孙继荣▲著

软件测试自动化

关键技术研究

Ruanjian Ceshi Zidonghua Guanjian Jishu Yanjiu

西南财经大学出版社

软件测试自动化
关键技术研究

Ruanjian Ceshi Zidonghua Guanjian Jishu Yanjiu

孙继荣▲著

西南财经大学出版社

图书在版编目(CIP)数据

软件测试自动化关键技术研究/ 孙继荣著. —成都:西南财经大学出版社,2015. 5
ISBN 978 - 7 - 5504 - 1921 - 6

Ⅰ. ①软… Ⅱ. ①孙… Ⅲ. ①软件—测试—自动化技术
Ⅳ. ①TP311. 5

中国版本图书馆 CIP 数据核字(2015)第 106969 号

软件测试自动化关键技术研究
孙继荣 著

责任编辑:张 岚
助理编辑:高 玲
封面设计:墨创文化
责任印制:封俊川

出版发行	西南财经大学出版社(四川省成都市光华村街 55 号)
网 址	http://www. bookcj. com
电子邮件	bookcj@ foxmail. com
邮政编码	610074
电 话	028 - 87353785 87352368
照 排	四川胜翔数码印务设计有限公司
印 刷	郫县犀浦印刷厂
成品尺寸	170mm × 240mm
印 张	9. 25
字 数	165 千字
版 次	2015 年 5 月第 1 版
印 次	2015 年 5 月第 1 次印刷
书 号	ISBN 978 - 7 - 5504 - 1921 - 6
定 价	48. 00 元

摘　要

随着现代信息技术的飞速发展，软件业竞争日趋激烈，软件系统日益复杂，对软件功能、性能的要求不断提高，同时，软件推出新版本的时间不断缩短。在这种情况下，如何保证软件质量成为企业关注的重点。仅仅依靠以密集劳动为特征的传统手工测试，已经不能满足快节奏软件开发和测试的需求。自动化测试为此提供了成功解决方案。自动化测试是测试体系新发展起来的一个分支，企业实施正确合理的自动化测试能够分担手工测试特别是回归测试的工作量，降低性能测试的难度，从而在保证软件质量的前提下，缩短测试周期，降低软件成本。

本书主要的创新性研究成果和贡献如下：

（1）基于程序切片的自动化测试工具 SATT 的框架模型：该工具分为 PS（切片模块）、TSM（用例集管理模块）、WTAG（白盒测试数据自动生成模块）、WTSR（白盒测试用例集约简模块）、BTAG（黑盒测试数据自动生成模块）、BTSR（黑盒测试用例集约简模块）、RTSR（回归测试模块）以及 SFL（错误定位模块）。该工具以 TSM 为中心，TSM 将保存软件测试各阶段所产生的信息，便于模块间的信息复用。其中 WTAG、WTSR、BTAG、BTSR、RTSR 和 SFL 模块都需要基于程序切片技术实现某些功能，当某个模块所需要的切片信息不在 TSM 中时，由 TSM 调用 PS 生成。

（2）基于复用的测试用例集管理框架：不同测试数据发现错误的能力差别很大，而测试用例集的大小直接决定测试的成本，所以测试数据的自动生成和测试用例集的约简技术是软件测试研究的关键课题。本书提出了一种测试数据的自动生成和测试用例集约简的框架模型，细化了 SATT 中的 TSM 模块功能，便于实现整个测试过程测试用例集的自动化管理和各种测试技术的信息复用。

（3）将基于 I/O 关系的黑盒测试用例集约简技术归结为问题空间（I,O），并提出了两种算法求解（I,O）：①Schroeder 的算法改进。首先，对 I/O 关系进

