

软件测试技术、 方法和环境

徐拥军 王 炯 郝 进 编著
周伯生 主审



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

软件测试技术、方法和环境

徐拥军 王 炯 郝 进 编著
周伯生 主审

北京航空航天大学出版社

BEIHANG UNIVERSITY PRESS

内 容 简 介

本书是作者近 20 年实践经验的总结与提高,全面论述了软件测试技术、过程和方法,以及组织级的软件测试体系建设和运用。一方面讲述了软件测试目的和原则、测试的组织形式、组织级软件测试体系建设及测试人员培养等。另一方面从技术角度论述了同行评审方法、测试用例设计方法、测试度量与分析、自动化测试及性能测试等。最后通过一个行业核心业务系统测试案例,展现了各种测试技术和管理方法如何综合使用。提出的基于迭代的测试过程和行业核心业务系统测试是实践的总结,适用于中国当前的现实情况。

本书有助于测试人员及其他技术人员快速提高测试能力,适合业内人员阅读、使用,也可以作为计算机专业的教学参考书。

图书在版编目(CIP)数据

软件测试技术、方法和环境 / 徐拥军等编著. --北京
:北京航空航天大学出版社,2012.6

ISBN 978-7-5124-0707-7

I. ①软… II. ①徐… III. ①软件—测试 IV.
①TP311.5

中国版本图书馆 CIP 数据核字(2012)第 008600 号

版权所有,侵权必究。

软件测试技术、方法和环境

徐拥军 王 炯 郝 进 编著

周伯生 主审

责任编辑 许传安

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: bhpress@263.net 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本:787×1092 1/16 印张:14.75 字数:378 千字

2012 年 6 月第 1 版 2012 年 6 月第 1 次印刷 印数:3 000 册

ISBN 978-7-5124-0707-7 定价:36.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。



序

《软件测试技术、方法和环境》一书是三位作者多年来实践经验的总结。他们通过理论联系实际、技术与管理相结合的方式,从管理方面阐述了软件测试的目的、原则、组织形式、组织级软件测试体系建设以及人员培养等问题,从技术角度论述了迭代测试过程、同行评审、测试用例设计、测试度量与分析、性能测试以及自动化测试等技术。最后用一个行业核心业务系统测试案例,展示了各种测试技术和管理方法的综合使用,是软件和系统测试人员的重要参考资料,也可供其他开发人员和各级管理人员阅读。

我受邀担任本书主审,不仅可以与多年的朋友深入合作,而且是因为软件测试确实是建造软件和系统的一项不可或缺的重要技术。在这方面出版一本有价值的书籍,可为我国的软件和系统建造产业作一份贡献。

目前我们所处的信息化时代,是人类进入综合利用物质、能量和信息三种资源的知识经济时代。千百年来以传统的物质产品的生产、流通、消费为基本特征的物质型经济,正在逐步转为以信息产品的生产、流通、利用和消费为基本特征的现代化知识型经济。复杂的巨型计算机系统是这个时代的必不可少的基础设施,这类系统需要测试技术来确认。

同时,当前全球以云计算为特征的第四代计算模式风起云涌,我国从关键设备、数据中心系统、云安全、云应用服务等多个层次,正在制订长期发展战略,以指导构建自主可控的中国云,以确保国家信息基础设施和信息资源的安全。显然这类超大规模巨型系统的开发和确认,也需要采用测试技术。

此外,就软件系统本身来说,美国国防部认为大型软件项目不成功的原因70%是管理问题,30%是技术问题,就此幸运而诞生和发展的CMMI已经取得了巨大的成效,证明过程不仅是信息产业生产力的三要素之一(另两个要素是人员和技术),而且是三要素的纽带和杠杆作用点。但是过程和技术必须两者兼顾,相辅相成。

我相信,积极、正确运用本书阐述的测试技术,可以对知识经济、云计算、过程改进提供重要的支持。我期望本书能对这三个大趋势真正做出有效的贡献,但我更热切地期望读者对本书的缺点和存在的问题提出批评、指正。

周伯生 于北京
2012年5月2日



前 言

我们生活在一个伟大的时代,可以随时随地得到想要的任何信息,但同时也往往陷入知识的海洋而找不到方向和重点。在竞争激烈的现实环境中,我们比以往任何时候都关注自身成长,特别渴望构建属于自己的知识体系和方法论,以便驾驭复杂多变的工作场景。作为一本软件测试的专著,本书希望帮助读者构建这样的知识体系,也希望帮助软件组织建立软件测试体系。因此本书主要论述两方面的内容,一是关于软件测试的过程、方法、技术和实践等,帮助测试从业人员及其他技术人员提升软件测试能力;二是关于组织级软件测试体系建设,帮助软件组织的管理者提高整个组织的软件测试能力。这两部分内容相互支撑,形成关于软件测试的知识体系和方法论。

