

关于软件测试的几点思考

聂长海

(南京大学计算机科学与技术系 南京 210093)

摘要 近些年来,软件测试越来越受到产业界、教育界和学术界的重视。结合作者在软件测试领域的教学和科研积累,对软件测试从课程、职业和科学三个层面进行思考,分别探讨了软件测试作为一门课程时,它的知识体的构成;作为一个职业时,它的职业技能和素养所包括的内容;作为一门科学时,它的主要科学问题有哪些。

关键词 软件测试,课程,职业,科学

中图分类号 TP311 **文献标识码** A

Thoughts on Software Testing

NIE Chang-hai

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)

Abstract In recent years, software testing is getting more and more attention from Industry, Education and Academic. Combining the author's experience in teaching and research in this area, this paper presented three level thoughts on the course, professional and science of software testing respectively, discussing on what constitutes the knowledge body as a software testing course, what is vocational skills and literacy of software testing as a professional, and what is the major scientific problems as a science.

Keywords Software testing, Course, Vocation, Science

软件测试作为软件质量保障的一种重要的方法,近年来已经得到软件产业界、学术界和软件工程师们普遍重视,目前,软件测试的教学内容已经从以前作为软件工程课程中的一章发展为一门独立的课程,国内外拥有计算机系的大学几乎都开设了这门课程,在国内的几个主要的售书网站上可以找到的软件测试教材和书籍就有 30 多种。产业界也逐渐认识到不仅软件开发可以成为一门职业,软件测试也是一门重要的职业,人们期待着能够培养出可以尽早发现软件错误的专门人才,这样可以尽量减少软件开发维护的成本和日后因为软件故障而造成的损失。学术界已经把软件测试作为一门重要的相对独立的科学在研究,关于软件测试的各种国际学术会议就有十几个,也有专门的学术期刊和关于软件测试的各种学术组织。

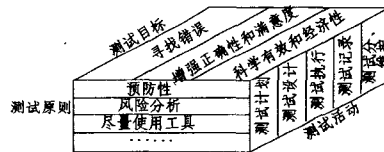
软件测试作为一门课程,应该有一套系统的知识体;作为一个职业,应该有自己的职业技能和素养;作为一门科学,应该有它独立的科学问题体系。下面我们分别从这三个方面进行探讨。

1 软件测试课程的知识体

作为一门课程,我们需要有足够的內容,即知识体。首先是软件测试的定义₁(注:本文下标的特殊用途是知识点编号),告诉学生什么是软件测试,可以从最早最简单的定义“软件测试是为了发现错误而执行程序的过程”开始,到稍微复杂

一点的定义,可以逐步讲解四五个定义(这些定义都可以从不同的教科书中找到,有的教科书就列出了这些定义并进行了比较^[1]),从这些定义中一方面可以看到人们对软件测试概念认识的深入,另一方面也逐步认识到软件测试的很多层面,从而系统理解软件测试。

图 1 给出了软件测试的三个层面:目标₂、活动₃和原则₄,强调三维一体。对于软件测试的目标,除了强调发现错误无论对开发方还是用户都是为了节约成本、减少损失、提高质量等软件测试目的和意义外,还要列举一些因为软件缺陷造成重大影响和损失的例子(这方面的例子很多,例如美国航天局火星登陆事故,跨世纪“千年虫”问题等^[2])。软件测试的原则可以包括十几条^[1],软件测试的过程可以分为五步或六步(见图 1 中测试活动),可以结合相应的例子进行讲解。所有以上内容再加上回答两个问题:为什么要开软件测试课程,怎样学好这门课程,构成软件测试这门课程的概论,可以在 2~4 个学时内完成教学。



软件测试的目标、活动和原则

图 1 软件测试三维一体图

另一个重要问题是软件测试的分类,这部分内容可以放

本文受国家自然科学基金资助项目(60773104,60721002),国家高科技研究发展计划(“863”计划)(2008AA01Z143),江苏省自然科学基金(BK2010372)资助。

聂长海(1971—),男,博士,副教授,主要从事软件测试领域的科研和教学工作, E-mail: changhainie@nju.edu.cn.

