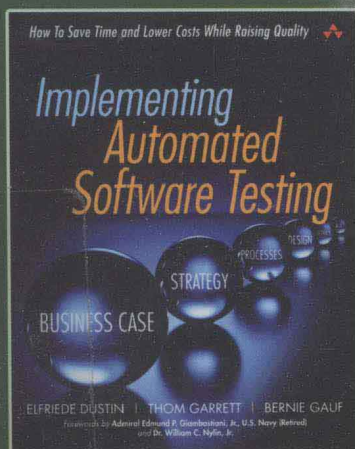


# 自动化软件测试 实施指南

Implementing Automated Software Testing  
How to Save Time and Lower Costs While Raising Quality



[美] Thom Garrett 著  
Bernie Gauf



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

国外计算机科学教材系列

# 自动化软件测试 实施指南

(英文版)

Implementing Automated Software Testing  
How to Save Time and Lower Costs While Raising Quality

Eifriede Dusitin

[美] Thom Garrett 著

電子工業出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书讲解什么是自动化测试,为什么要进行自动化测试,以及实施自动化测试过程中的各种方法和需要注意的问题。全书分为两个部分,第一部分中作者以自身的经验诠释了自动化测试的重要性以及好处;第二部分具体讲解了自动化测试中的6个要点,包括需求收集、制定自动化测试策略、测试自动化软件测试框架、持续跟踪过程并调整、实施自动化测试过程以及恰当的人员安排。附录中还给出了具体的自动化测试实例。

本书适合作为软件测试自动化相关课程的双语教材,也适合软件测试人员、质量评估人员、项目管理人员和软件开发人员作为参考用书。

Original edition, entitled **Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality**, 9780321580511 by Elfriede Dustin, Thom Garrett, Bernie Gauf, published by Pearson Education, Inc, publishing as Addison-Wesley, Copyright©2009 Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. China edition published by PEARSON EDUCATION ASIA LTD., and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, Copyright©2011.

This edition is manufactured in the People's Republic of China, and is authorized for sale only in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版专有出版权由 Pearson Education (培生教育出版集团) 授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书在中国大陆地区生产,仅限在中国大陆发行。

本书贴有 Pearson Education (培生教育出版集团) 激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2011-1030

### 图书在版编目(CIP)数据

自动化软件测试实施指南 = Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality: 英文 / (美) 达斯廷 (Dustin, E.), (美) 加勒特 (Garrett, T.), (美) 高夫 (Gauf, B.) 著.

北京: 电子工业出版社, 2011.3

ISBN 978-7-121-12950-6

I. ①自… II. ①达… ②加… ③高… III. ①软件-测试-指南-英文 IV. ①TP311.5-62

中国版本图书馆CIP数据核字(2011)第024628号

策划编辑: 马 岚

责任编辑: 许菊芳

印 刷: 三河市鑫金马印装有限公司

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路173信箱 邮编: 100036

开 本: 787 × 980 1/16 印张: 22.75 字数: 510千字

印 次: 2011年3月第1次印刷

定 价: 49.00元

凡所购买电子工业出版社的图书有缺损问题, 请向购买书店调换; 若书店售缺, 请与本社发行部联系。联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zltts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

# 出版说明

21世纪初的5至10年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择和自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育集团、麦格劳-希尔教育集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

## 教材出版委员会

主 任	杨芙清	北京大学教授 中国科学院院士 北京大学信息与工程学部主任 北京大学软件工程研究所所长
委 员	王 珊	中国人民大学信息学院院长、教授
	胡道元	清华大学计算机科学与技术系教授 国际信息处理联合会通信系统中国代表
	钟玉琢	清华大学计算机科学与技术系教授、博士生导师 清华大学深圳研究生院信息学部主任
	谢希仁	中国人民解放军理工大学教授 全军网络技术研究中心主任、博士生导师
	尤晋元	上海交通大学计算机科学与工程系教授 上海分布计算技术中心主任
	施伯乐	上海国际数据库研究中心主任、复旦大学教授 中国计算机学会常务理事、上海市计算机学会理事长
	邹 鹏	国防科学技术大学计算机学院教授、博士生导师 教育部计算机基础课程教学指导委员会副主任委员
	张昆藏	青岛大学信息工程学院教授

# Foreword

---

*by* Admiral Edmund P. Giambastiani, Jr.

Today, the world turns over so rapidly that you have to build in a culture of change and innovation on a day-to-day basis. Innovation is an every-single-day part of a soldier, sailor, airman, Marine, or Coast Guardsman's life. Over the course of my military career, I was fortunate to see and experience the dramatic impact innovation has on the war fighter. One area where there has been tremendous innovation is in the field of information technology. The systems that we deploy today are now comprised of millions of lines of software, computer processor speeds we thought were unimaginable a decade ago, and networks that provide extraordinary bandwidth.

Despite these innovations, the need to respond to emerging threats is greater than ever, and the time in which we need to respond continues to decrease. From an information technology perspective, this means we need to be able to make software changes and field the associated capability improvements more rapidly than ever before. Rapidly but effectively testing changes is vital; however, for many programs more than 50% of the schedule is currently devoted to testing.