行关联性分析后划分成若干相关组；其次，对各相关组分别处理，仅对各输出涉及的输入进行组合覆盖，并利用组内元素的关联性把组合覆盖的结果通过公共元素进行水平拼接；最后，把各相关组的结果进行水平拼接。②基于着色问题的黑盒测试用例集约简。将问题空间（I,O）表示为I/O关系图，引入着色问题对（I,O）进行约简，用一个结点代表同色等价类，然后去除包含关系。约简后的问题空间，大大降低了问题本身的复杂度，再利用贪心算法实现约简。两种算法都能生成数量较少的测试用例集并无损测试用例集的检错率。

（4）对Gupta提出的迭代松弛法自动生成测试数据进行了五步改进：基于谓词切片的路径可达性分析，利用谓词切片构造标准化谓词函数，对含有非线性谓词的路径进行相容性分析，改进非线性谓词函数进行线性算术表示的方法，最后只对相容的约束系统进行最小二乘法求解。改进后的迭代松弛法能够避免对不可达路径求解的过程，简化迭代过程中线性约束系统构造的方法，加速含有非线性谓词的路径求解的过程，减少迭代次数，避免由于约束系统构建不当造成方程无解而错误得出路径不可达的结论。

（5）提出了两种将程序切片技术和测试信息相结合的软件错误定位策略：①渐增式方法主要由代码优先级分配、精炼法和扩充法三个算法组成，可自动进行错误定位。②二分法主要由针对TBFL的测试用例集约简、诊断矩阵优化和二分法错误定位三个步骤组成，是个交互式的半自动系统。与同类TBFL工具相比较，两种策略都能更快速更准确地定位到错误。

（6）将测试用例优先级策略和测试用例集约简技术相结合，提出了3种回归测试用例集优化策略：渐增式约简算法BU、精简式约简算法BD、测试用例优先级策略BUP。同时定义了一个新的指标ARRS来衡量一个优化后的回归测试用例集按序执行对需求覆盖的平均速率。与其他启发式算法H,GE,GRE比较，BD总是能取得最高的约简度，BU能取得近似最好的约简度和最高的ARRS值。

目　录

1 绪论

1.1 课题研究的背景和意义

软件测试是根据软件开发阶段的规约或程序的内部结构精心设计测试用例，并用这些用例去运行程序发现错误的过程。它的目标是以较少的测试用例、时间和人力找出软件中潜在的各种错误和缺陷，以确保系统的质量。软件测试在软件的整个开发过程中占有非常重要的地位，是保证软件质量、提高软件可靠性的关键。软件测试工作的好坏，将直接决定软件产品的质量。大量统计资料表明，开发过程中40%~70%的资源都要用于测试，甚至更多[1]。近年来，随着软件应用领域的不断拓展，软件设计技术的不断发展，软件复杂度不断提高，以及软件开发规模越来越大，处理的问题愈来愈复杂，传统的软件测试技术和方法以及测试工具已无法满足大型的、复杂的软件测试需要，软件测试技术需要有革新性的发展。软件测试自动化技术正是在这样的背景下受到了密切的关注。

使用软件测试自动化技术能完成许多手工测试无法实现或难以实现的测试。正确、合理地实施自动化测试，能够快速、彻底地对软件进行测试，从而提高软件质量，节省经费，缩短产品发布周期。另外，自动化测试还能排除一些人为的因素（如遗漏、手误等）。软件测试自动化技术成为软件测试发展的必然趋势，成为近年来软件测试的重要研究方向。因此，开发有效、可重复、操作简便的自动化测试工具是非常有价值的。

本书中对软件测试自动化技术的研究正是在这样的背景下展开的，自动化软件测试技术展现出的强大发展潜力使之成为软件测试研究的重要内容。如何提高自动化软件测试的性能，扩展自动化软件测试的功能，提高自动化软件测试的实用性和软件产品的可靠性，是自动化软件测试需要解决的主要问题。本

书对软件测试自动化中的一些关键技术和方法进行了研究，将程序切片技术引入软件测试的各个方面，并分别实施贪心算法、启发式算法、图着色方法、迭代松弛法、二分法等不同的研究方法。对软件测试自动化的各种技术和实现进行改进和创新，对提高软件测试的自动化程度和保证自动测试的有效性都有极大的现实意义和研究价值。

