ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.3, March 2009, pp.567−582
doi: 10.3724/SP.J.1001.2009.03443
© by *Institute of Software, the Chinese Academy of Sciences*. All rights reserved.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基础软件平台质量评估*

兰雨晴+, 赵 同, 高 静, 接 卉, 金茂忠

(北京航空航天大学 计算机学院,北京 100191)

Quality Evaluation of Foundational Software Platform

LAN Yu-Qing+, ZHAO Tong, GAO Jing, JIE Hui, JIN Mao-Zhong

(School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

+ Corresponding author: E-mail: lanyuqing@buaa.edu.cn

Lan YQ, Zhao T, Gao J, Jie H, Jin MZ. Quality evaluation of foundational software platform. *Journal of Software*, 2009,20(3):567−582. http://www.jos.org.cn/1000-9825/3443.htm

Abstract: The research on the software quality model and software quality evaluation model has always been a hot topic in the area of software quality assurance and assessment. A great amount of domestic and foreign researches have been done in building software quality model and quality assessment model, and so far certain accomplishments have been achieved in these areas. In recent years, platform building and systematization have become the trends of developing basic softwares based on operating systems. Therefore, the quality evaluation of the foundational software platform becomes an essential issue to be solved. This article analyzes and concludes the current development of researches on software quality model and software quality assessment model focusing on summarizing and depicting the developing process of quality evaluation of foundational software platform. It also discusses the future development of researches on quality assessment of foundational software platform in brief, trying to establish a good foundation for it.
Key words: foundational software platform; quality evaluation; quality attributes; quality model

摘 要: 关于软件质量模型和软件质量评估模型的研究,一直是软件质量保障和评估领域的研究热点,国内外在这两方面进行了大量的研究,并取得了一定的研究成果.近年来,以操作系统为核心的基础软件呈平台化、体系化的发展趋势,基础软件平台的质量评估成为亟待解决的问题.在总结、分析软件质量模型、软件质量评估模型研究发展现状的基础上,重点归纳和描绘了基础软件平台的质量评估发展历程,并简要探讨了基础软件平台质量评估研究的发展方向,力求为展开基础软件平台的质量评估建立良好的基础.
关键词: 基础软件平台;质量评估;质量属性;质量模型
中图法分类号: TP311 **文献标识码**: A

有很多学者、组织对软件质量评估的发展历史进行了较为全面的概括和总结[1−3].这些总结都比较完整地记录了较为早期的软件质量评估发展过程.然而,随着不断升温的网络化、平台化、服务化的趋势,传统的质量

* Supported by the National High-Tech Research and Development Plan of China under Grant Nos.863-317-01-04-99, 2006AA01Z186 (国家高技术研究发展计划(863))

Received 2007-07-18; Accepted 2008-08-07

© 中国科学院软件研究所 http://www.jos.org.cn

评估方法在新发展趋势中的适用性正受到挑战.本文以基础软件平台为对象,对其质量评估的研究和发展方面比较有代表性的工作进行考察和总结,归纳和描述了基础软件平台的质量评估发展演化历程,并试图提出自己的解决方案,力求为确定适用于基础软件平台的质量评估方法建立良好的基础.

本文从基础软件的定义开始,引入了基础软件平台的定义及现状,简要分析以前方法及其不再适用的原因,然后根据评估的特征,将针对基础软件平台的评估大体上分为两个阶段:分散阶段和集中阶段.根据每个阶段的研究成果对基础软件平台的质量评估做出总结,并提出展望.

1 基础软件平台

1.1 基础软件定义

根据信息产业部电子科学技术情报研究所的定义,基础软件又称为系统水平级软件、系统基础软件、管理软件,是指构建在各种应用软件基础之上,为各种应用提供顺利运行的软件.

除此之外,国内外还有一些企业也曾给出过各自的定义.Webopedia把系统软件归纳为以操作系统为中心,包含其他相关功能软件进行计算资源管理的软件环境;国际数据公司(IDC)定义的基础软件和开发工具包括Unix,Linux,Windows 等操作系统,数据库管理系统(DBMS)、安全、存储管理系统及应用开发工具;知名研究机构弗里斯特(Forrester)认为,基础软件是为企业应用软件提供必要的运行和支撑的软件,包含操作系统、数据库管理系统、中间件等;2004 年,IBM 公司提出了“整合从 IBM 软件开始”的口号,除了业界公认的中间件webSphere 外,IBM 把 DB2,Lotus,Tivoli,Rational 等数据库、办公、系统和开发工具都列入中间件的范畴,其定义的中间件是介于操作系统和应用软件的所有软件.但现在普遍接受的观点仍不是广义的中间件,数据库等产品并不包含在中间件中.目前,业界已把中间件列为基础软件.

1.2 基础软件平台定义

基础软件平台是指由操作系统、中间件、数据库、安全产品以及办公套件等通用性软件组建的一系列应用支撑平台.基础软件平台的质量指的是平台整体的质量,既包含平台内部的每一个构成软件的质量,又包含各构成软件之间交互的质量.

从根本上说,基础软件平台的作用是为了进一步解放软件生产力,提高软件开发效率,降低软件开发难度.每一个基础软件平台的诞生,都向软件生产力的提高迈进了一步.

从功能方面而言,基础软件平台本身还要起到如下基本作用:

1) 提供软件系统运行的支撑环境;

2) 提供相关接口,屏蔽技术细节;

3) 支持应用系统的开发、部署与管理等功能.

1.3 平台化发展现状

基础软件平台的核心是操作系统.目前 3 个主流的基础软件平台分别基于 Windows,Unix 和 Linux,经过长时间的发展在全世界范围内达到了一个微妙的平衡.本节主要介绍这 3 个主流基础软件平台,并对它们进行简要对比.

(1) Unix 平台

Unix 平台是由操作系统、数据库、中间件、系统管理、安全工具、开发环境等组成的软件环境,其体系组成如图 1 所示.

Unix 操作系统软件总是伴随着相关厂商所提供的服务器硬件的升级换代而不断变革着,也正是与硬件的紧密结合,才使其能够拥有久经考验的可扩展性、可靠性和易管理性,Unix 依然是不少企业创建关键业务 IT 基础设施的选择.网格、动态分区、自我管理能力等技术也在不断融入到新一代 Unix 操作系统之中.虽然 Unix 是开放的系统,但实际上,IBM,HP 和 Sun 三大厂商之间的产品兼容性并不好,都受限于相应的服务器硬件.当然,这种开放性在不断得到改善.Unix 操作系统不只是运行在自己的硬件平台上,也开始对多种硬件平台提供支

© 中国科学院软件研究所 http://www.jos.org.cn

持.Sun Solaris 不仅支持 Sparc 芯片的服务器硬件,还支持 x86 体系架构的产品.HP-UX 也从高端服务器扩展到安腾服务器平台.3 种 Unix 操作系统与 Linux 的互动性更是显示了开放的趋势.

不难看出,Unix 平台仍将是 3 个平台中最为开放的平台,也正是这种开放才能够保证用户不被某一个软件供应商锁定,各种新技术也能在开放的环境中接受考验.