软件工程的范畴很大,为什么专门论述软件测试呢?这与本书的总指导兼审校者周伯生教授和几位作者关于软件工程的认识有关。一方面,在当今世界及中国的软件开发和项目实施中,软件测试所占的分量越来越大(已达项目总工作量的40%左右),其重要性越来越大,但各有关方(包括软件公司、用户、管理者和技术人员等)对测试的重视、关注和投入都很不够,测试从业人员力量也比较弱,这种状况往往导致不仅做不好测试,更影响整个项目的成败和效率。另一方面,在这种情况下,在测试过程、方法和技术等方面的一点点改进都有可能对软件工程实践起到显著的改善作用,而在一个组织内建立一套测试体系,则更能极大地促进软件测试,乃至其他方面的持续发展和提升。所以说,应该狠抓软件测试,并在组织范围内进行测试体系的建设。

随着软件工程理论和实践的发展,软件测试已经成为一门独立的学科,我们可以看到软件测试方面的专著。阅读这些经典著作有助于我们了解与测试相关的概念、模型和方法等理论知识。但由于内容过于庞杂,与实际情况有一段距离,要掌握起来也不是那么容易;而且,目前大部分的著作主要涉及测试方法和技术层面,较少论及组织级测试体系建设方面的内容。面对这些问题,我们认为有必要写一点东西,将测试理论和实践结合起来,将测试技术和测试管理结合起来,将项目级技巧和组织级体系结合起来。

我们来自神州数码公司和北京赛柏科技公司,都是长期从事软件开发、测试和研发管理的技术人员,在软件公司里承担着测试经理、测试部经理和研发管理者的角色,也面临着同样的问题。为了做好软件测试,我们引入了必要的测试理论和方法,并结合理论在各种软件研发和工程项目中承担具体的测试任务,同时还要考虑测试人员培养和测试体系建设等问题。在这个过程中,我们将相关的经验和教训总结出来,慢慢形成了关于测试的一些原则、过程、方法、技术和实用做法,并在实践中不断对它们进行检验和改进,逐步形成了一些比较成体系的东西,成为了整个组织的测试体系。这样的工作大约从2001年开始,持续了十余年的时间。回头来看,这样的工作还是有一定的借鉴意义的。从2009年开始,在周伯生老师的鼓励和指导下,我们将相关的内容进行整理和再提炼,以便汇集出版。这就是本书的源起。因此,本书不是测试理论的集中介绍,也不是测试技术的细节描述,而是测试经验和教训的结晶,希望带给



读者一个正在测试现场的真实体验,加深读者关于测试本质的认识,帮助读者利用本书介绍的方法和体系来构建一个属于自己的测试体系和方法论。

本书共 11 章。其中前 5 章由徐拥军执笔(也是本书的统稿人),第 7 章由王炯执笔,第 6 章和第 11 章由郝进执笔,第 8 章至第 10 章由 3 人共同撰写。下面简要介绍一下。

第 1 章“测试技术引论”阐述与软件测试相关的比较本源、本质和基础的内容。首先从系统工程角度来看待软件测试,为软件测试及后面的章节内容引入一个方法论,而后面的章节也在自觉地利用系统工程方法来揭示软件测试的奥秘;再回顾软件测试的起源和发展历史,以正本清源、辨伪存真;然后从实际工作体会出发,重新认识测试的目的和作用;在此基础上,本章基于系统工程思想,提出了软件测试的基础原则,即“6W 原则”。这些原则也是我们关于项目测试实践和测试体系建设的经验教训的总结。

第 2 章和第 3 章重点论述与人相关的两个专题。第 2 章“测试组织形式”介绍软件测试几种典型的组织形式,分析它们的不同特点、优势和劣势,说明各自成功的关键点所在,并开发了测试组织形式的选择表,帮助软件组织选择合适的测试组织形式。第 3 章“测试人员成长过程”关注测试工程师的成长之路,涉及测试能力培养、测试心理调适、测试与开发协作等专题,对测试人员的能力和素质要求进行分解,提出测试人员如何“过五关”(心理关、业务关、技术关、专业关和管理关等),一步步迈向成功之路,成为测试专家,并给出了自我评估和制订发展计划的方法。对于组织来说,这些内容对于创建学习型组织、打造高性能团队是有益的。

第 4 章“组织级测试体系总体设计”转入正题,给出组织级软件测试体系的总体设计,说明测试体系的内涵、内容、结构、各要素间的关系、建设的过程和方法等。本章还引入了测试成熟度模型,它定义了测试体系的不同等级及其内涵,指出了测试体系建设努力的方向。本书后面的章节是测试体系建设的一个分解和细化的过程。我们将在每一章重点阐述一个专题,在每一个专题中先介绍业界通行的和我们总结的相关原则、过程、方法、技术和指南等;然后论述与组织级测试体系建设相关的内容。

第 5 章“基于迭代的测试过程”从组织级软件测试体系建设的角度谈软件测试过程的建立。通过分析不同测试过程模型的优缺点,结合现实情况,提出了基于迭代的软件测试过程的概念,说明了它的两层内涵和五大特点,并在基于迭代的测试过程的背景下分专题论述如何监视和控制测试过程的重要方面。实践初步证明了此过程适应当今中国各行业软件开发和测试的情况,符合互联网时代软件开发和运营的新趋势。