本书的研究工作获得了计算机信息系统安全体系结构研制项目（863 项目：2002AA144020）、四川省软件重点实验室开放课题基金 2004—2005、四川省青年软件创新工程项目（04hj027－027）、四川省重点科技攻关项目（05GG021-003-2）以及国家中小企业创新基金（06C26225101730）的支持。

1.2 国内外的研究现状和发展动态

1.2.1 软件测试技术和软件测试自动化技术

在谈到软件测试时，许多人都引用 Myers 在《The Art of Sotfware Testing》一书中的观点[2]：

（1）软件测试是为了发现错误而执行程序的过程。

（2）测试是为了证明程序有错，而不是证明程序无错误。

（3）一个好的测试用例在于它能发现至今未发现的错误。

（4）一个成功的测试是发现了至今未发现的错误的测试。

从 20 世纪 90 年代开始，产业界开始意识到被动的以监测和发现错误为目的的软件测试无法避免软件开发过程中由于软件需求和设计等方面的缺陷所带来的巨大风险，所以在整个产业界开始从软件质量控制（SQC，Sotfware Quality Control）转移到软件质量保证（SQA，Sotfware Quality Assurance），从而使软件测试从单纯的缺陷检测和发现覆盖到整个软件开发过程，同时软件测试流程和软件测试技术也成为独立的研究方向。

传统的测试方法是基于 V 模型来完成的，V 模型是最具代表意义，也最广为人知的软件测试生命周期模型[3]。它最早由 Paul Rook 在 20 世纪 80 年代后期提出，旨在改进软件开发的效率和效果。它包含测试过程，完整的开发流程如图 1-1 所示，在 V 模型中，“V”形象地描述了各阶段的活动呈“V”字形排列，同时也有确认（Validation）、验证（Verification）的意思。测试分为单元测试、集成测试、系统测试、验收测试，所有的测试都是在系统编码阶段之后开始的。V 模型的价值在于它非常明确地标明了测试过程中存在的不同级

别，并且清楚地描述了这些测试阶段和开发过程各阶段的对应关系。但是，V模型本身的缺陷在应用时带来了很多不便，例如，测试过程基于严格的开发步骤、测试滞后、系统测试与需求分析相脱离等。

W模型由Paul Herzlich于1993年提出，相对于V模型，W模型增加了软件开发中各阶段应同步进行的验证和确认活动。如图1-2所示，W模型由两个V模型组成，图中明确表示出了测试与开发的并行且相对独立的关系[4]。