在概论部分介绍如何分类,分类后分别有哪些测试方法。软件测试课程的知识体就集中在各种测试方法上,这门课程的后续部分就是围绕软件测试的分类,介绍各类软件测试的目的、方法和工具。

软件测试根据是否运行程序可分为静态测试⁶和动态测试⁶。静态测试包括桌面检查⁷、代码审查⁸和代码走查⁹等方法。动态测试根据测试用例设计是否依据内部结构可以分为黑盒测试¹⁰和白盒测试¹¹,白盒测试包括语句覆盖¹²、判定覆盖¹³、条件覆盖¹⁴、判定/条件覆盖¹⁵、条件组合覆盖¹⁶、路径覆盖¹⁷、线性代码序列及跳转测试¹⁸等;黑盒测试包括等价类划分¹⁹、边界值分析²⁰、因果图分析²¹、错误猜测²²、状态转换测试²³等^[3,5]。

根据软件开发的阶段可以将软件测试划分为单元测试²⁴、集成测试²⁵、系统测试²⁶、验收测试²⁷、回归测试²⁸、验证测试²⁹、确认测试³⁰、 α 测试³¹、 β 测试³²和 λ 测试³³等。

根据被测软件的开发方法和应用环境的不同可以分为面向对象软件测试³⁴、面向方面软件测试³⁵、面向服务软件测试³⁶、构件软件测试³⁷、嵌入式软件测试³⁸、Web应用软件测试³⁹等,后面还要出现普适计算环境下的软件测试⁴⁰、云计算环境下的软件测试⁴¹等。

根据软件不同特性和方面的测试可以分为:负载测试⁴²、压力测试⁴³、性能测试⁴⁴、安全性测试⁴⁵、安装测试⁴⁶、可用性测试⁴⁷、稳定性测试⁴⁸、授权测试⁴⁹、用户接受性测试⁵⁰、一致性测试⁵¹、配置测试⁵²、文档测试⁵³、兼容性测试⁵⁴和 Playtest⁵⁵等^[4,6]。

根据不同特殊的测试技术可以有:组合测试⁵⁶、蜕变测试⁵⁷、变异测试⁵⁸、演化测试⁵⁹、FUZZ测试⁶⁰、基于性质的测试⁶¹、基于故障的测试⁶²、基于模型的测试⁶³、基于操作剖面的测试⁶⁴、基于用例和/或用户陈述开发测试用例⁶⁵、基于规格说明的测试⁶⁶、统计测试⁶⁷、逻辑测试⁶⁸、随机测试⁶⁹、自适应随机测试⁷⁰、GUI测试⁷¹、冒烟测试⁷²和探索测试⁷³等^[3]。

针对以上60多种形式多样的软件测试方法,一方面要分别介绍各种方法的概念、功能特点和优缺点、使用步骤、应用举例和相关工具等,另一方面,这些方法之间既有明显的区别,又有很多联系,需要对它们之间的相互关系进行比较。其中有的方法可能很短时间就能介绍完,例如文档测试、配置测试等各种针对软件不同特性和方面的测试方法;有的方法则需要几个学时才能介绍清楚,例如面向对象的测试、嵌入式软件测试等针对不同开发方法和应用场景软件的测试方法。综上所述,除了软件测试的概念、管理和测试文档等内容,软件测试的各种方法组成了这门课程的主体,可以有30个学时左右的内容。

2 软件测试的职业技能和素养

几乎所有的职业都有一定的门槛,所谓门槛,就是其特殊的职业技能和基本素养,一个职业的重要程度不仅取决于社会需求,也取决于门槛的高低。例如,飞行员是一个非常重要的职业,过硬的身体和心理素质以及熟练的飞行驾驶技能要求形成了很高的职业门槛,使得能够从事这个职业的人员很少,培养一个成熟的飞行员的成本是非常昂贵的。