第 6 章和第 7 章转入测试方法专题,分别论述同行评审过程和方法以及测试用例设计方法。前者涉及静态测试;后者涉及动态测试。第 6 章“同行评审过程和方法”强调好的软件组织不仅要认真进行动态测试,还要重视以同行评审为核心的静态测试,特别要重视需求同行评审和设计同行评审。为此,本章介绍了需求和设计评审、各种代码评审方式和开发人员自测等的过程和方法,并说明实施 CMMI 和 TSP、PSP 相结合的软件过程改进框架,可以帮助组织和技术人员改进这些环节的工作质量。本章也确立了同行评审的度量原则和精确的度量方法。第 7 章“测试用例设计方法”介绍了白盒测试用例设计方法和黑盒测试用例设计方法,并讨论了测试用例设计的策略。本章介绍的设计实例和第 11 章中的实际设计实例,方便读者理解各种设计方法。



第8章“测试度量与分析过程”介绍软件度量和测试度量的基本概念和模型,并择其精要,从实用角度分两个方面(即测试计划度量和测试过程度量)来说明度量与分析的内容、过程、方法、技术和案例等。这些内容有助于读者制订准确有效的计划,度量和分析缺陷数据,改进测试过程。本章还论述了如何建立测试度量与分析体系,并介绍了一个实用的质量监控系统,即“赛柏质量监控系统”,利用它可以进行测试度量与分析,持续提升项目测试效率和组织级测试过程性能。

第9章“自动化测试体系建设”论述自动化测试专题。讨论自动化测试的策略和误区,论述自动化测试基础建设、自动化测试框架、测试工具、测试脚本开发等专题,并结合案例介绍进行说明。本章介绍了我们自己开发的测试工具 Sm@rtest,利用它可以开展自动化测试。本章还介绍了自动化测试过程建立方面的内容,强调自动化测试是组织级测试体系的重要内容,应该逐步开展起来。

第10章“性能测试过程和方法”论述性能测试专题。本章采取提纲挈领的办法,阐述了性能测试的重要方面,包括如何理解系统性能和性能测试的特点,如何确定性能测试的需求,如何进行性能测试的规划和设计,如何实施性能测试,如何做系统调优等。在此基础上探讨了性能测试体系建设和性能测试队伍建设等问题。本章的内容有助于读者理解性能测试的本质,体会性能测试的禀性。

第11章“行业核心业务系统测试实践”对我们执行的一个实际项目测试案例进行了剖析,用以说明实际项目中的软件测试是什么样的。本书阐述的相关原则、过程、方法和技术等是如何在实际项目中实际应用的,以及建立的组织级软件测试体系如何应用于实际项目中,并发挥作用。其中介绍了我们提出的“测试用例设计六步法”,它是基于测试要素分析和正交矩阵设计方法在税务行业的特化应用,还有“流程用例设计方法”和“基于状态图的测试用例设计方法”等都独具特色。本章介绍的测试用例设计实例、大项目测试过程掌控、项目测试实际总结和体验报告等内容会给读者以正在现场的感觉,读起来应该是流利自然的。

本书的出版首先要感谢北京航空航天大学周伯生教授,他是在国内外享有声誉的中国软件工程专家,并一直致力于中国软件企业的软件过程改进和质量提升。作为本书的总指导和主审者,为本书确立了总体结构和质量标准,并时常指导我们:“(1)写一本书是自己的一个里程碑。(2)书是写给他人看的,可能要给几千人看。为此我们要高度负责,不要怕麻烦,要精益求精。(3)一本书是一个整体,因此要严谨、严密、严格。写书是一件非常辛苦的事,愈到后来愈要有耐性。”在他的指导下,我们几易其稿,他也反复审校,很多内容相当于由他重写。可以说,没有他的鼎力支持,就没有本书的出版。在这里,还要感谢周伯生教授领导的北京赛柏科技有限责任公司(Beijing Cyberspi)为本书提供了资料。

在这里感谢北京航空航天大学出版社及其责任编辑的大力支持。

我们还要感谢神州数码公司(Digital China)的CTO 谢耘先生,正是在他的鼓励和指导之下,我们将日常工作中的点点滴滴积累起来,终于成书。在这里还要感谢神州数码工程院的领导和同事们,包括各项目的开发人员、北京和西安测试部的测试经理、测试工程师和质量经理等,他们的出色工作和工程实践为本书提供了丰富的素材。

本书是为软件测试人员而写的,相信包括测试入门者和测试经理、测试工作者等都可以从



中受益,借此“过五关”就可成为高级测试者,一书在手就可以概览测试理论,借鉴实践经验,改进和提升测试工作。其中的一些章节值得多读几遍,并结合实践深入体会。本书也是为软件组织(软件公司、行业信息中心、专业测试中心、长期工程项目等)的各级管理者(总经理、部门经理、项目经理、技术经理和测试经理等)而写的,帮助他们建立组织级的软件测试体系,提升组织的测试能力,并进而提升整个软件开发和项目实施能力。本书也适用于各类技术人员(需求、设计、开发、运维等)和软件用户,帮助他们理解软件测试、同行评审、性能测试等,他们也是软件测试的参与者。本书也可以作为大学、软件学院及其他各类教育机构的教材,供教师和学生们的教学和学习使用。另外,本书还有一定的普遍意义,体现在系统工程方法运用,系统化思维、组织级体系建设、总结、提炼和创新等方面。这不仅适用于测试和信息技术领域,也适用于其他各类业务、技术和管理领域,值得借鉴。