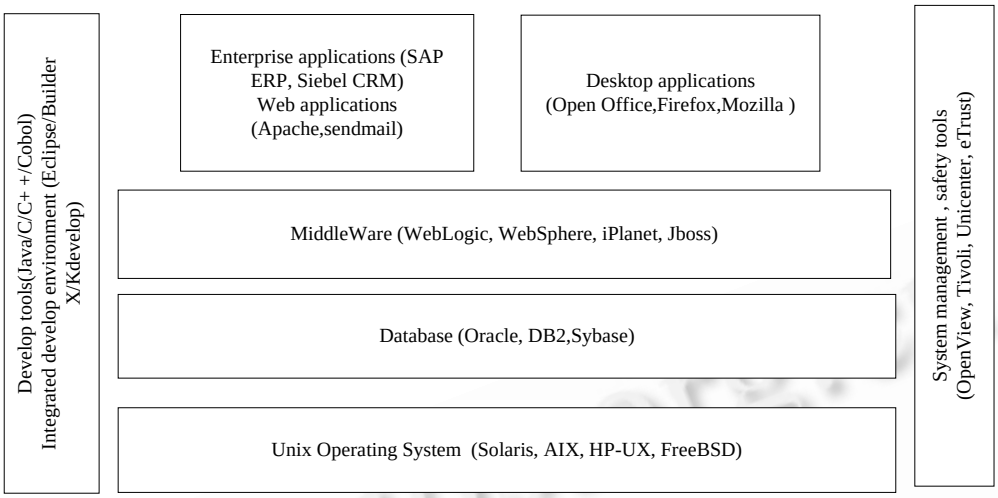


Fig.1 Unix platform

图 1 Unix 平台

(2) Linux 平台

这里以最著名的 LAMP(Linux,Apache,MySQL,PHP/Perl/Python)为例展开介绍.LAMP 是由开源操作系统 Linux、开源 Web 服务器 Apache、开源数据库 MySQL 以及开源编程语言 PHP/Perl/Python 所组成的一种软件应用平台.LAMP 提供了一个低价、高效的软件平台,并逐渐与.Net,J2EE 平台呈三足鼎立之势.

Linux 操作系统无疑是开源软件发展最快、市场应用较为成功的范例,具有免费、开源、多用户、多进程等特征.Apache 是一种开源 Web 服务器软件,目前在全球 Web 服务器市场的占有率达 60%.MySQL 是一个基于 MySQL 协议的开源轻型数据库系统,具有速度快、运行稳定的特点,且对于个人用户(不用于商业用途)免费.PHP 是一种服务器端的脚本语言,具有免费、开源、支持数据库操作等特点.PHP 可以和 HTML 混合编辑,是进行小型 Web 开发的理想工具.Perl 是一种借鉴 C/C++,Basic,Pascal 等语言中众多语法规则强大灵活的动态语言.同样,Python 也是一种扩展性较强的高级动态编程语言.Linux,Apache,MySQL 及 PHP/Perl/Python 遵循已广泛应用的开源协议 GNU/GPL,技术较为成熟,且用户范围广.

Linux 平台在服务器端的发展最为迅速,目前已经成为网站 Web 服务器市场的主流平台之一,这主要源于 Linux 的优异、安全和可靠的性能.同时,Linux 应用软件众多,嵌入式 Linux 也是当前发展的热点之一.Linux 在嵌入式领域的低成本、小内核以及模块化有着自己的特色,很多 Linux 厂商在该领域加大投入开展研发工作.桌面领域因为 Windows 地位的根深地固,加上 Linux 之上的应用软件欠缺,目前发展依然缓慢.

Linux 平台的开发工具普遍缺乏类似于 Windows 平台上的开发工具,应用软件的丰富程度与 Windows 平台想比相差甚远,再加上 Linux 桌面至今仍很少被使用等种种因素,都阻碍了 Linux 广泛使用.

(3) Windows 平台

WAMP(Windows,Apache,MySQL,PHP/Perl/Python)平台是在 LAMP 平台出现之后,为了更好地集成已有系统或各取所长搭建新系统而逐渐确立的一种软件平台.WAMP 主要由商业软件 Windows 和开源软件 Apache,MySQL 及 PHP 共同组成.2005 年前后,WAMP 平台开始逐步走入人们的视野.WAMP 平台的出现打破了以往业界对.Net 平台与 LAMP 非此即彼的观念.前一种选择更容易管理,但价格较贵;后一种选择具有成本低

© 中国科学院软件研究所 http://www.jos.org.cn

和安全性好的优势,但是系统复杂性更高.

Windows 操作系统在桌面端处于垄断地位,拥有庞大的用户群;在服务器端也占有 50%以上的份额,并且市场占有率正稳步上升.尽管 WAMP 平台比较复杂,但能够为企业提供较低的成本和更好的安全性.

Microsoft 从桌面开始构建 Windows 平台,正在逐步跨越服务器、嵌入式领域;并以其桌面环境为基础,以其简单、易用的开发工具、无缝集成的开发环境为依托,通过其完善的培训、认证体系,吸引众多的独立软件开发商、开发人员、硬件供应商、系统集成商开发 Windows 平台上的应用软件,来满足社会各领域、各层次的用户需求.

Microsoft 总是企图自建一套完整软件标准体系、规范,尤其是在底层技术、基础软件、开发工具方面,并且这种技术标准很少公开.Microsoft 的这种战略不但有效阻止其他竞争者的加入,而且牢牢把握着对 Windows 平台的控制权.Microsoft 为获得控制权,常用的策略主要有两种:一方面,完善自己的产品,同一产品不断更新总会增加用户期待的新功能;另一方面,不断延伸产品线,使自建平台的内容不断丰富,防止其他产品的入侵.IE 浏览器把 Netscape 挤出 Windows 之外,并最终赢得几乎整个浏览器市场便是明显的例证.

(4) 三大主流平台对比

根据兰科(RANC)2007 年调查结果的简要分析,可得三大主流平台对比,见表 1.

Table 1 Comparison of three platforms

表 1 三大主流平台对比图

Field	Windows	Unix	Linux
History	15 years (26 years from DOS)	38 years	16 years
Application	server, desktop, embedded	server, desktop	server, desktop, embedded
Dominant field	desktop	server (commercial application)	server(Networking application)
Advantage	monopolistic state of desktop; simple, easy to use; ample applications, preferable development tools; perfect training system and certification system; ISV support, plenty of technologist;	mature system, safety, reliable; steady performance; many commercial applications; support multiple programming languages;	Open source; Low costs; Government support;
Disadvantage	Shortage of openness to platform Low performance safety issue	Differences among releases; High price of the hardware;	high learning cost; lack of applications; compatibility problems; Less technologist;
Chance	Commercial applications; Embedded applications; Networking applications;	Make current commercial customers steady; Expand to Desktop;	Networking applications; Embedded applications; Expand Desktop applications;
Threaten	Impact from network applications; importance weaken by Web service; move to Linux platform;	transplant to Linux/Windows;	Open Source commercial Unix; Windows open source; dispute about intellectual property rights ;

1.4 平台化研究厂商

由于基础软件平台在软件产业链的基础地位和主导作用,国内外已有不少厂商投入到平台化的研究当中.这里对国内外比较活跃的平台化研究厂商作简要介绍.

(1) 国外平台化研究厂商

国外两家厂商最先开展平台化工作研究,并取得了一定的成果.一个是 SpikeSource 公司,SpikeSource 公司为企业应用提供了面向 Web 应用的 Linux+Apache+MySQL+PHP/Java 架构的开源基础软件平台;另外一个是全球领先的企业级开放源代码解决方案供应商红帽公司,它于 2006 年年初开始提供经过兼容性测试的开源基础软件和构件套件 Redhat Application Server 的日常支持,以简化和标准化开源应用软件的开发,这将使开发人员得以集中精力在自己的应用而不是基础软件平台的配置上.两家都在通过测试实践验证自己的质量评估体系,并不断完善和发展.

随着 2008 年 Oracle 正式收购 BEA,Oracle+BEA+SUN 系统平台也正式加入平台化的领先队伍.其平台构成如下:Oracle Database 10g,BEA Weblogic Server 9.0,Solaris 10,硬件环境则是基于 SUN Fire E6900.此平台的

© 中国科学院软件研究所 http://www.jos.org.cn

推荐方案主要为大中型建设行业企业部署和运行工程项目管理系统提供高可用、稳定的支持平台.另外,此平台能够简单部署并具有较强的管理特性,帮助降低成本,完全满足大型、集团型企业的应用需求.