软件测试从业人员不仅需要系统地掌握软件测试课程中的知识,还要在实践中不断摸索,将理论联系实际,善于不断

尝试新的测试方法和测试工具,测试新软件,探索新问题,积累新经验。软件测试既是一门科学,也是一门艺术,测试员丰富的经验和敏锐的洞察力往往是成功的关键。一个测试人员如果掌握的软件测试方法越多,会用的测试工具越多,测试过的软件越多,碰到过的问题越多,那么他在新的测试任务到来的时候,成功的可能性就越大,这样他的价值也就越大。

从最广泛的意义上讲,测试无处不在,因为各行各业都会有各自的产品,在他们的产品出厂前,都需要进行测试,以保证产品质量合格,同时我们生活的每个角落,例如到商场买东西,要看看产品质量,跟别人交往,要测试对方是否友好等,特别是在社会生活中我们也一直在接受来自学校、单位或其他个人的测试。正因为测试无处不在,所以可以说人人都是测试员,每个人都在利用自己独立的视角或者说独具慧眼在生活中进行着辨识和选择。软件是一种特殊的产品,软件测试是一种特殊的测试,因此有着非常广泛的从业人员基础,软件测试从业人员可以有广泛的背景,从对专业知识一无所知的门外汉到百分之百的领域专家,当然更多的测试员是介于两者之间。但在这个行业中要成为一个成熟的软件测试专业人员,还是非常困难的,需要厚实的专业基础、不断的学习、实践和积累。

软件测试的专业基础涉及数学、计算机科学、软件工程、计算机工程以及信息技术等非常广泛的领域,而不是仅仅学好软件测试这门课程就可以了,而且如果没有这些专业基础,软件测试课程也是学不好的,关于这一点,从软件测试课程中广泛的知识体也可以看出。一般来讲,都是计算机或相关专业的毕业生毕业后根据兴趣和需要,经过一定的培训或学习后,开始从事软件测试职业。

人们总是喜欢根据从业人员的能力和成熟度将一个职业分成几个等级,在软件测试领域我们也可以将从业人员分成5等级(也可以更多,这里只给出5个等级作为参考):1级称之为用户级测试人员,这是软件测试人员的最低级别,没有受过软件测试的专门训练,直接从用户角度通过使用软件来发现问题;2级是软件测试操作员,受过软件测试的专门训练,不仅可以从用户角度进行用户级测试,还可以完成上级下达的具体的测试任务;3级是软件测试技术员,能够非常熟悉地驾驭某些软件测试专门技术,设计一些高级的测试用例,进行高效的软件测试,可以将一些测试任务下达给2级测试操作员,并具有很多软件测试成功的经验;4级是软件测试系统工程师,系统掌握所有的软件测试方法,具有计算机等相关专业背景,并曾是成熟的软件开发人员,具有充分的软件测试经历;5级是软件测试总工程师,既是软件测试系统工程师,具有丰富的软件测试经验,又能做项目经理,负责管理整个软件测试项目^[3]。

3 软件测试中的科学问题

软件测试领域存在着很多非常困难的科学问题,例如,对于一个特定的软件,如何选择一组有效的测试方法,对之进行科学的测试?如何从庞大的可用测试用例空间中选择少量的测试用例对该软件进行有效的测试?软件测试什么时候可以停止?等等。正是这些问题推动着软件测试领域学术研究的繁荣,很多研究者试图从自己的视角和基础出发给出新的解决方案^[2]。

2000年, M J Horrold 对软件测试领域进行总结, 提出软件测试领域的主要问题是探索各种新的软件测试方法和过程, 开发相应的工具, 并进行实证研究^[7]。2007年, Antonia Bertolino 又重新对该领域进行总结, 她首先肯定人们在可靠性测试、测试过程、协议测试、测试充分性准则及相互比较、构件测试及面向对象软件测试方面取得的成绩。然后指出了人们在测试输入生成、测试预期输出、测试成本及有效性、基于假设的测试、针对不同计算方式软件的测试、功能性和非功能性测试等领域的挑战, 提出了建立统一的测试理论、基于测试的建模、测试自动化和效率最大化测试工程等四项研究目标^[8]。