由于作者水平有限,本书中一定存有少许谬误和问题,敬请读者指正。在此我们要特别感谢北京大学董士海教授、清华大学郑人杰教授、中国科学院顾毓清研究员、北京航空航天大学刘超教授,他们都是中国著名的软件工程专家,他们在百忙中评阅了本书的书稿,并提出了宝贵的建议。

作者 E-mail 联系方式:

徐拥军: xuyjxu@sina. cn

王炯: wangjiongbook@sina. cn

郝进: haojinhj@sina. cn

作者
于 2012 年 2 月,北京



目 录

第 1 章 测试技术引论	1	3.1.5 过管理关	29
1.1 从系统工程角度看测试	1	3.2 测试能力自评和发展	32
1.1.1 从系统工程角度看测试的作用	1	3.3 小结	35
1.1.2 从系统工程观点看软件测试	3	第 4 章 组织级测试体系总体设计	36
1.2 软件测试发展简史	5	4.1 测试体系的内容	36
1.2.1 软件测试的起源和发展历史	5	4.1.1 组织级软件测试体系指的是什么? 这是首先要回答的问题	36
1.2.2 软件测试与质量的关系	7	4.1.2 组织级软件测试体系建设的意义何在? 这是要回答的第二个问题	36
1.2.3 软件测试与 V&V 的关系	7	4.1.3 组织级软件测试体系包括哪些内容? 这是要回答的第三个问题	37
1.3 测试的目的和作用	8	4.2 测试体系建设过程	39
1.4 软件测试 6W 原则	10	4.2.1 组织级测试过程的改进过程	39
1.4.1 WHEN 原则: 尽早地、及时地开始测试	10	4.2.2 组织级软件测试的结论	41
1.4.2 WHAT 原则: 测试对象包括各阶段重要产出物	11	4.3 测试成熟度模型	42
1.4.3 WHO 原则: 全员参与测试	12	4.3.1 TMMi 成熟度级别	42
1.4.4 WHERE 原则: 针对用户最容易遇到的缺陷进行测试	13	4.3.2 TMMi 关键过程域	45
1.4.5 HOW 原则: 综合运用多种测试方法和技术	14	4.4 小结	51
1.4.6 WHY 原则: 测试要适时终止	14	第 5 章 基于迭代的测试过程	53
1.5 小结	15	5.1 测试过程模型	53
第 2 章 测试组织形式	16	5.1.1 V 模型	53
2.1 测试组织形式	16	5.1.2 W 模型	57
2.1.1 项目内测试组形式	16	5.1.3 H 模型	58
2.1.2 测试管理部形式	18	5.1.4 测试过程模型选择策略	59
2.1.3 测试中心形式	19	5.2 基于迭代的测试过程	59
2.2 测试组织形式选择	21	5.3 测试过程监控策略	64
2.3 小结	23	5.3.1 测试目标/策略和计划监控	64
第 3 章 测试人员成长之路	24	5.3.2 项目产出物质量监控	65
3.1 测试人员要“过五关”	24	5.3.3 测试执行顺序监控	66
3.1.1 过心理关	24	5.3.4 软件版本监控	68
3.1.2 过业务关	25	5.3.5 冒烟测试监控	68
3.1.3 过技术关	26	5.3.6 回归测试监控	69
3.1.4 过专业关	27	5.3.7 BUG 处理监控	70
		5.4 小结	72



第 6 章 同行评审过程和方法	73	8.4 建立测试度量分析体系	116
6.1 同行评审概述	73	8.4.1 测试度量分析原则	116
6.2 代码评审和走查	74	8.4.2 测试过程性能基线	117
6.2.1 代码评审	75	8.4.3 项目级测试度量分析过程	119
6.2.2 代码走查	77	8.5 测试度量支持工具示例	121
6.2.3 桌面检查	77	8.5.1 缺陷管理	121
6.3 需求评审和设计评审	78	8.5.2 测试用例管理	121
6.3.1 同行评审小组组成	78	8.5.3 质量预警	121
6.3.2 同行评审过程	78	8.5.4 度量分析	123
6.3.3 评审注意事项	79	8.6 小结	126
6.3.4 同行评审实践	80	第 9 章 自动化测试体系建立	127
6.4 开发人员自测	81	9.1 自动化测试策略	127
6.5 从 CMM 到 PSP/TSP	82	9.2 自动化测试基础建设	128
6.6 同行评审度量	83	9.2.1 测试环境	129
6.7 小结	84	9.2.2 持续集成平台	129
第 7 章 测试用例设计方法	85	9.3 自动化测试框架和工具	131
7.1 白盒测试用例设计	85	9.3.1 自动化测试框架	131
7.1.1 逻辑覆盖测试	85	9.3.2 自动化测试工具	132
7.2 黑盒测试用例设计	91	9.3.3 测试脚本开发	133
7.2.1 等价类划分	91	9.3.4 自己动手开发测试工具	134
7.2.2 边界值分析	93	9.3.5 测试工具 Sm@rtest 介绍	135
7.2.3 因果图	94	9.4 自动化测试实践案例	138
7.2.4 错误推测	98	9.4.1 ESB 平台介绍	138
7.3 测试用例设计的策略	98	9.4.2 ESB 产品自动化测试需求	139
7.4 小结	99	9.4.3 ESB 平台自动化测试方案	139
第 8 章 测试度量与分析过程	100	9.4.4 ESB 自动化测试效果	141
8.1 软件度量概念	100	9.5 自动化测试过程建立	141
8.1.1 度量元	100	9.5.1 自动化测试过程建立	141
8.1.2 度量模型	102	9.5.2 组织级自动化测试体系的建设	142
8.1.3 资源模型	102	9.6 小结	142
8.2 测试计划度量	104	第 10 章 性能测试过程和方法	144
8.2.1 测试规模估计	105	10.1 对性能测试的理解	144
8.2.2 测试工作量估计	106	10.1.1 从理发店模型理解性能	144
8.2.3 测试人数和工期估计	107	10.1.2 理解系统性能度量元	146
8.2.4 测试计划制订	108	10.1.3 性能测试的特点	148
8.3 测试过程度量分析	109	10.2 性能测试规划和设计	149
8.3.1 测试用例度量	109	10.2.1 性能测试目标确定	149
8.3.2 缺陷度量	111	10.2.2 性能测试需求分析	149
8.3.3 缺陷分析	113		