(2) 国内的平台化研究厂商

在过去的几十年间,我国以操作系统、数据库和中间件为核心的基础软件关键技术取得了重要进展,桌面应用软件也取得重大突破,基本形成以操作系统、数据库、中间件等为核心的自主国产基础软件平台体系.

在市场推动以及国家计划项目的推动下,很多厂商纷纷加入了平台化研究.本文依据现有的工作基础、国家计划项目支持、产品成熟度、市场占有率等因素,分别在国产操作系统、数据库、中间件、办公软件产商中选取贡献比较突出的,见表 2.

Table 2 Main enterprises in the field of domestic foundational software platform

表 2 国内平台化研究厂商

Product type	Product name	Manufacturer
Operation system	NeoShine Linux Server,Desktop operation system kylin server operation system Red flag Linux Server,Desktop operation system	China Standard Software Co. National University of Defense Technology Red Flag Software Co.
Database	KingBase database DM database OpenBase database OSCAR database	KingBase DM Neusoft BJSASC
Middleware	CVIC Middleware Software StarBus TongWeb	CVIC Software Engineering Co. National University of Defense Technology Tong Tech
The application of office software	Evermore Office Red Office WPS Office NeoShine Office	Evermore Software REDFLAG2000 Kingsoft Corp. China Standard Software Co.
Intelligent report system	RAQ Report System BI-Pilot Commercial intelligent development platform	RAQSOFT Bi-Soft
Safety product	CA Security Authentication System/PKI digital sign, uniform safety authorities System safety and management VPN/Firewall	Beijing Creative Century Information Technology、National Cybernet Jilin University Information Tech,Wellhope NUDT,Information Engineer University Huadun,Bright Oceans

2 传统质量评估方法

由于专门针对基础软件平台的质量评估方法研究刚刚起步,所以,大部分的用户和厂商仍然在使用传统的质量评估方法对各基础软件进行评估.而对于基础软件平台,尚没有有效的质量评估方法.通常的做法是将平台里的基础软件分别评估,最后进行质量的简单平均.这里我们列举了迄今为止,基础软件厂商最常用的评估方法,并通过分析基础软件平台的特征,指出这些方法在基础软件平台评估方面的局限性.

国内外对于软件质量改进和评估的研究,主要集中在两个方面,一个面向最终产品,另一个面向过程.

2.1 质量的定义

有许多种关于软件质量的定义.在 RUP 中,质量的定义包含以下 3 点[4]:

1) 满足或超出认定的一组需求;

2) 使用经过认可的评测方法和标准来评估;

3) 使用认定的流程来生产.

在中华人民共和国国家标准-计算机软件质量保证计划规范 GB/T 12504-90[5]中,软件质量是指软件产品中能够满足给定需求的各种特性的总和,这些特性称作质量特性,包括功能性、可靠性、易使用性、效率、可维护性和可移植性等.在 ANSI/IEEE Std 729-1983[6]中,软件质量是指与软件产品满足规定的和隐含的需求能力有关的特征与特性的全体,具体包括 4 点:

1) 软件产品质量满足用户要求的程度;

2) 软件各种属性的组合程度;

© 中国科学院软件研究所 http://www.jos.org.cn

3) 用户对软件产品的综合反映程度;

4) 软件在使用过程中满足用户要求的程度.

ISO9000(2000 版)里把质量定义为一组固有特性满足要求的程度.朱少民则认为软件质量是由 3 部分组成的[1]:

1) 软件产品的质量,即满足使用要求的程度;

2) 软件开发过程的质量,即能否满足开发所带来的成本、时间和风险等要求;

3) 软件在其商品环境中所表现的质量.

当然,还有许多人也对软件质量给出了相应的定义.例如,Pressman[7]认为软件质量是符合明确陈述的功能和性能需求、明确文档化了的开发标准和所有专业开发软件预期的隐含特性;SEI 的 Humphrey[8]则将软件质量定义为“在实用性、需求、可靠性和可维护性等方面,达到了优秀的水平”等等.

也有人通过研究得出,任何技术系统的基本质量需求为 3 个[9]:功能性——正确性、可靠性等,性能——如反应时间、输入输出、速度等,经济——花费有效性.

通过以上定义容易看出,虽然对于质量并没有一个统一的定义.但是,绝大多数的组织机构或个人都认为“满足需求”是软件质量最重要的一个方面.其次就是软件的无故障性、实用性、性能或者效率等.

2.2 面向产品的质量评估

人们通常用软件质量模型来描述影响软件质量的特性[2].

不同质量评估方法的区别通常表现在:评估的过程模型和质量模型上.大部分的评估过程模型都遵循 ISO/IEC 14598 标准提供的软件质量评价过程.而对于其余的少数评估过程模型,由于篇幅限制,这里只介绍对基础软件平台有参考价值的集成软件评估模型,如熊鹏程、范玉顺提出的模糊层次分析法[29].层次分析法(analytic hierarchy process,简称 AHP)[21,30]是一种在多指标情况下决策的方法,常被用于按照相关目标来分析计算各指标的权重.此方法虽然比较简单、易用,但是没有考虑到将个人的主观意志映射到具体数字时的不确定性.由于这种度量因子的模糊性和不确定性,按照传统 AHP 方法获取的权重数据不全面而且不准确.因此,熊鹏程、范玉顺提出了模糊层次分析法,拟引入模糊规则,用模糊区间来替代具体的权重数值,通过相关计算最后获得相应权重,提高了获取相关质量特性度量因子的准确度.

由于评估的过程模型大体上来说比较统一,质量模型就成为了影响评估结果的关键.现有许多种有关软件质量的模型,本文重点介绍影响比较大、应用比较广的模型.其中,主要介绍了 7 个层次型质量模型和 1 个关系型质量模型.在 7 个层次型质量模型中,1)~4)分别与国内外的质量标准相关;5)是企业自定制模型;6)和 7)则是随着新出现的技术和软件形态而出现的较新的质量模型.

1) McCall 质量模型

此模型由 McCall 等人于 1979 年提出[10].它的质量由 11 个特性构成,这 11 个特性又分为产品操作、产品修订和产品转移 3 类.McCall 模型对于每个质量要素都给出了衡量标准,共 43 个.而每个软件质量衡量标准又可以有多个不同的度量.如对于软件的简洁性可以用许多不同的度量来确定其复杂度.比较常用的有 McCabe 的基回路数、Halstead 的度量、程序代码行数等等.

McCall 的质量度量把质量因素的各度量标准进一步分解成一些条款,根据一个软件是否满足条款来对软件进行评分.大多数情况下,满足条款时+1 分,不满足时+0 分.

2) Boehm 模型

Boehm 模型[11]由 15 个质量要素构成,与 McCall 模型的产品操作质量要素基本相同,且均没有给出质量要素之间的关系.两者唯一的差别在于特征的种类(Boehm 模型包括 McCall 模型没有的硬件生产力的质量特征),所以它们具有类似的优缺点.例如,它们都没有从软件生存周期不同阶段的存在形态来考虑,仅仅考虑一种产品形态,不利于软件产品早期的缺陷发现和维护成本的降低[3].

3) ISO/IEC 9126 模型

国际标准化组织制定了国际标准 ISO/IEC9126《软件质量模型》[12],这个质量模型被广泛引用.它定义了外

© 中国科学院软件研究所 http://www.jos.org.cn

部和内部质量的质量模型,将软件构件质量属性划分为 6 个特性(功能性、可靠性、易用性、效率、维护性和可移植性),并进一步细分为若干子特性.