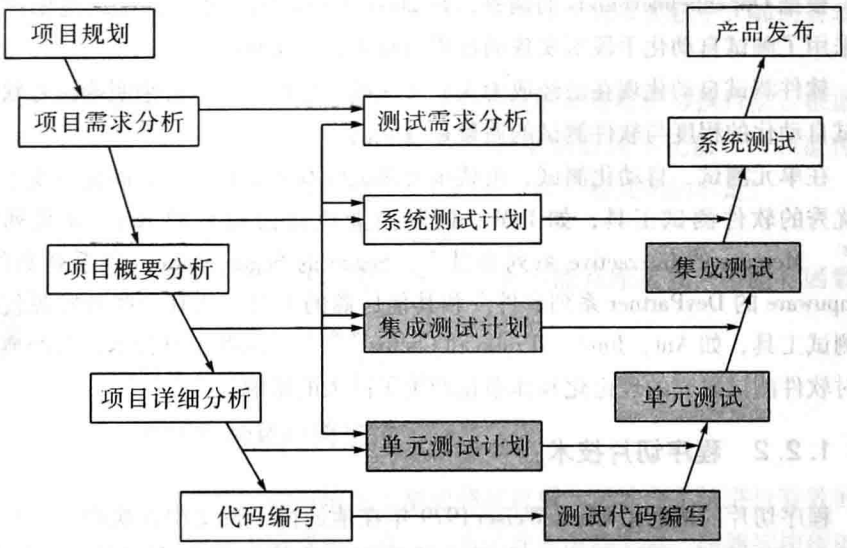

图1-1　基于V模型的开发过程和测试过程

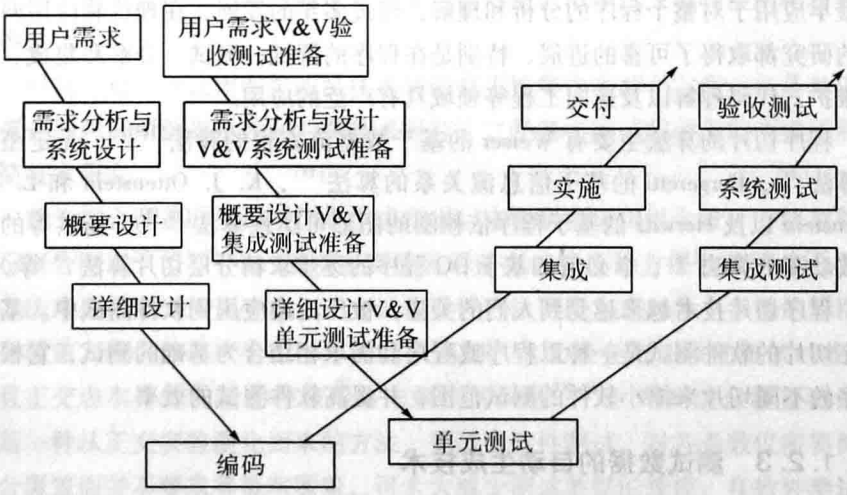

图1-2　W测试模型

作为软件质量保证的一个重要手段，软件测试的发展趋势是：软件测试自动化、测试方法工具化、测试人员专业化、测试部门独立化、测试对象精细化，多种测试工具和手段结合，共同提高软件测试效率。实现软件测试自动化的大潮已经不可逆转。目前，软件测试自动化主要集中在软件测试流程的管理自动化和动态测试的自动化（如功能测试自动化和性能测试自动化等），还有少部分的静态测试（如代码走查），它们常常比较容易从开发过程剥离出来。

根据 OppenheimerFunds 的调查，在 2001 年前后的 3 年中，全球范围内由于采用了测试自动化手段所实现的投资回报率高达 1500%。

软件测试自动化现在已经成为人们关注的一个焦点，但必须明确注意软件测试自动化的程度与软件测试的质量是独立的。

在单元测试、自动化测试、负载压力测试以及测试管理等方面也涌现了大量优秀的软件测试工具，如 IBM（国际商业机器公司）的 Rational 系列套件[5]、Mercury 的 Interactive 系列套件[6]，Segue 的 Segue Software[7] 系列套件，Compuware 的 DevPartner 系列套件[8]和其他厂商的工具，还有一些开发源代码的测试工具，如 Ant，Junit，JProbe 和 Cactus[9,10] 等。这些工具技术以及经典论著对软件测试研究的理论化和体系化产生了巨大的影响。

1.2.2 程序切片技术

程序切片的概念由 Mark Weiser1979 年在他的博士论文中首次提出，后来他又在 1981 年和 1984 年进一步阐述了程序切片的思想和实现算法[11,12,13]。Weiser 声称一个切片与人们在调试一个程序时所做的智力抽象相对应。程序切片最早应用于对整个程序的分析和理解，经过多年的发展，在理论和应用两方面的研究都取得了可喜的进展，特别是在程序的调试、测试、分解和集成、软件维护、代码理解以及逆向工程等领域具有广泛的应用。

程序切片的算法主要有 Weiser 的基于数据流方程的算法[11-13]、无定型切片算法[14]，Bergeretti 的基于信息流关系的算法[15]，K. J. Ottenstein 和 L. M. Ottenstein 以及 Horwitz 的基于程序依赖图的图形可达性算法[16,17]、杨洪等的基于波动图的算法[18]、李必信的基于 OO 程序的逐步求精分层切片算法[19]等。