10.2.3 性能测试方案设计	150	11.2.2 核心系统测试需求分析	171
10.3 性能测试实施	153	11.3 核心系统测试用例设计	173
10.3.1 性能测试准备	153	11.3.1 模块测试用例设计	173
10.3.2 测试程序开发	154	11.3.2 流程测试用例设计	184
10.3.3 性能测试执行	157	11.3.3 用画图法设计测试用例	188
10.3.4 系统调优	161	11.3.4 测试用例集管理	189
10.4 性能测试体系建设	165	11.4 核心系统测试执行	190
10.4.1 组织级性能测试体系建设	165	11.4.1 测试数据准备	191
10.4.2 性能测试队伍建设	166	11.4.2 测试执行	192
10.5 小 结	168	11.4.3 测试执行跟踪管理	196
第 11 章 行业核心业务系统测试实践		11.5 核心系统测试效率提升	201
.....	169	11.6 小 结	205
11.1 核心业务系统测试特点	169	附录 1 术 语	206
11.2 核心系统测试需求分析	170	附录 2 参 考 文 献	217
11.2.1 核心系统功能需求介绍	170	跋	219



第 1 章

测试技术引论

作为本书的开篇,本章阐述跟软件测试相关的比较本源、本质和基础的内容。首先从系统工程角度来看软件测试,为软件测试及后面的章节内容引入一个方法论;再回顾软件测试的起源和发展历史,阐明测试与质量、V&V等的关系,以正本清源,辨伪存真;然后从实际工作体会出发,重新认识测试的目的和作用;在此基础上,本章基于系统工程思想,提出了软件测试的基本原则,即“6 W 原则”。这些原则也是关于项目测试实践和测试体系建设的经验教训的总结。



1.1 从系统工程角度看测试

软件测试不是孤立的。它是软件工程的一部分,而组织级软件测试体系也不是孤立的,是整个组织级软件过程和质量体系的一部分。而指导系统(体系)建设和运作的方法论是系统工程。所以,我们先来从系统工程的角度来看软件测试,从系统工程的基本观点(包括整体性观点、综合性观点、科学性观点、关联性观点、实践性观点等)来重新审视软件测试。通过这样的审视,会得到一些新的启示。可以说,本大节的内容为后面的章节确立了方法论,而后面的章节也在自觉地利用系统工程方法来揭示软件测试的奥秘。

1.1.1 从系统工程角度看测试的作用

中国著名的科学家钱学森是系统工程科学的奠基者之一。他在《论系统工程》^①一书中指出,我们把极其复杂的研制对象称为“系统”,即由相互作用和相互依赖的若干组成部分结合成的具有特定功能的有机整体,而且这个“系统”本身又是它所从属的一个更大系统的组成部分。研制这样一种复杂工程系统所面临的基本问题是:怎样把比较笼统的初始研制要求逐步地变为成千上万个研制任务参加者的具体工作,以及怎样把这些工作最终综合成一个技术上合理、经济上合算、研制周期短、能协调运转的实际系统,并使这个系统成为它所从属的更大系统的有效组成部分。这样复杂的总体协调任务不可能靠一个人来完成。这就要求以一种组织、一个集体来代替之前的单个指挥者,对这种大规模社会劳动进行协调指挥需要有一种叫做“系统工程”的科学方法来进行管理。“系统工程”是组织管理“系统”的规划、研究、设计、制造、试验和使用的科学方法,是一种对所有“系统”都具有普遍意义的科学方法。

这样的描述适用于国家尖端技术的研究和实践,同样地适用于软件工程实践。一个大型

^① 钱学森. 论系统工程[M]. 长沙:湖南科学技术出版社,1988.