另外,ISO 建议软件质量模型由 3 层组成:

- 高层:软件质量需求评价准则(SQRC);
- 中层:软件质量设计评价准则(SQDC);
- 低层:软件质量度量评价准则(SQMC);

这种分层模型被广泛采用,很多的质量模型都是确定质量需求之后,再进一步分层来细化各评价准则.

4) 上海软件中心(SSC)的软件质量度量模型[13]

在 SSC 模型中,设置了 6 个质量特性和 22 个质量子特性.其中,它的质量特性与 ISO/IEC9126 相同,质量子特性则大部分是参照了 McCall 模型定义的.

5) 企业自定义模型

这里主要列举两个.惠普公司侧重于 FURPS 标准,即[14]:功能性(functionality)、可用性(usability)、可靠性(reliability)、性能(performance)、服务性(serviceability).IBM 公司则利用 CUPRIMDSO[14]的层次来监控对软件产品的满意度,即功能性(capability)、使用性(usability)、性能(performance)、可靠性(reliability)、可安装性(installability)、可维护性(maintainability)、文档/信息(documentation/information)、服务(service)以及综合性(overall).

6) 面向对象的质量模型

随着面向对象技术的迅猛发展,形成了一些面向对象系统的质量模型,较为典型的是 Bansiya 在 1997 年提出的 QMOOD(quality model for object-oriented design)模型[15],这是一个设计层的面向对象系统的质量模型,主要针对面向对象系统的外部高层质量特性,如可复用性、灵活性、功能性等.后来,他又在 QMOOD 的基础上提出了面向对象层次模型的修改模型.此模型将面向对象系统分为 4 个 level.每两个相邻的 level 之间有一个相应的连接 link,共 3 个.其大体思想是先从高层到低层确定质量属性,然后再通过实践活动,逆向得到最终的产品质量.它的基本质量属性是以 ISO/IEC9126 的 6 个质量属性为原型改造而成.

7) 软件构件质量模型

软件复用已经成为软件工程的主要研究课题之一,被认为是使软件开发真正走上工程化和产业化道路的希望.基于构件的软件开发(component-based software development,简称 CBSD)是软件复用的有效方法.随着可复用构件数量的不断增加,构件的质量将影响到最终的软件质量和 CBSD 的推广应用.但是,目前缺乏对构件质量有效的评价模型,用户在选择构件时缺乏有效的度量参考[16].

1997 年,曾有专门的委员会致力于 IEEE 软件构件质量标准的制定[17],然而这一计划最终被搁浅,因为该委员会达成共识:在当时要推出一个可作为标准的候选草案为时尚早.

我国软件构件标准工作组中的软件构件质量标准项目组成立于 2003 年 10 月,该项目组由北京航空航天大学软件工程研究所(简称北航软件所)、北京大学信息学院软件所、长城软件公司和上海软件中心 4 家单位组成.软件构件的质量模型《软件构件产品质量》[18],是在综合研究 COM/DCOM/COM+,CORBA,EJB 等软件构件质量特性的基础上,抽象出软件构件的基本质量特性和特定质量特性.此标准是信息产业部的第一个有关软件构件质量的电子行业标准,它对软件构件质量保证和评估提供了规范.梅宏等人开展了保障软件构件服务质量的模型的研究,并提出了一种构件运行时可信度量模型[19].Lee 提出一种定量的软件质量评估模型来度量基于构件开发过程的过程制品[20],该模型采用计算质量特性的权重和 AHP(analytic hierarchical process)[21]技术对构件软件进行评估,该模型有助于开发出高质量的构件软件.Ma 等人认为了由第三方构件认证组织对软件构件进行认证,才是可信安全的方法.他们给第三方构件软件认证组织提供了一个测试软件构件功能的框架[22],框架定义了构件开发商需要提供的测试数据,以及第三方利用这些数据测试构件软件的过程.Goulão 和 Abreu 提出的构件质量模型[23]与 ISO/IEC 9126 模型基本类似,两者的最大区别在于质量子特性.

© 中国科学院软件研究所 http://www.jos.org.cn

8) 关系型质量模型

上面所列举的 7 种模型均为层次模型.这些模型的共同点就是由质量特性作为基本构成,质量特性可以由子特性来定义和度量,每一个子特性又可以使用不同的度量元.另外一类软件质量模型是关系型质量模型,该类模型一般定义各种质量特征之间的关系,比如:正相关、负相关和无关.典型的关系型质量模型有:Perry 模型[24]和 Gillies 模型[25,26].但该类模型也只定义了质量特征的两两关系,且没有定义质量特征的测量方法.

更多的软件模型之间的对比,请见 Kemerer[27]和 Rubin 的文献[28].

2.3 面向过程的质量评估

软件质量的一个重要特征是,至少在理论上,它能在软件开发的每一个阶段被测量.Cerino 曾讨论过在每个阶段的质量测量活动[31].

现有的面向过程的质量评估主要是通过可靠、可信、成熟的软件过程来保证软件质量,通过对软件过程能力成熟度的评估来推断软件质量.著名的有美国卡耐基梅隆大学软件工程研究所(CMU/SEI)提出的软件能力成熟度模型(简称 CMM)[32],已经成为软件开发组织改进开发过程的重要参考模型.CMM 包括 6 个模型、5 个等级、18 个过程域、52 个目标和 300 多个关键实践.在 5 个级别中,除了初始级外,每个成熟度级别都包含若干个关键过程域(key process area,简称 KPA).

CMMI 的阶段式类似于 SW-CMM,将所有的过程域按 5 个成熟度等级来组织,从低到高分别为:初始级(initial,第 1 级)、已管理级(managed)、已定义级(defined)、量化管理级(quantitatively managed)和优化级(optimizing,第 5 级).整体来看,这样的组织方式除了名称略有变化之外,各个成熟度等级代表的能力成熟度与 SW-CMM 对应等级相比都是大同小异的.CMMI 共有 22 个过程域,虽然大部分的过程域都有所变化,但主要的改动集中在等级 3 的过程域中.这一层新增加了 4 个过程域,与原来 SW-CMM 等级 3 中的过程域一起进行了重新划分,形成了一个更细致、更实际、更符合软件工程发展方向的成熟度等级[33].

软件质量保证(software quality assurance,简称 SQA)是 CMMI 可重复级的一个 KPA.软件质量保证的整个过程[34]包括:

- 制定软件质量保证活动计划.
- 检验软件产品和活动是否按照合适的标准、步骤和需求运作.
- 通过实际软件产品和活动的相关指标和理想指标的比较来监控及调整软件开发活动和软件质量目标以满足用户及最终用户对软件产品的需要和期望.

2000 年底,国际标准化组织发布了 ISO 9000 国际标准,该标准鼓励在制定、实施和改进质量管理体系(QMS)时采用过程方法.与此同时,很多人热衷于净室软件工程[35](clearnroom software engineering),这是把软件生产过程放在统计质量控制下的软件工程管理过程.其特点是劳动质量管理、重视生产过程和定量分析.

IEEE 的 Daskalantonakis 通过在摩托罗拉公司的实践,提出一种在整个公司范围内度量的软件测量方法[36].这是一种多维化的方法.它使用 G/Q/M 方法来实施测量,并强调整个过程的全局项目控制.Zubrow 则在文献[37]中给出了如何着手现今的软件质量测量的建议,并且将 ISO 质量标准与 CMMI 联系起来.它将 ISO 要求的质量目标一步一步地细化,最后用表格的方式提出具体的问题,这些问题均针对 CMMI 具体的活动过程.

美国国家标准《项目管理知识体系指南》中则认为:项目质量管理过程[38]包括保证项目满足原先规定的各项要求所需的实施组织的活动,即决定质量方针、目标与责任的所有活动,并通过诸如质量规划、质量保证、质量控制、质量持续改进(如适用)等方针、程序和过程来实施质量体系.