程序切片技术越来越受到人们的关注，被广泛地应用到软件测试中。基于程序切片的软件测试是一种以程序或程序和需求相结合为基础的测试，它根据程序的不同切片来缩小软件的测试范围，并提高软件测试的效率。

1.2.3 测试数据的自动生成技术

设计和生成有效的测试数据是决定软件测试质量的重要因素之一。一个有

效的测试数据设计方法可以生成高质量的测试数据，并尽可能降低测试数据的总数，从而降低软件开发成本。

测试数据的自动生成将有效地减轻测试人员的劳动强度，节省测试时间和软件开发成本，拓展测试人员的能力。根据估算，对于一个典型的大型软件项目，若能自动生成测试数据，则能节省整个软件开发费用的4%[2,20]。多年来，人们对它进行了广泛而深入的研究，取得了许多研究成果，使之在功能测试、结构测试、性能测试等领域得到广泛的应用。然而在测试实践中，如何自动生成高质量的测试数据仍然面临许多困难。

目前在测试数据自动生成领域中，较为常见的算法有符号执行法、根据需求规约随机产生法[24]、迭代松弛法[25,26,27]、分支函数最小化法[28,29]、遗传算法[24,30-33]等。在进行测试数据生成时，它们还存在一些共同的不足：

（1）不能对大规模程序进行测试数据生成。

（2）对生成测试数据的程序限制较多，如不能有库函数，不能有函数调用等。

（3）面向对象的程序测试数据生成困难等。

1.2.4 测试用例集约简技术

在软件测试过程中，如何精选少量的测试数据，对软件系统进行有效的测试，是软件测试研究的关键课题。现在，测试理论渐趋完善，而测试用例的选择问题却仍处在研究阶段，怎样选取测试用例才能够使被测软件得到充分的测试，在测试时选择不一样的测试序列是否会对测试结果产生影响等问题还在研究中。

目前，对于测试用例集约简技术的研究主要集中在两个方面：一是基于组合覆盖的测试用例集生成和约简技术研究；二是基于测试需求集的测试用例集约简技术研究。

基于组合覆盖的测试用例集生成技术，主要研究应用组合设计方法对软件进行组合测试[22,23,31,34-37]。当今主要是运用正交试验设计方法和两两组合覆盖的方法来进行测试用例生成。正交试验设计方法可以实现对各个参数的等概率组合覆盖，但仍然会产生相当多的测试用例，这无疑会增加软件测试的成本，并且正交表本身的构造还存在很多未解决的问题[38-40]。两两组合覆盖测试方法是一种从正交实验演化而来的方法，更适合软件测试，对各参数仅需要两两组合覆盖而并不要求等概率覆盖，可大大减少测试数据的规模，在软件测试领域已得到成功运用。但是，由于组合测试数据生成是一个NP问题，大多数方

法只能提供近似解，每种方法在特定情况下都存在不足[55]，并且仅仅要求两两组合覆盖不能保证测试的充分性。Schroeder 提出利用 I/O 关系进行测试用例集约简，并证明约简后的测试集保证同初始集一样的检错能力[41,42]。但他给出的三种算法要么复杂度太高，要么约简度太低。综合考虑测试用例的规模、覆盖能力以及测试费用，基于 I/O 关系的测试用例集约简技术具有非常高的价值。

基于测试需求集的测试用例集约简技术，一般是根据某种测试准则得到对应的测试需求集，然后针对各需求分别产生测试用例，形成初始的测试用例集，再运用各种算法约简，得到简化的测试用例集覆盖相同的需求集。约简方法主要有贪心算法[43,44]，H 算法[45]，GE，GRE 算法[74-77]和整数规划方法等。