软件系统是一个“复杂系统”，进行软件系统建设的软件工程组织(包括开发方和用户方)也是一个“复杂系统”。大型软件系统的规划、设计、开发、测试和使用，在系统工程思想、系统工程科学和系统工程技术的指导下进行，才会产生好的效果。软件测试是软件系统工程中的重要环节，软件测试组织是软件工程组织中的重要组织，当然也要接受系统工程思想、科学和技术的指导。

从系统的角度看，软件“系统”是由相互作用和相互依赖的若干组成部分结合成的，具有特定功能的有机整体。这里的特定功能指的是用户需求，用软件技术来描述就是《需求规格说明书》；而“组成部分”是软件子系统、功能模块、构件和服务等，在软件体系架构中设计出组成部分之间的关系。在进行需求分析和系统设计时，要利用系统工程的思想和技术，设计出结构简单、层次分明、模块间耦合度小、可扩展性好的软件系统。按照系统工程的思想，能够设计和开发出高质量的软件系统。

软件测试是现代软件质量保证中的重要技术手段。软件测试是验证一个软件系统是否满足预定的功能需求，达到预定的非功能属性的过程。从系统工程(控制论)的观点来看，软件测试过程就是对正在开发的系统的一个“反馈”(feedback)过程，反馈系统中的错误、缺陷、问题和不符合项等。图 1-1 是在瀑布生命周期模型中，系统构建和反馈过程的图示。其中，右边的箭头指的是系统构建的过程，左边的箭头指的是测试(反馈)的过程。

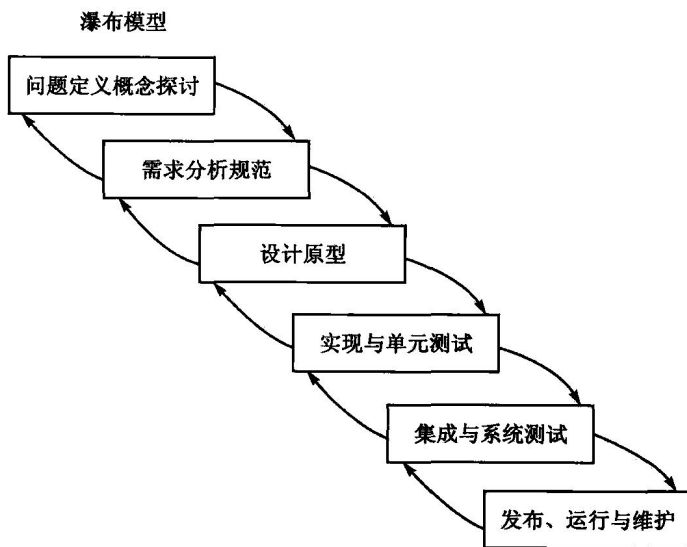


图 1-1 瀑布生命周期模型中的构建和反馈过程

在软件工程中，与质量相关的过程主要是验证过程和确认过程，即 V&V(Verification and Validation)^①。关于验证和确认的定义在 2.1 节中有介绍。简而言之，验证过程就是对每个阶段的产出物相对输入需求做评审或测试；而确认一般是指在开发过程结束时做系统测试。从上面的描述可以看出，无论是验证还是确认，都是测试对开发的反馈过程。在 RUP 模型中，每一次迭代中有一个构建和测试(反馈)的过程，就是为了加强反馈，以便尽早地发现问题，

^① 周伯生，董士海. 软件工程环境引论[J]. 计算机研究与发展, 1986 年.



规避风险^①。

1.1.2 从系统工程观点看软件测试

系统工程的基本观点包括整体性观点、综合性观点、科学性观点、关联性观点、实践性观点等^②。下面以系统工程的基本观点来看软件测试,得出对于软件测试有益的一些启示和原则。

1. 整体性观点

所谓整体性观点即全局性观点或系统性观点,也就是在处理问题时,采用以整体为出发,以整体为归宿的观点。这种观点的要点是:1)处理问题时需遵循从整体到部分进行分析,再从部分到整体进行综合的途径,首先要确定整体目标,并从整体目标出发,协调各组成部分的活动;2)组成系统的各部分处于最优状态,系统未必处于最优状态;3)整体处于最优状态,可能要牺牲某些部分的局部利益和目标;4)不完善的子系统,经过合理的整合,可能形成性能完善的系统。

系统工程的整体性观点给我们的启示是:不能孤立地看软件测试这件事,需要将它放在整个软件工程的大视野来考察;软件测试是软件工程中的一个组成部分,而且它不是一个决定性的部分,所以软件测试不能离开软件工程项目中的其他组织(如分析、设计、开发等)而存在;软件测试要做得好,依赖于其他组织做得好及其他组织对测试的参与和支持,软件测试不能离开项目的整体目标而单独追求自身的目标。同样,软件工程组织也离不开测试。举例来说,在一个工程项目中,开发人员为了赶进度,开发的软件代码质量不高,又没有好好测试(没有缺陷反馈),则到了项目的后期,会出现成堆的缺陷(BUG),很多代码不可用,集成不起来,导致很多工作需要返工。系统工程中有一个基本定律,即“系统性能、功效不守恒定律”,说的是当系统发生变化时,物质、能量守恒,但性能和功效不守恒,且不守恒性是普遍的和无限的。在软件工程领域,软件变更越多,返工越多,系统质量会越差,系统的原有性能和功效会大大减弱。