在过程中很好地控制度量评估的实践活动是很难做到的.关于软件工程实验或实践中的数据收集等方面的内容可以参考 Basili 和 Weiss 的文献[39].

2.4 现有方法的不适用性

综上所述,近 50 年来,软件质量评估技术得到人们的广泛研究,其研究成果主要集中在评估单个软件或软件构件等方面,基于集成软件或面向对象系统的评估也有一些文献报道.也就是说,国内外在单个软件产品质量

© 中国科学院软件研究所 http://www.jos.org.cn

和软件产品过程质量评估的研究方面形成了相对成熟的理论和实践体系.这些方法和理论经过了多年的发展,确实能够在一定程度上评估基础软件平台,而且现在仍有相当数量的用户和厂商在用这些方法进行评估,但是,这些方法在评估基础软件平台时都显然存在各自的缺陷和不足.

1) 面向产品的质量评估方法

传统的单软件评估方法在实践中使用很多,发展也比较成熟,但是它并没有针对平台的特性,其研究对象只是单个的软件,而不是由多个软件集成的平台,因此也就并没有将平台中构成软件之间的交互特性纳入到评估过程当中,评估结果不能如实地反映该平台的质量情况.

集成软件质量评估模型考虑到了多指标情况下的决策,但是对于如何求权重,还没有一个科学的、广泛被认可的方法.

软件构件的评估方法用来评估软件构件的质量,它关心的是该构件对其所在软件质量的贡献,而不是该软件的整体质量.并且,暂时还没有已证明合理性的算法来从构件质量的优劣得到其所服务的软件的质量优劣.所以,对于基础软件平台来说,并无大的参考价值.而基础软件质量评估的研究成果,则可以应用到软件构件的评估当中,促进构件评估的发展.

现在普遍采用面向对象来开发,面向对象系统的质量模型用于评估时有一定的优势.但它针对的只是面向对象的特性,与平台特性的关系不大.

2) 面向过程的质量评估方法

这些方法确实可以在一定程度上评估基础软件平台开发过程的优秀程度.但是,基础软件平台有其自身最重要的特殊性:它是由多个厂商的产品组成的,所以其开发过程很可能分布在不同地区的不同厂商之间.如何从地理分布不一致的各厂商得到精确、持续的度量数据,成为最大的难题.

保证了各厂商的开发过程,只能从一个方面说明该软件的质量;暂时没有研究结果表明,多个软件的集合体的质量为这些软件的质量累加.所以,这并不能保证整个平台的质量.

基础软件平台的开发过程是延续的,并不是完成了某一版本就停止了开发过程.所以,传统的过程评估方法很难解决开发过程的地域性、延续性以及收集数据困难等不利因素.

3 基础软件平台质量评估

由于基础软件平台是一个兴起不久且在不断发展的事物,对它的评估也正处于不断摸索中.我们将其评估分为两个大的阶段:分散阶段和集中阶段.其中,分散阶段指的是在基础软件平台评估的初期,评估需求凸显,但是由于厂商和用户均没有很强烈的意识将其作为一个整体,所以,产品大都处于分开评估的阶段.在此阶段,当用户挑选一个包含操作系统、数据库、中间件的平台时,最初的做法通常是分别评估 3 类软件,挑选出每类当中相对较优的产品.集中阶段出现的时间并不长,但是已有很多组织和学者正着手这方面的研究,有部分成果已经公开发表.

3.1 分散阶段

在这个时期,大部分的厂商或用户都是根据需求对基础软件平台中的某个构成软件进行单独的评估.由于操作系统在平台中的独特性,对于操作系统的评估又是研究最多、相对成熟的.针对操作系统的评估一直以来就没有停止过.

在共同准则的官方门户网站(www.commoncriteriaportal.org)中,列出了全世界范围 42 个成功获得认证的操作系统.对操作系统的评估成为评估机构的重要业务,许多机构致力于此,如 atsec 在 2006 年对红帽 Linux 4 Update 2 CC EAL3 开展的评估.

© 中国科学院软件研究所 http://www.jos.org.cn

Table 3 User requirement for Linux productions (%)

表 3 Linux 产品用户需求 (%)

Subdivision of sample group	Reliability	Safety	Accessibility	Efficiency	Compatibility	Management	Expansibility	Portability	Other
Server side	94.2	92.1	14.3	15.8	36.3	19.3	15.3	8.5	4.2
Desktop side	72.0	58.4	80.1	10.3	33.8	17.1	8.6	7.8	12.2

在中国,“赛迪顾问”的数据统计出 2006 年中国各 Linux 软件产品的用户需求,见表 3[40].

更多的针对操作系统的评估集中在操作系统的某个具体特性上.其中,最为显著的是安全性和互操作性.在操作系统的安全性评估或安全操作系统方面,国内外都有许多机构一直从事相关研究,如南京大学.还有许多学者给出了自己的研究成果[41−43].互操作性是指一个软件系统与另一个软件系统互相间具有接收、处理并共享所发送信息的能力[44].随着开放标准和互操作性逐渐成为软件工业发展的核心问题,许多研究人员对互操作性做出了深入的研究.如文献[45]中针对开源软件互操作性的探讨,以及文献[46]中对互操作性和兼容性的介绍.

由于服务质量的管理已经成为对中间件平台及其分布式系统的一个很重要的质量需求,所以对中间件的评估则大都集中在 QoS(quality of service)上.

所谓服务质量 QoS 是指服务性能的聚集效应,它反映了用户对特定服务的满意程度.早期的 QoS 工程,企图集中于标准中间件的单一类别的扩展[47].近年来,出现了一些通用中间件框架来促进多种类的 QoS 管理.经验表明,这样的框架并不能很好地实现 QoS 管理.因此,文献[47]中提出了一个全面、模型驱动的开发过程来加强中间件平台顶端的分布式应用的服务质量.具体做法建立在 OMG 的模型驱动架构上.

文献[48]中提出了一种应用层的基于中间件的 QoS 管理模型,研究端系统的 QoS 和资源管理.该框架允许用户和应用对 QoS 管理策略和适应性调整策略进行应用级的定制.该文在资源管理中提出了通用资源接口及其具体实现方法,并从应用层的角度,在网络层和操作系统上提出通用接口应具备的功能,以及如何与 QoS 管理器协调工作.

3.2 集中阶段

在集中阶段,厂商和用户开始逐渐意识并关注平台作为一个整体的特性.评估的对象不再仅仅局限于单个的构成软件,而是基础软件平台整体.

1) 国外研究现状

SpikeSource 公司和红帽公司都有自己完整的质量评估体系,在对自己提供的平台产品进行评估的同时,不断完善和发展各自的评估体系.比较著名的有 BRR 模型.

开源软件的迅猛发展,对开源软件的质量评估体系提出了新的挑战,2005 年由 SpikeSource 公司、O’Reilly CodeZoo、Intel 公司及 Carnegie Mellon 大学共同提出了开源软件评估模型 BRR(business readiness rating for open source)[49]计划,该模型主要评估开源软件,尤其是由开源软件组成的软件包的质量,其评估结果将软件进行了分级,共有 1~5 个等级,其中 1 表示不可接受,5 表示优秀.BRR 将软件评估分成 4 个阶段:

- 快速评估阶段.主要确定要评估的构件列表,设计一些基本问题来快速测量每一个构件,并排除那些不满足用户需求的构件.
- 目标使用评估.根据重要性给软件的使用分类,共 12 类,1~12 重要性逐步降低.对前 7 个(或更少)种类设定一个度量的重要性百分比.
- 数据收集、处理阶段.
- 数据转换阶段.

BRR 还给出了可供选择的评估初始化种类,共 12 种.见表 4.该模型目前还在发展当中.