我们将软件测试的研究划分为两大类, 一类是图2中上面部分, 研究针对各类具体软件的测试方法, 另一类是下面部分, 可称为软件测试的基本方法。下面的方法体系可为上面

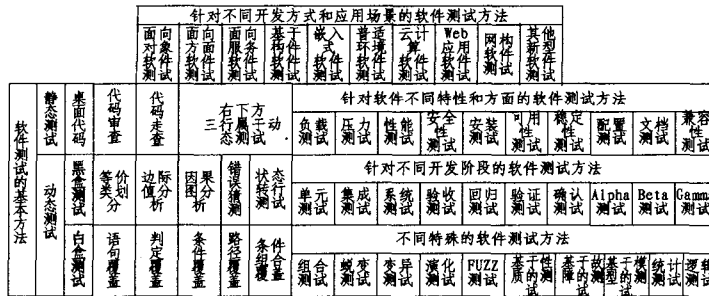


图2 软件测试的方法体系

具体软件的测试服务, 同时具体软件测试也给基本测试方法提出了新的要求。

图2下面部分的软件测试基本方法中的每一种方法都有其独特的测试目标, 用以解决不同的测试问题, 采用不同的测试用例生成方法, 从而具有不同的特点。例如, 组合测试目标是检测待测软件系统中各种因素相互作用引发的故障, 采用组合设计方法生成测试用例, 可以有效地发现交互性错误。边界值分析基于人们的实践经验; 边界点引发错误可能往往较大, 所以在设计测试用例时, 充分采用边界值。每一种方法在使用时, 其效果往往因人而异, 原因就是每一种方法都值得研究, 我们需要对其中每一种方法都要进行认真研究, 探索各种方法的具体使用步骤、工具支持、使用成本、存在的风险等问题。

目前关于某种特殊的测试方法的研究非常多, 例如蜕变测试、变异测试、组合测试、演化测试等等, 这些研究一方面研究和尝试如何充分发挥特定的测试方法的作用, 另一方面, 通过一些具体的应用, 收集该方法有效性的证据, 并在实践中发现和解决一些新的问题。例如在软件发生修改后, 研究如何以最小的测试用例集、最少成本的测试保证修改的正确性以及修改未带来任何副作用; 怎样定量评估各种测试充分性准则的有效性等问题。其实, 测试用例集的最小化和测试有效性评估也是所有测试方法的共性问题, 除此之外, 还有测试目标及测试用例生成问题; 测试用例的优化, 即寻找一个最好的测试用例执行顺序, 使之达到最好测试效果; 还有在用不同的测试方法时, 如果发现故障, 如何进行故障定位问题; 不同测试方法的测试充分性问题等。

除了对单个测试方法进行系统研究, 我们还需要从软件测试不同层面进行系统的交叉结合和比较研究, 例如针对各种测试覆盖标准的比较研究, 软件开发过程中不同阶段的测试间关系研究, 面向各种具体开发方法和应用环境的软件测试方法的比较研究, 软件的各种特性测试, 以及各种测试技术之间的比较研究等, 目标是探索并建立软件测试研究和实践的多维度和系统化的科学体系。例如, 人们已经对随机测试和自适应随机测试^[9]、随机测试和组合测试等进行了比较研究^[10,11]。

结束语 软件测试领域在最近的10多年里发展迅速, 本文从软件测试课程、软件测试职业和软件测试科学三个层面试图对该领域进行一个系统的梳理, 以期抛砖引玉, 为软件测试领域的产学研紧密结合和更好更快发展提供新的系统化基础。系统地总结了软件测试课程知识体, 以便为软件测试课程的发展和教学改革提供新的基础和指导; 探讨了软件测试职业的职业技能和素养, 以为软件测试从业人员自身发展和提高提供参考; 系统探讨了目前软件测试研究领域的科学问题, 以为该领域科学工作者从事更加有效的研究工作提供新