2. 综合性观点

所谓综合性观点就是在处理系统问题时,把对象的各部分、各因素联系起来加以考察,从关联中找出事物规律性和共同性的研究方法。这种方法可以避免片面性和主观性。同时,现代工程基本上是现成科学的运用,关键在于综合,综合是最大的科学。系统工程就是指导综合研究的理论和方法。

软件测试就是一个综合性很强的工作,需要综合地考虑和运用人、技术、过程、方法、工具、环境等,以求达到好的效果。其综合性体现在下列方面:

(1) 软件测试贯穿软件生命周期的始终,从需求分析阶段就开始了,到系统运行和维护阶段还在继续,直到软件最终停止使用为止。软件测试不单是开发完成之后做系统测试的那一段。

① Ivar Jacobson,等. 统一软件开发过程[M]. 周伯生,等译. 北京:机械工业出版社,2002.

② 钱学森,宋健. 工程控制论(第3版)[M]. 北京:科学出版社,2011,2.



(2) 软件测试不单是测试人员的事。为了做好测试,要综合发挥软件项目中各种角色的作用,还包括用户单位人员、系统最终使用者等。不同的角色进行各种不同的测试(验证和确认)工作,在测试过程中如何组织管理这些人员,是一个很大的问题。

(3) 软件测试要综合利用各种技术、过程、方法和工具来做测试分析、设计和执行工作,以达到较好的效果,不能片面地依赖某一种。

(4) 软件测试的支撑环境,如网络和硬件、系统软件、持续集成系统、测试自动化系统等,本身就是一个综合性的集成系统。

软件测试中的综合,不是简单因素的加法,而是因地制宜地、因势利导地综合考虑和利用上述对象和因素,这才是真正的综合性观点。

3. 科学性观点

所谓科学性观点就是要准确、严密、有充足科学依据地去论证一个系统发展和变化的规律性。不仅要定性、而且必须定量地描述一个系统,使系统处于最优运行状态。

在强调采用定量方法的同时,必须注意以下两个问题:

(1) 必须在定性分析的基础上进行定量分析。定量分析必须以定性分析为前提。只进行定性分析,不能准确地说明一个系统,只有进行了定量分析之后,对系统的认识才能达到一定的深度,结论才能令人信服。然而没有定性分析作指导,定量分析就失去了依据。因此,必须强调要摆正定性分析和定量分析的辩证关系,在处理问题时,一定要在定性分析的基础上应用数学方法,建立模型,进行优化,从而达到系统优化的目的。

(2) 合理处理最优和满意的关系。在处理系统问题时,使系统达到最优比较困难,有时利用满意的概念会使问题得到圆满的解决。因此在处理问题时,要处理好满意和最优的关系。这一原则也是不违背科学性的观点的,因为寻求满意解也是科学。

科学性观点对软件测试的启示在于:对测试过程的管理,要先做定性管理,再做定量管理。通过定性管理打基础,通过定量管理上台阶。在做定性管理的过程中,先收集与测试过程相关的数据,如测试工作量估计数据、测试缺陷数据、测试效率数据、测试速率数据等。在测试数据积累到一定程度时,适当地做一些测试过程数据分析,形成组织的测试过程基准(benchmark),利用它对未来的项目测试进行定量的管理。只有达到了定量管理的程度,才能说软件测试过程具有更好的合理性、科学性、可预测性、可控制性,对于开发人员也才具有更强的说服力和更好的满意度。同样,软件测试也要处理好技术最优和用户满意之间的关系。

4. 关联性观点

所谓关联性观点是指从系统各组成部分的关联中探索系统的规律性。一个系统是由很多因素相互关联而成的,正是这些关联决定了系统的整体特性。因此在处理系统时,必须努力找出系统各组成部分之间的关系,并设法用明确的方式描述这些关系的性质,来揭示和推断系统整体特征。也只有抓住这些联系,用数学、物理、经济学的各种工具建立关系模型才能定性和定量地解决系统问题。不然对一些复杂的问题会感到无从下手。

一个复杂应用系统要用关联性观点来分析、设计和建设,一个软件工程组织及其软件生产线也要以关联性观点来规划和建设。要提高软件测试效果和效率,需要先分析影响软件测试



效率的各种要素及之间的关系,需要用关联性观点;要打造强有力的软件测试团队,要建设组织级软件测试管理体系,都需要用关联性观点。具体的分析和求解过程,将在后面的章节中阐述。

5. 实践性观点

实践性观点就是要勇于实践,勇于探索,要在实践中丰富和完善以及发展系统工程学理论。系统工程不是束之高阁的空头理论,也不是玄妙的数学游戏,它是来源于实践,并指导实践的理论和方法,只有在实践中系统工程才会大有作为,并得到迅速的发展。采用“问题导向”,摒弃“方法导向”是系统工程实践的主要方法。

软件工程、软件测试的实践性都很强,与传统产业(如建筑业、制造产业)相比,其实践时间较短,标准和规范较少,随意性太大,因而需要在实践中不断总结经验教训,探索出适合现实需要的测试过程、方法和技术等。