Enrico Johansson 认为[50],软件平台是产品线的基础,具有非常重要的地位.早在 2001 年,他就开始着手对于软件平台的质量需求进行调查.在调查过程中,他主要选取了两个大型软件开发组织和 34 个参与者.为了使结果更加合理可靠,他使用 IPC(incomplete pairwise comparison)方法来分析实验数据.其研究结果表明,对不同涉众来说,软件平台的质量需求是不一样的.软件平台可能关注质量的不同层面,例如可维护性、效率等.他还指出,

© 中国科学院软件研究所 http://www.jos.org.cn

由于软件平台对其支撑的产品有很大的影响,其质量远远重要于单软件的质量.同时,要在软件平台中达到这个质量要求需要付出相应的代价.

Table 4 Evaluation categories of BRR

表 4 BRR 评估种类

Evaluate species	Description
Functionality	How does the software satisfy the requirements of customers?
Usability	Is the user interface friendly? Could the end-user easily use the software? Is the software easy to install, configure, develop or maintain?
Quality	How is the quality of design, coding, testing? Are the three processes complete? Is it of zero defects?
Safety	How about the ability of the software to handle the safety issue? What safety level can it get?
Performance	How about the performance of the software?
Expansibility	How about the expandability of the software relative to a broad environment?
Architecture	Is the architecture of the software excellent? How about it's modularity, flexibility, expansibility, openness, integration?
Support	How about the support for software component?
Document	How about the document of the software?
Adopted degree	How many communities, agorae or industry organization has adopted this component?
Community	How about the scope of the community relative to the software? Is the community active in a wide range of fields?
Professional	Is the development process and development organization professional? How about the level of them?

2) 国内研究现状

近年来,在中国国产基础软件产业蓬勃发展的同时,因国产基础软件间缺乏适配而导致国产基础软件平台出现了可用性、可靠性、可维护性等方面的问题,无法满足各行业领域的企业级应用系统的需求.在这样的背景下,科技部、信息产业部等国家相关部委组织国内各方面的力量,利用已有科研成果,联合攻关,进行国产基础软件间的适配和优化,构建基于国产基础软件核心产品的国产基础软件平台体系,旨在形成以国产基础软件为核心的国产基础软件平台体系解决方案,提高国产基础软件产品在集成环境下的可用性、可靠性、适应性及规范性,进而为国产软件的发展营造一个公共的研发、集成与应用推广环境,完善和健全产业链.

科技部在“十一五”863 计划中也安排了专门的探索类课题,研究国产基础软件平台的质量评估模型.目前,国产基础软件平台的质量模型初步建立,其度量等也在不断地反复验证和改进中.这对促进国产基础软件和国产基础软件平台在信息化建设中的应用,以及基于国产基础软件平台的企业级应用具有重要意义.其研究成果中的质量模型如图 2 所示.

在这个模型确定的过程中,该课题组提出了一系列的方法来确保结果的真实性.其中,改进的德尔菲法被用来确定可能影响国产基础软件平台的度量因子;大量的企业真实数据被收集并作了简要的去噪处理;然后,逐步回归法[51]被用来确定最终对平台质量影响显著的因子,从而形成质量模型;最后,该课题组从两方面验证了结果的可信度:第一,通过主成分分析[52],设定阈值来计算并筛选对国产基础软件平台质量影响显著的因子,将此结果与逐步回归的结果进行对比,说明了结果的可靠性和方法无关性;第二,该小组将自己的质量模型应用于部分国产基础软件企业进行评估,将评估结果与同环境下的专家评估结果对比.

3) 小 结

当前,对基础软件平台的研究远远低于其需求.很大程度上是由于这种新的软件形态与以往的传统软件形态有很大区别,其独特性使得研究者很难找到一种方法能在各种情况下优于其他各种算法.

然而,简要分析以上模型可知,虽然对基础软件平台作为整体的评估尚处在起步阶段,但国内外都已有组织和机构在开展工作.并且,随着基础软件平台的影响的增大,对其重要性认识的加深,我们相信在今后一段时间内会有更多研究成果和评估实例发表.

© 中国科学院软件研究所 http://www.jos.org.cn

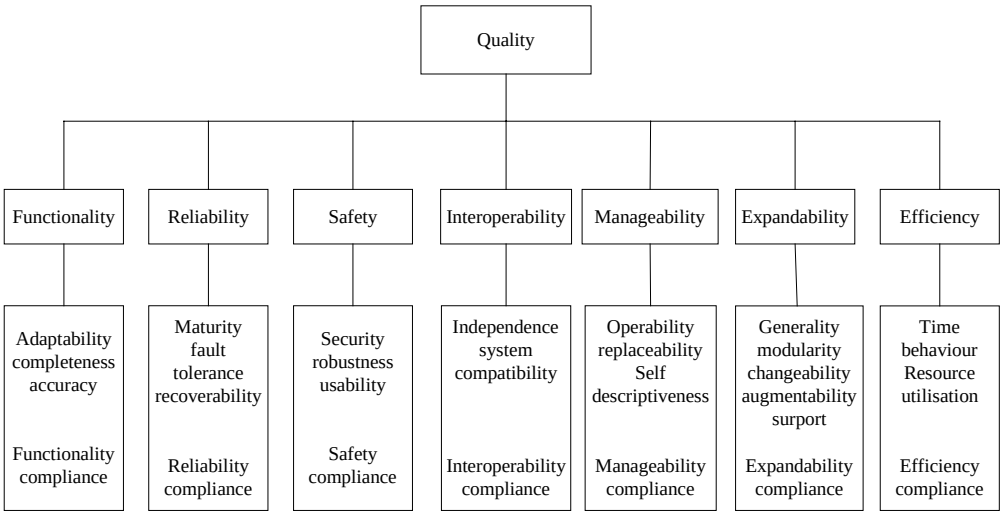


Fig.2 Quality model of domestic foundational software platform

图 2 国产基础软件平台质量模型

3.3 一个实例

2004 年底开始,中国政府出台了采购政策《国务院办公厅关于地方人民政府使用正版软件的通知》(国办函[2004] 41 号),强调正版化,推广国产化.在这个过程中,国产基础软件占了很大比率.然而,各级政府使用这些软件时,却通常遇到这样一个问题:在每个单个软件测试通过的情况下,整个平台的质量和性能却未能达到期望值,尤其是在互操作性、可靠性和可维护性等方面出现问题.这使得国产基础软件平台很难满足各行业领域的企业级应用系统对国产基础软件支撑环境的需求.实际上,这就是我们所说的,在基础软件的质量得以保证的前提下,如何来评估整个平台的质量.

问题出现以后,国家科技部等有关部委将基础软件平台的适配和评估定位为一个重点研究方向,并投入经费支持这个方向的项目.笔者也对此课题展开过研究.在研究过程中,调查了包含政务、计生、卫生、农业、教育等在内的 5 个领域的使用者,发现大部分调查者都选择使用本文第 3 节中的方法,将平台中的构成软件分开评估;也有部分单位使用第 4 节中的方法.研究表明,使用这些方法之后,他们对于自己所选或即将选择的平台质量有一个大概的了解.但同时,他们也指出,在短期内很难得知这些方法得出的结果是否可信,也缺乏大量的工程实践来证明这些方法对于基础软件平台评估的合理性.并且,从质量需求到具体的度量数据收集,怎样合理地以整个平台为主体,也需要进一步研究.

