共用软件平台,能够为今后开发的各类指控类软件提供服务,但是实际上在首次测试过程中发现,脱离平台,单独运行作战编成软件,有部分功能是无法实现的。说明该构件对环境的适应还没有达到独立运行、独立工作的能力。对该分支打分为 0 分。

③GUI 控制是在不同操作系统以及不同显示设备的情况下,自适应的能力。

在对该构件的测试过程中在这一性能方面做了相应的测试,准备了不同型号的计算机、不同型号的显示设备、安装的操作系统也不同,测试后发现,被测构件在安装 GUI 的控制方面基本没有,无法满足多种外设的实际使用需求。对该分支打分为 3 分。

④输入输出控制是在不同操作系统及显示设备的情况下,构件 I/O 能否正常使用能力。被测构件可以在不同操作系统及显示设备下使用,对该分支打分为 10 分。

2.5　构件测试支撑能力

构件测试支撑能力只能在构件被测试过程中才能验证和检查,并在构件验证过程中集中于支持测试操作的能力[6]。通过实验,得出以下结论:

作战编成构件在开发之初,在需求设计阶段没有将日后的测试支撑能力作为分析要点,因此该作战编成构件的测试支撑能力基本上为空,在测试过程中可以发现所以针对函数的代码级测试都是以开发方的开发角度来进行测试的,这也就加大了测试的难度,浪费了测试资源。由于各细分支得分均为 0,在这一分支上为该构件打 0 分。

2.6　针对构件的度量计算

本文定义了 5 条衡量特性来确定构件的质量,并进一步建立一套可测试性五边形法则来体现该模型的实际意义。以 5 条特性分别作为五边形模型的 5 个顶点,其中中心点为 o,它到 5 个顶点的距离为 oa, ob, oc, od, oe,它们的取值是通过对 5 条特性的具体指标打分而得到的,以 oa 为例,计算公式为:

$$|oa| = \sum(A_1 + A_2 + \cdots + A_i)/i$$

其中 A_i 为各打分细则的小计分数,i 为分支个数。

令 $D_a = |oa|$, $D_b = |ob|$, $D_c = |oc|$, $D_d = |od|$, $D_e = |oe|$。

根据表 1 至表 4 则有表 5 各值。

表 5　作战编成可测试性因素列表

Table 5　Campaign organize testability factors

可测试性因素	初始版本
可理解性	0.88
可观察性	0.99
可控制性	0.25
可移植性	0.58
测试支撑能力	0

以 $\triangle aob$ 为例,它的面积 S_{aob} 的计算公式为:

$$S_{aob} = 0.5 \times \sin 72° \times D_a \times D_b$$

则五个内三角形的面积之和为

$$S = 0.48 \times (D_a \times D_b + D_b \times D_c + D_c \times D_d + D_d \times D_e + D_e \times D_a)$$

S 即为该构件的可测试性得分,经过计算,初始版本的构件可测试性得分为

$$S_{初始版本} = 0.48 \times [(0.88 \times 0.99) + (0.99 \times 0.25) + (0.25 \times 0) + (0 \times 0.58) + (0.88 \times 0.58)] \approx 0.78$$

得出的结论在五边形图示上可以看出它的模型饱和度,如图 1 所示。

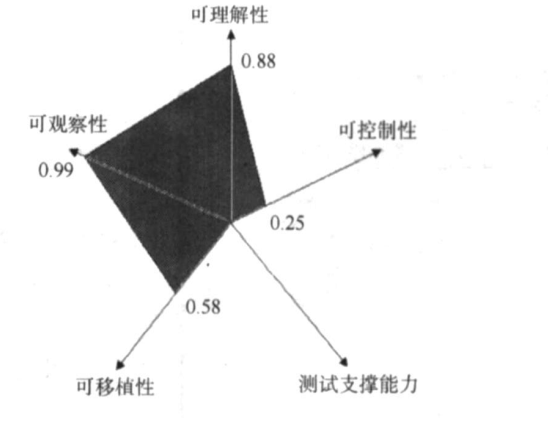

图 1　作战编成可测试性第一次结果

Fig. 1　Campaign organize testability result 1

将以上对该构件的缺陷描述以及得分情况提交给构件开发方,使之有针对性地修改并完善构件。

2.7 软件回归测试分析

针对之前所做过的测试工作,总结第一次的可测试性验证报告并提交给开发方人员,在经过 3 个月的修改完善之后,收到了该构件的二次回归被测件。

再次对其进行测试,与前一版本进行比较,得出以下几个方面的结论。

首先,在第一次所提交的可测试性缺陷报告中指出,在构件可控制性指标中存在很多问题,储如行为控制、特性定制等,开发方针对这一方面做出了很大的修改;

其次,在可移植性的修改方面,开发方把构件的运行环境需求与外层平台剥离开来,使二次开发的难度降低,同时使构件可以在各种操作系统上运行;

另外,在测试支撑能力上,也在部分重要功能的函数上增加了测试模式,在调试模式下,能够把返回值的代码显示给使用者,对测试工作的帮助很大。

总结二次回归软件,开发方并未单纯地将测试方提交的测试问题报告逐一改正,而是从文中所指出的五大类分支的角度上来完善软件。这样,一改以往开发方与测试方的机械对话方式,能够在开发高度上对开发工作做出良好地判断和纠正,达到了测试的真正目的。