1.2 软件测试发展简史

1.2.1 软件测试的起源和发展历史

1936年 Alan M. Turing 提出了抽象计算模型“图灵机”,1950年提出了图灵测试,以判断计算机是否拥有智能,这可看作软件测试的起源。1957年 Charles L. Baker 将程序测试从调试中正式分离出来。1958年 Gerald M. Weinberg 建立了第一支软件测试队伍,针对为美国第一个空间飞行计划 Mercury 项目而开发的操作系统进行测试。

1969年 Edsger W. Dijkstra 提出了测试是为了发现 BUG,而不是证明没有 BUG 的著名论断。1972年在北卡罗来纳大学召开了首届软件测试会议,1973年,William C. Hetzel 编辑出版了这次会议的文集《程序测试方法(Program Test Methods)》一书,并专门论述测试和确认问题。与此同时,测试方法不断有新的发展,例如,在1973年 William Elmendorf 提出了用于功能测试的因果图,Fred Gruenberger 对三角形问题进行了探讨;1975年 John Good Enough和 Susan Gerhart 发表了《测试数据选择的原理》的文章,并阐述了决策表的使用;1977年 Edward F. Miller 引入决策路径作为一种结构化测试技术。

1979年,Glenford Myers 出版了《软件测试艺术》^①,这是关于软件测试的第一本专著,奠定了现代软件测试的基础。他将测试定义为“是为发现错误而执行的一个程序或者系统的过程”,并引入了黑盒测试。1981年 Edward Miller 和 William E Howden 合编的论文集《Tutorial, software testing & validation techniques》收集了有关测试和验证的主要论文,前后再版了多次,在这个领域起了重要作用。以后,1982年 Gerald Weinberg 在《再思考系统分析和设计》一书中阐述了迭代式开发和测试,1983年 IEEE 829 软件测试文档标准发布,1984年第一

^① Glenford J. Myers. The Art of Software Testing[J], Wiley Publishing, Inc., 1979.



个测试工业会议(USPDI)在美国召开。

1986年 Paul E. Rook 提出了 V-模型,阐述了软件生命周期中每个阶段与软件测试阶段之间的关系。1993年 Paul Herzlich 提出了 W-模型;1988年 David Gelperin 和 William Hetzel 在他们的 ACM 文章《The Growth of Software Testing》中讨论了四个测试模型和测试的演进。1988年 Cem Kaner 提出探索性测试方法;1990年 Boris Beizer 提出了 BUG 分类法,并提出谁测试的软件愈多,对测试就会愈得心应手。

1996年提出了测试能力成熟度模型 TCMM(Testing Capability Maturity Model)、测试支持度模型 TSM(Testability Support Model)和测试成熟度模型 TMM(Testing Maturity Model)等。其中,1996年美国伊利诺伊理工学院开发了测试成熟度模型(TMM)和 2005年 TMMi 基金会发布的测试成熟度模型集成(TMMi),是评估和改进测试过程的标准。

1996年 James Bach 引入启发式测试策略模型(Heuristic Test Strategy Model),1999年又与 Cem Kaner 等人提出上下文驱动测试(Context-Driven Testing)。1999年 Mark Fewster 和 Dorothy Graham 出版了专著《软件测试自动化》。

从 20 世纪 80 年代开始,测试工具开始盛行。2000年 Martin Fowler 发表了《持续集成》一文;2002年, Kent Beck 提出用测试驱动开发, Ward Cunningham 建立了集成测试框架(Framework for Integrated Test),这是一个自动化测试的开源工具。2007年 ISO 成立了软件测试新标准 ISO 29119 工作组,2008年 Leo van der Aalst 提出软件测试作为服务(Software Testing as a Service, STaaS)。随着新的软件应用领域和软件类型的出现,出现了一些更加专业的测试技术类型,如 Web 应用测试,手机软件测试、嵌入式软件测试、安全测试等。

中国在软件测试方面的起步稍晚一些,但进步很快。例如,从 20 世纪 80 年代起,北京航空航天大学周伯生教授等就主持制订了软件工程和软件测试领域的多项国家标准和军用标准^{①②};1990年左右,北京大学、北京航空航天大学等主持的国家科技攻关项目“青岛工程”推出了一批软件工程和软件测试工具;1990年清华大学郑人杰教授出版的《计算机软件测试技术》^③一书,堪称经典;从 1993年起,北京航空航天大学针对 Lotus 软件产品开展了专业化的软件测试实践;1997年,周伯生教授通过自创的基于过程模型的测试用例自动生成技术及其配套的测试用例生成工具软件,创造了 1 000 万行代码遗留缺陷为零的记录;从 20 世纪 90 年代后期,特别是本世纪初开始,中国的软件从业公司和重点用户单位就建立了专业化的软件测试组织,开展专业化的软件测试实践。

通过历代大师们的努力,软件测试从开发中独立出来,澄清了很多基本的概念,发明了很多测试的过程和方法,并随着现代软件工程和软件实践的发展而不断发展。

① 周伯生,张子让,制定软件工化生产规范的若干问题[J].航空标准化与质量,1986年6期。

② 周伯生,张子让,黄征,张社英.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—1990,计算机软件配置管理计划规范 GB/T 12505—90。

③ 郑人杰.计算机软件测试技术[M].北京:清华大学出版社,1990年。