的思路。

不同的产品有不同的测试方法, 所有这些测试方法全体构成一个大测试领域, 软件是一种产品, 软件测试是大测试领域中的一员, 如图3所示。软件测试领域的发展已经并将继续从大测试领域获得营养, 即从其他产品测试方法中获得启发或借鉴, 例如从人才选拔测试过程中可以学习到像给学生考试一样考查软件; 同时人们在软件测试领域本身取得的进步反过来也将推动大测试领域的进步。

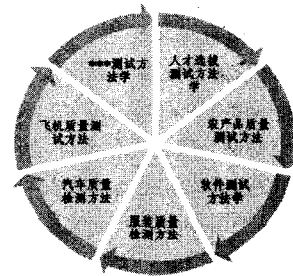


图3 大测试领域中各类产品测试方法体系之间可以相互借鉴

注: 本文编号的软件测试知识体中的知识点都可以从近期出版的软件测试教材, 或从谷歌和百度搜索中找到详细的解释, 特别建议那些感兴趣的软件测试领域从业人员、教师、学生和研究人员尝试利用网络搜索方法逐个查找文中标注的概念, 以进一步了解相应的知识, 本文限于篇幅, 不赘述每个概念解释。

参考文献

- [1] (美) Craig R D. 系统的软件测试[M]. 杨海燕, 罗洁雯, 译. 北京: 电子工业出版社, 2003
- [2] Patton R. 软件测试[M]. 周子滨, 姚静, 等译. 北京: 机械工业出版社, 2001

(下转第27页)