经过第 2 次测试,得出了一套新的模型数据。如表 6 所示。

表 6 作战编成修改后可测试性因素列表

Table 6 Campaign organize testability factors modified

可测试性因素	修改后版本
可理解性	0.88
可观察性	0.99
可控制性	0.75
可移植性	0.8
测试支撑能力	0.5

使用五边形法则,得出如下结论

$$S_{\text{修改后版本}} = 0.48 \times [(0.88 \times 0.99) + (0.99 \times 0.75) + (0.75 \times 0.5) + (0.5 \times 0.8) + (0.88 \times 0.8)] \approx 1.48$$

得出的结论在五边形图示上可以看出它的模型饱和度如图 2 所示。

前后对比两个模型的饱和度不难发现,通过有针对性地提出修改意见,使开发方能够迅速在宏观上了解构件缺陷点,然后在不影响正常开发工作的情况下,适时地修改软件,这是一种新的开发方式,同时也减轻了测试人员的工作量。

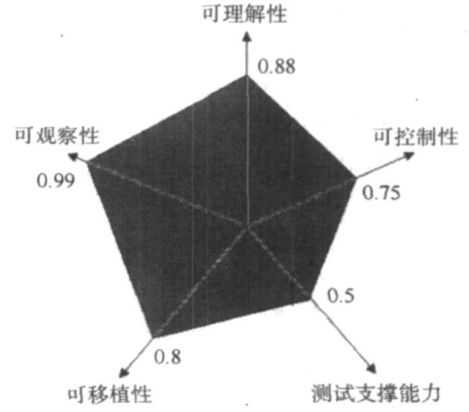

图 2 作战编成修改后可测试性结果

Fig. 2 Campaign organize testability result 2

3 结论

本文所描述的构件可测试性模型是针对军用构件化软件的特殊开发条件和环境而总结出来的,它将会有一个非常广阔的应用前景,文中所讨论的可测试性模型在一定程度上还有很大的不足,需要在日后不断完善,使它所提供的数据能够让开发人员迅速地掌握进一步的开发重点,弥补设计缺陷,以达到开发出健壮性更好的构件化软件。

参考文献:

[1] GAO J. Component testability and component testing challenges[D]. Washington: San Jose State University, 2005. [2007] http://www.engr.sjsu.edu/gaojerry/techreport/.

[2] GAO J. A component testability model for verification and measurement[C]//The 29th Annual International Computer Software and Applications Conference 2005: 2-8.

[3] ROY S F. Testability of software components[J]. IEEE Transactions on Software Engineering, 1991, 17(6): 4-5.

[4] GAO J, ZHU Youjin. Tracking software components[R]//Technical Report In San Jose State University, in Submitted for Publication: 1999: 2-3.

[5] VOAS J M, MILLER K W. Software testability: the new verification[J]. IEEE Software, 1995, 12(3): 17-18.

[6] Software engineering a practitioner's approach[M]. 4th. New York: ROGER S Pressman, 1997.

(下转第 105 页)

[C] //Third IEEE International High-Assurance Systems Engineering Symposium, 1998: 158 - 165.
[5] 许晓春,梅琳,胥光辉,等. EASTT 源码解析系列之二——中间结构管理器[J]. 共创软件,2002(1):35 - 38.
[6] 刘久富,孙德敏,杨忠,等. 嵌入式软件的动态测试[J]. 微计算机信息,2006,22(2):82 - 84.
[7] 晏华,袁海东,尹立孟. 代码自动插装技术的研究与实现[J]. 电子科技大学学报,2002,31(1):62 - 66.
[8] 钟治平,徐拾义. 程序插装技术在软件内建自测试中的应用[J]. 计算机工程与应用,2004(17):117 - 118.
[9] 唐科,汪文勇,刘利枚. 嵌入式软件覆盖测试的研究[J]. 成都信息工程学院学报,2005,20(5):541 - 545.
[10] 文昌辞,王昭顺. 软件测试自动化静态分析研究[J]. 计算机工程与设计,2005,26(4):987 - 989.

Application of XML technology in software test framework

XU GuangHui[1]　DU Yu[1]　LI Xiang[2]　XU YongSen[3]
(1. Institute of Command Automation, PLA University of Science and Technology, Nanjing, Jiangsu 210007;
2. East China Testing Center of Engineering Software, Shanghai 200233;
3. Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu 210093, China)

Abstract: The system framework of software test tool is analysed. Based on result of the analysis, a well-extended system framework is presented. In the presented system framework, XML file format is used for the Source Code Information File. Different Info-get Template File, which is in XSLT format, is used to get the application-related information. As the application extends, the system only need to extend the Info-get Template File without having to change the Source Code Information File.
Key words: software testing; system framework; XML; XSLT

(上接第 100 页)

Research and practice of component-based software testability model

FU Cheng[1]　GONG YunZhan[2]　HONG Hui[1]
(1. Institute of China Electronic Systems Equipment Company, Beijing 100039;
2. College of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China)

Abstract: This paper analyses the development of an military component-based software. Based on others' work, it defines a pentagon-based measurement rule for component-based software. This method can reflect the testability factors of the component being tested correctly, make the developers realize the defaults of that component, and it can also help developers to design components with well testability. Through the authors' test case experience, the pentagon-based measurement rule is proved helpful to increase the testability of the component-based software, and being certificated by both develop engineers and test engineers.
Key words: software component; testability; software test