例如,我们曾对北京市某区政府进行调查,该单位购买的是红旗桌面操作系统 Redflag Linux Desktop (DT)5.0,人大金仓数据库 KingbaseES 4.1,中间件是东方通的 TongWeb 4.0,办公套件则使用的永中科技的 Evermore Integrated Office 4.1.该单位在使用此平台的过程中曾遇到不少问题.由于无法在购买之前详细了解该平台的质量,当业务应用系统遇到问题时,通常无法定位是软件故障还是硬件故障.即使确定是软件故障,也很难定位到具体是平台中的哪个构成软件.于是,该单位的解决方案是:与所有厂商达成协议,该平台出现问题时,所有的厂商都必须到场,一起解决问题.这不仅浪费使用者的时间,也大大增加了开发商的售后服务成本.

这仅仅是在中国的一个实例,在世界范围来说,这种例子比比皆是.基础软件平台评估得不完善,给基础软件平台的实际应用带来极大的不便,并使得其质量成为影响其扩大使用面的瓶颈之一.用户呼吁基础软件平台的评估模型的出台.这不但会帮助用户在购买之前更好地根据实际需求来挑选更适合的平台,帮助他们评估现有的平台,也会促进基础软件开发商和集成商更有针对性地、更好地改进自己产品的质量,从根本上提高平台的整体质量.因此,我们有必要对现有的各种方法进行整合和优化,同时探索新的基础软件平台评估方法.

© 中国科学院软件研究所 http://www.jos.org.cn

4 总结与展望

软件行业出现了合理、有效的基础软件平台评估的迫切需求,应该加强相关投入和研究,促进基础软件平台的进一步发展.2008 年初,中国政府正式批准实施“核心电子器件、高端通用芯片及基础软件”重大专项(简称“核高基”重大专项).国家在该专项投入之大,前所未有,一个项目就达百亿元以上,而过去 50 年相加投入才几十亿,这也从另一个方面说明了基础软件研究的重要性.

在进一步研究的过程中,我们认为如下几个方向值得探索:

1) 从工程实践的角度,研究如何有效地收集基础软件平台的度量数据,并且分析各度量数据的意义和相互关系;

2) 从层次模型的观点出发,分析基础软件平台的整体质量及其质量特性、子特性的含义、内容等,并建立层次模型;

3) 从网状模型的观点出发,建立基础软件平台质量的网状模型,分析模型中各因素的关系并定义一套关系的准则;

4) 研究如何有效处理收集的数据,用一种最适用的算法来根据度量数据计算出该平台的整体质量或用户指定的质量特性的值;

5) 开发支持基础软件平台质量评估的工具.这个工具应当建立在评估模型已存在的基础上,现阶段暂时还没有相关研究.国内有部分学者曾研究过评估其他软件形态的支撑工具,可作为参考.如华南理工大学的万江平等人开发的基于 Web 架构的软件质量评估工具[53],北航自动控制系的景涛等人开发出 SRATE 工具[54],北航软件所研发的 QESuite2.0 软件测试管理平台[55],中国科学院软件研究所互联网软件实验室的“测试过程管理与质量评价(STPM&QEA)”[56].这些工具均有较大的局限性,可支持的质量模型较少,难以支持多人、异地评估等.所以,基础软件平台的评估工具应当在功能上更齐全,并且简单易用.这样才能更好地支持评估,并有利于收集更多的数据.

应该指出,评估模型的研究过程最好能够建立在对基础软件平台的体系架构、用户质量需求研究的基础上,并且可以参考国内外软件质量相关标准,从而建立适用于基础软件平台的质量模型和质量评估模型.开展基础软件平台质量模型和质量评估模型的研究,对促进基础软件和基础软件平台在信息化建设中的应用,以及基于基础软件平台的企业级应用具有重要意义.

致谢 在此,我们向对本文的工作给予支持和建议的同行,尤其是北京航空航天大学软件所的成员表示感谢.本文试图涉及基础软件平台质量评估的方方面面,但由于作者涉猎面所限,难免有疏漏之处.也由于涉及方面太多,难免在某些方面流于表面而欠缺深度.非常感谢读者的宽容和理解.

References:

[1] Zhu SM. Software Quality Assurance And Management. Beijing: Tsinghua University Press, 2007 (in Chinese).

[2] Zheng RJ, Yin RK, Tao YL. Practical software engineering. 2nd ed., Beijing: Tsinghua University Press, 1997 (in Chinese).

[3] Yuan YY. Practical quality model for evaluating software products. Computer Engineering, 2003,29(5):32−34 (in Chinese with English abstract).

[4] Rational unified process. Rational Software White Paper, Rational the Software Development Company.

[5] GB/T 12504-90, Specification for Computer Software Quality Assurance Plan (in Chinese).

[6] IEEE Standard Glossary of Software Engineering Terminology. Standards Coordinating Committee of the Computer Society of the IEEE, 1983.

[7] Pressman RS. Software Engineering: A practitioner’s Approach. 4th Edition, R.S. Pressman & Associates, 1999.

[8] Webb D, Humphrey W. Using the TSP on the TaskView project. CrossTalk The Journal of Defense Software Engineering, l999. 3−10.

© 中国科学院软件研究所 http://www.jos.org.cn

[9] Ferrari D. Considerations on the insularity of performance evaluation. IEEE Trans. on Software Engineering, 1986,SE-12(6): 678−683.

[10] McCall JA, Richards PK, Walters GF. Factors in software quality, Vol. I, II, III. Final Technical Report, RADC-TR-77-369, Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, 1977.

[11] Boehm BW, Brown JR, Lipow M. Quantitative evaluation of software quality. In: Proc. of the 2nd Int'l Conf. on Software Engineering. Long Beach: IEEE Computer Society, 1976, 592−605.

[12] Software Product Evaluation-Quality Characteristics and Guideline for Their Use. ISO/IEC Standard ISO-9126, 1991.

[13] China Software Association in Shanghai, Shanghai Software Center. Software Quality and its Evaluation Methods. Beijing: Tsinghua University Press, 1990 (in Chinese).

[14] Kan SH, Wrote; Wu MH, Ying J, *et al*., Trans. Software quality engineering: Metric and models. 2nd ed., 2004 (in Chinese).

[15] Bansiya J, Davis CG. A hierarchical model for object-oriented design quality assessment. IEEE Trans. on Software Engineering, 2002,28(1):20−28.

[16] Mao GB, Li XJ, Ge XK, Yang MF, Zhu SY. The metrics and application of software-based component quality model. Computer Applications and Software, 2005,22(5):150−170 (in Chinese with English abstract).

[17] Yang CH, Yan HH, Jin MZ, Gao ZY. Software component quality metrics. Computer Engineering and Design, 2006,27(3):411−420 (in Chinese with English abstract).

[18] Li XL, Liu C, Jin MZ, Gao ZY. Software component: Production quality. MII, Software Component Study Group, 2006 (in Chinese).

[19] Zhou MH, Mei H, Zhang L. A multi-property trust model for reconfiguring component software. In: Proc. of the 5th Int'l Conf. on Quality Software (QSIC 2005). IEEE Computer Society Press, 2005. 142−149.

[20] Lee K, Lee SJ. A quantitative software quality evaluation model for the artifacts of component based development. In: Proc. of the 6th Int'l Conf. on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and the 1st ACIS Int'l Workshop on Self-Assembling Wireless Networks. ACM Press, 2005. 20−25.

[21] Satty TL. The Analytic Hierarchy Process. New York: McGraw2Hill, 1980.

[22] Schulmeyer GG, McManus JI. Handbook of Software Quality Assurance. Prentice Hall, 1999. 115−145.

[23] Goulão M, e Abreu FB. The quest for software components quality. In: Proc. of the 26th Annual Int'l Computer Software and Applications Conf. IEEE Computer Science, 2002.

[24] Perry WE. Quality Assurance for Information Systems: Methods, Tools and Techniques. John Wiley & Sons., 1991.

[25] Gillies A. Modeling software quality in the commercial environment. Software Quality Journal 1. 1992. 175−191.

[26] Gillies A. Software Quality: Theory and Management. 2nd ed., Int'l Thomson Computer Press. 1997.