1.2.5 回归测试自动化技术

回归测试实际上是重复已经存在的测试，这是自动化测试最有应用潜力和最能体现价值的地方。每次构建完成后执行自动化的回归测试，以验证被测试系统的改变是否影响了系统的其他功能。

现有的回归测试用例集优化技术主要有三种：测试用例选择技术[114,116,120]、测试用例集约简技术[45,75-77,119]和测试用例集优先级技术[121,126,127]。

1.2.6 软件错误定位技术

软件错误定位是个复杂而耗时的工作，常用的方法是利用调试工具进行断点设置跟踪程序的执行并分析。程序切片可以将出错点锁定在一定区域内，有助于程序排错，并且可以自动化实现。Weiser 在文献［46］中首先将程序切片引入软件错误定位，通过错误变量的程序切片来排除无关变量以缩小错误的查找范围。李必信在其博士论文中进一步提出首先构建输入变量的前向切片，然后构建出错变量的后向切片，在两者的交集中寻找错误[19]。文献［47,48,49］中提出了一种基于执行切片的技术，这种方法在定位某些错误上非常有效，但是如果错误代码既出现在成功切片又出现在失败切片中时，该方法等价于检查全部的代码，并且这种情况出现的概率比较高。Jones 等清楚地阐明了测试过程和错误诊断的关系，但很少见到有文献将测试过程信息用于错误诊断中[50]。本书在程序切片的基础上，充分利用测试过程提供的信息，对代码含错的可能性赋予不同的优先级进行错误定位。

1.3 本书的主要研究内容以及创新点

软件测试是一项费时、费力并且单调乏味的活动，测试人员需要设计、执行、分析大量的测试用例，测试用例的生成和测试用例集的组成直接影响软件测试的成本和质量。通过对以上国内外现状的分析，本书主要在以下几个方面进行了深入研究：

（1）提出了一种基于程序切片技术的自动化软件测试工具架构。该工具分为切片模块、用例集管理模块、白盒测试数据生成模块、白盒测试用例集约简模块、黑盒测试数据自动生成模块、黑盒测试用例集约简模块、回归测试模块以及错误定位模块。该工具以测试用例集管理模块为中心，负责管理和维护测试各阶段产生的大量信息，便于在以后的阶段中复用这些信息。

（2）提出了一种测试数据的自动生成和测试用例集约简的框架模型，便于在整个测试过程实现测试用例集的自动化管理。不同测试数据发现错误的能力差别很大，而测试用例集的大小将直接决定测试的成本。所以测试数据的自动生成和测试用例集的约简技术是软件测试研究的关键课题。

（3）将基于I/O关系的黑盒测试用例集约简技术归结为问题空间（I,O），该技术无损用例集的检错能力，在引入程序切片技术分析程序的I/O关系后，提出了两种算法对（I,O）进行约简：①改进P. J. Schroeder的算法；②引入着色问题首先对问题空间（I,O）进行约简，进而利用贪心算法实现约简。

（4）引入程序切片技术解决面向路径的测试数据自动生成问题，给出了基于程序切片技术的测试数据自动生成框架。改进Gupta提出的迭代松弛法自动生成测试数据：首先对路径进行静态可达性分析，然后利用谓词切片构造标准化谓词函数，继而对含有非线性谓词的路径进行相容性分析，并改进对非线性谓词函数进行线性算术表示的方法，最后只对相容的约束系统进行最小二乘法求解。通过以上五个步骤的改进，能够避免迭代松弛法对不可行路径求解的过程，同时加速含有非线性谓词的可行路径求解的收敛过程。

（5）将程序切片技术和测试信息相结合提出了两种软件错误定位策略：渐增式方法和二分法。渐增式方法主要由代码优先级分配、精练法和扩充法三个算法组成，可自动进行错误定位。二分法主要由针对TBFL的测试用例集约简、诊断矩阵优化和二分法定位三个步骤组成，是个交互式的半自动系统。

（6）在回归测试中，将测试用例优先级技术和测试用例集约简技术相结