- Proceedings of the 10th International Conference on High Performance Computing(HiPC 2003). Heidelberg, Berlin; Springer, 2003;300-309
- [42] Weide B W, Ogden W F, Sitaraman M. Expressiveness issues in compositional performance reasoning [C]//Proceedings of the 6th ICSE Workshop on Component-based Software Engineering: Automated Reasoning and Prediction. <http://www.csse.monash.edu.au/~hws/cgi-bin/CBSE6/Proceedings/proceedings.cgi>, 2010-02-11
- [43] Zhao Lei, Jarvis S A, Spooner D P, et al. Predictive Performance Modelling of Parallel Component Compositions [J]. Cluster Computing, 2007, 10(2): 155-166
- [44] Fumento N, Mayer A, McGough S, et al. An Integrated Grid Environment for Component Applications[C]//Proceedings of the Second International Workshop on Grid Computing. Heidelberg, Berlin; Springer, 2001; 26-37
- [45] Newhouse S, Darlington J. Computational Communities: A Marketplace for Federated Resources[C]//Proceedings of the 9th International Conference on High-performance Computing and Networking, 2009. Heidelberg, Berlin; Springer, 2009; 667-674
- [46] Aldinucci M, Campa S, Ciullo P. A framework for experimenting with structured parallel programming environment design[C]//Parallel Computing: Software Technology, Algorithms, Architectures and Applications, PARCO 2003. Vol. 13 of Advances in Parallel Computing. Dresden, Germany; Elsevier, 2004; 617-624
- [47] The Common Component Architecture Forum, CCA Forum homepage[OL/OL]. <http://www.cca-forum.org>, 2010-02-16
- [48] Damevski K, Parker S G. Parallel Remote Method Invocation and M-by-N Data Redistribution[C]//Proceedings of the 4th Los Alamos Computer Science Institute Symposium (LACSI 2003). Santa Fe, New Mexico, USA; Los Alamos National Laboratory, 2003; 10
- [49] Bertrand F, Bramley R, Bernholdt D E, et al. Data Redistribution and Remote Method Invocation in Parallel Component Architectures[C]//Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS '05). Volume 01. Washington, DC, USA; IEEE Computer Society, 2005; 40
- [50] Allan B, Armstrong R C, Wolfe A P, et al. The CCA Core Specification in A Distributed Memory SPMD Framework[J]. Concurrency and Computation: Practice and Experience, 2002, 14(5); 323-345
- [51] Allan B, Rob A. Ccaffeine Framework; Composing and Debugging Applications Interactively and Running Them Statically[C/OL]// http://compframe.org/compframe2005/CompFrame05_Program/files/01031457.pdf, 2010-03-01
- [52] Zhang Ke-ming, Damevski K, Venkatachalapathy V, et al. SCIRun2: a CCA framework for high performance computing[C]//Proceedings of The 9th International Workshop on High-Level Parallel Programming Models and Supportive Environments. <http://sest.vsu.edu/~kdamevski/files/hips2004-SR2.pdf>, 2010-03-01
- [53] Parker S G. A component-based architecture for parallel multi-physics PDE simulation[J]. Future Generation Computer Systems, 2006, 22(1): 204-216
- [54] Balay S, Gropp B, Curfman L, et al. A microkernel design for component-based parallel numerical software systems[C]//Proceedings of the SIAM Workshop on Object Oriented Methods for Inter-operable Scientific and Engineering Computing. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.7992>, 2010-03-01
- [55] Denis A, Perez C, Priol T, et al. Padico: a component-based software infrastructure for Grid computing[C/OL]. Proceedings of International Parallel and Distributed Processing Symposium. <ftp://ftp.inria.fr/INRIA/publication/publi-pdf/RR/RR-4974.pdf>, 2010-03-01
- [56] Parker S G, Weinstein D W, Johnson C R. The SCIRun computational steering software system[C]// Arge Erlend Bruaset A M, Langtangen H P, eds. Modern software tools for scientific computing. Cambridge, MA, USA; Birkhauser Boston Inc, 1997; 5-44
- [57] Shirazi B A, Hurson A R, Kavi K M. Scheduling and load balancing in parallel and distributed systems[M]. Los Alamitos, CA, USA; IEEE Computer Society Press, 1995
- [58] Dahlgren T, Epperly T, Kumpf G, et al. Babel Users' Guide [OL/OL]. https://computation.llnl.gov/casc/components/docs/users_guide/users_guide.html, 2009-03-02
- [59] 杨美清. 软件复用及相关技术[J]. 计算机科学, 1999, 26(05): 1-4
- [60] 樊晓光, 褚文奎, 万明. 基于领域本体的软构件检索[J]. 计算机科学, 2009, 36(6): 156-158

(上接第 3 页)

- [3] Ammann P, Offutt J. Introduction to Software Testing[M]. United Kingdom, Cambridge University Press, 2008
- [4] (美)Myers G J. 软件测试的艺术(原书第二版) 中文版[M]. 王峰, 陈杰, 译. 北京: 机械工业出版社, 2006. 1
- [5] Computing Curricula 2001; Final report of The Joint Task Force on Computing Curricula IEEE Computer Society and Association for Computing Machinery, 2001
- [6] Standard for Software Component Testing, British Computer Society[S]. SIGIST. 2001, <http://www.testingstandards.co.uk/>
- [7] Harrold M J. Testing: a roadmap[M]. ICSE-Future of SE Track, 2000; 61-72
- [8] Bertolino A. Software Testing Research; Achievements, Challenges, Dreams[M]. FOSE. 2007; 85-103
- [9] Chen T, Kuo F, Liu H. Adaptive random testing based on distribution metrics[J]. Journal of System and Software(JSS)
- [10] Schroeder P J, Bolaki P, Gopu V. Comparing the fault detection effectiveness of n-way and random test suites[C]//ISESE. IEEE Computer Society, 2004; 49-59
- [11] Nie Changhai, Leung H. A survey of combinatorial testing[J]. ACM Computing Survey, 43(2)