[27] Kemerer CF. An empirical validation of software cost estimation models. Communications of the ACM, 1987,30(5):416−429.

[28] Rubin HA. A comparison of software cost estimation tools. System Development 7, 1987. 1−3.

[29] Xiong PC, Fan YS. Integrated software quality evaluation model based on fuzzy analytic hierarchy process. Computer Application, 2006,26(7):10−38 (in Chinese with English abstract).

[30] Ernest H. Forman, Decision by Objectives. http://mdm.gwu.edu/Forman/DBO.pdf

[31] Daskalantonakis MK. A practical view of software measurement and implementation experiences within motorola. IEEE Trans. on Software Wngineering, 1992,18(11):50−59.

[32] CMU/SEI, CMM, 2001.

[33] CMMI for Development (Version 1.2). Software Engineering Institute, Carnegie Mellon, 2006.

[34] Ma YS, Oh SU, Bae DH, Kwon YR. Framework for third party testing of component software. In: Proc. of the Software Engineering Conf. 2001. IEEE Computer Science, 2001. 431−434.

[35] Goulão M, e Abreu FB. Towards a component quality model. In: Proc. of the 28th Euromicro Conf on Work in Progress Session. IEEE Computer Science, 2002.

[36] Cerino DA. Software quality measurement tools and techniques. In: Proc. of the COMPSAC'86. Washington: IEEE Computer Society, 1986. 160−167.

[37] Zubrow D. Measuring software product quality: The ISO 25000 Series and CMMI. European SEPG, 2004.

© 中国科学院软件研究所 http://www.jos.org.cn

[38] A Guide to the Project Management Body of Knowledge. 3rd Edition. ANSI/PMI 99-001-2004. (in Chinese).
[39] Basili VR, Weiss DM. A methodology for collecting valid software engineering data. IEEE Trans. on Software Engineering, 1984,SE-10(6):728−738.
[40] CCID, 2005-2006 Annual Report on China's Linux Software Market (in Chinese).
[41] Jin L, Lin ZQ, Mao B, Xie L. Security policies based on object in security—enhanced operating system. Computer Science, 2003, 30(10):9−10 (in Chinese with English abstract).
[42] Shi WC, Sun YF. The development of research O13. Secure Operating Systems, Computer Science, 2002,29(6):5−12 (in Chinese with English abstract).
[43] Tanenbaum AS, Herder JN, Bos H. Can we make operating systems reliable and secure? Computer, 2006.
[44] http://qpit.ltd.uk
[45] Lu SQ. Discussion on Interoperability of Open-Source Software. Software World, 2007 (in Chinese).
[46] Komatsoulis GA. Introduction to Interoperability and Compatibility. National Cancer Institute Centre for Bioinformatics. 2007.
[47] Weis T, Ulbrich A, Geihs K. Quality of service in middleware and applications: A model-driven approach. In: Proc. of the 8th IEEE Int'l Enterprise Distributed Object Computing Conf (EDOC 2004). IEEE Computer Science, 2004.
[48] Yang SC, Liu WD, Gao CF. Research and application of a middleware-based QoS management model. Computer Engineering and Applications, 2003,39(15):184−187 (in Chinese with English abstract).
[49] Brr Group. BRR White Paper. http://www.openbrr.org /wiki/index.php/Downloads
[50] Johansson E, Wesslen A, Bratthall L, Höst M. The importance of quality requirements in software platform development——A survey. In: Proc. of the 34th Hawaii Int'l Conf. on System Science. 2001.
[51] Lan YQ, Zhao T, Gao J, Guo SH, Zhao H. DSR method for identifying the quality model of foundational software platform. In: Proc. of the 7th World Congress on Intelligent Control and Automation. IEEE Computer Science, 2008.
[52] Lan YQ, Zhao T, Guo SX, Gao J, Zhao H. Construct quality index system of foundational software platform using improved PCA method. In: Proc. of the 7th World Congress on Intelligent Control and Automation. IEEE Computer Science,2008.
[53] Wan JP, Zhuo YL, Xu X. Design and realization of Web architecture-based tool for software quality evaluation. Application Research of Computers, 2005,22(3):205−207 (in Chinese with English abstract).
[54] Jing T, Jiang CH, Liu YX. A software tool for software reliability analysis, testing and evaluation. Computer Engineering and Applications, 2005,41(1):23−26 (in Chinese with English abstract).
[55] Jia XH. Research of software external quality evaluation technology based on software testing [MS. Thesis]. School of Computer Science & Technology, Beihang University, 2007 (in Chinese with English abstract).
[56] http://www.fz863soft.com/site/cn/article/display.jsp?articleId=331

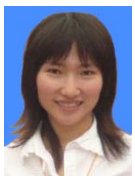
附中文参考文献:

[1] 朱少民.软件质量保证和管理.北京:清华大学出版社,2007.
[2] 郑人杰,殷人昆,陶永雷.实用软件工程.第 2 版,北京:清华大学出版社,1997.
[3] 袁玉宇.一个实用的软件质量评估模型.计算机工程,2003,29(5):32−34.
[5] 中华人民共和国国家标准-计算机软件质量保证计划规范 GB/T 12504-90.
[13] 中国软件行业协会上海分会,上海计算机软件技术开发中心.软件质量及其评价技术.北京:清华大学出版社,1990.
[14] Kan SH 著;吴明晖,应晶,等译.软件质量工程——度量与模型.第 2 版,北京:电子工业出版社,2004.
[16] 毛国蓓,李雪静,葛孝堃,杨明福,朱三元.基于软件构件质量模型的度量及应用.计算机应用与软件,2005,22(5):150−170.
[17] 杨春河,晏海华,金茂忠,高仲仪.软件构件质量度量.计算机工程与设计,2006,27(3):411.
[18] 李晓丽,刘超,金茂忠,高仲仪.《软件构件产品质量》标准介绍.信息产业部软件构件标准工作组特别报道,2006.
[29] 熊鹏程,范玉顺,基于模糊层次分析法的集成软件质量评估模型.计算机应用,2006,26(7):10−38.
[38] 项目管理知识体系指南(第 3 版),美国国家标准.
[40] 赛迪顾问,2006-2007 年中国 Linux 软件市场研究年度报告.
[41] 金雷,林志强,茅兵,谢立.安全操作系统中基于客体的保护机制,计算机科学,2003,30(10):9−10.
[42] 石文昌,孙玉芳.安全操作系统研究的发展,计算机科学,2002,29(6):5−12.

© 中国科学院软件研究所 http://www.jos.org.cn

[48] 杨胜春,刘卫东,高朝发.基于中间件的 QoS 管理模型的研究与应用.计算机工程与应用,2003,39(15):184–187.
[53] 万江平,卓永乐,徐晓.基于 Web 构架的软件质量评估工具的设计和实现.计算机应用研究,2005,22(3):205–207.
[54] 景涛,江昌海,刘永祥,胡德斌.软件可靠性分析、测试与评估工具——SRATE 介绍.计算机工程与应用,2005,41(1):23–26.
[55] 贾兴华.基于测试的软件外部质量评价方法的研究与实现[硕士学位论文].北京航空航天大学,2007.

兰雨晴(1969－),男,内蒙古呼和浩特人,副教授,主要研究领域为软件质量工程,软件项目管理.



赵同(1984－)女,硕士生,主要研究领域为软件质量评估.

高静(1970－)女,博士生,副教授,主要研究领域为软件测试.

接卉(1980－)女,助理工程师,主要研究领域为软件质量评估.

金茂忠(1941－),男,教授,博士生导师,主要研究领域为软件测试,软件开发环境.

www.jos.org.cn
www.jos.org.cn

© 中国科学院软件研究所 http://www.jos.org.cn