Innovative Defense Technologies (IDT) has taken the lead on providing an innovative solution to testing that I believe is needed in order for us to keep pace with an ever changing threat. With this book, *Implementing Automated Software Testing*, they have developed a guide that can help implement successful automated software testing programs and efforts. This book includes experience-based case studies and a thorough dissection of automated software testing issues and solutions. This book articulates how to develop the business case for automated software testing and provides a lifecycle approach to automated software testing programs. With *Implementing Automated Software Testing*, IDT is

providing timely and necessary material that allows responsible parties to implement an effective automated software testing program.

Admiral Edmund P. Giambastiani, Jr.  
United States Navy (Retired)  
Vice Chairman, Joint Chiefs of Staff (2005–2007)

# Foreword

---

by Dr. William Nylin, Jr.

When I first began developing software systems in the mid-1960s, testing was primarily a programmer's responsibility; end users validated a relatively small sample of test cases. During the next two decades, more time was spent on testing, but there was significant test case overlap between programming staff testing and end users or a specific testing group. The redundant testing was expensive and delayed project implementation. When errors were discovered, the time lag for correction and revalidation was expensive. Today, a significant portion of the time and cost to market for software products is spent on testing. With the increasing complexity and size of the software included in products, my expectation is that the amount of testing required in the future will only continue to increase. Vast improvements in testing technologies are required, and automated software testing is one of the most promising answers.

The purpose of automated software testing is to increase testing efficiencies via effective use of time and resources, to allow for increased test permutations and combinations, as needed, to avoid test execution redundancy while increasing test coverage, and to allow for automated results analysis, resulting in increased quality and reliability of the software within the same or a reduced testing time frame. IDT's book *Implementing Automated Software Testing* provides extensive technical guidance for implementing an effective automated software testing program. The book provides experience-based automated software testing recommendations and solutions applicable to software testing programs across the board. Applying the automated software testing best practices and guidelines provided in this book will help improve your testing program and ultimately support your business in delivering software products on time, on budget, and with the highest quality. In addition, the book gives practical and

realistic advice on how to compute your return on investment for automated testing solutions. It helps the user understand where to best utilize automated testing and when it may not be cost-effective.

Finally, an additional advantage of automated software testing is the ability to formally audit the testing process. Section 404 of the Sarbanes-Oxley Act of 2002 (SOX 404) requires, as of 2004, that each annual report of a public company include a report by management on the company's internal control over financial reporting. Furthermore, the company's external auditors are required to attest to management's assessment. Management information systems are perhaps the most critical components of internal control systems. Thus, the ability to have an independent audit of the testing processes for new systems can be critical for future large-system development and implementation.

Dr. William Nylin, Jr.  
Executive Vice Chairman and Director  
Conn's, Inc.

# Preface

---

Is your test automation strategy a losing proposition? Are you soured on the notion of automated software testing based on less than adequate past results? Are your test automation silver bullets missing their mark? Are you disappointed in your test automators? We at IDT<sup>1</sup> have identified a boilerplate solution, strategies, and ideas, all provided in this book, that can help increase the chances of your automated testing success.

Given the arsenal of system and application software testing strategies, techniques, and solutions, automated software testing is one of the most effective practices that if implemented correctly can help increase testing efficiencies and ultimately reduce the testing cost while contributing to increased systems and software quality in terms of faster, broader, and more efficient defect detection.

This book is a guide that can help organizations implement successful automated software testing programs and efforts. The book does not provide gimmicks or magical solutions, as none exist, but it provides experience-based discussions and recommendations. It includes a thorough dissection of automation issues, such as in Part I of the book, where we describe what automated software testing is and is not; why a business case is required for successful automation, including step-by-step instructions for developing one; why to automate and when. Then we summarize why automation often fails and the pitfalls and blunders that can be prevented; we describe the tools that are available to help implement successful automation efforts, with a focus on open-source testing tools. In Part II of the book we present six keys to successfully implementing automated software testing. These are

- Key 1: Know Your Requirements
- Key 2: Develop the Automated Test Strategy

---

1. [www.idtus.com](http://www.idtus.com).

- Key 3: Test the Automated Software Test Framework (ASTF)
- Key 4: Continuously Track Progress—and Adjust Accordingly
- Key 5: Implement AST Processes
- Key 6: Put the Right People on the Project—Know the Skill Sets Required

IDT conducted two separate surveys related to automated software testing with approximately 700 total responses from test professionals all over the world, across organizations that were diverse in size and in what they do. The survey showed two very consistent themes:

- About 70% of survey respondents said they believe automation is high-payoff, but they are generally not sure why to automate and how automation applies to their project.
- Half of the survey respondents also said they felt they lacked the experience, time, or budgets to implement automation.

Most seem to agree: Automated software testing is useful, and an increasing need for it exists. However, the lack of experience seems to be the reason why automation is not implemented more often with a higher success rate. Finding people with the skills for the project is therefore important; a summary of skills required is provided in Chapter 10. For more details on the outcome of this survey, see Chapter 4.

## **Material Coverage and Book Organization**

### **Part I: What Is Automated Software Testing and Why Should We Automate?**

Chapter 1, What Is Effective Automated Software Testing (AST)?, describes what automated software testing is. The definition of *automated software testing* we use throughout this book is the “application and implementation of software technology throughout the entire software testing lifecycle (STL) with the goal to improve STL efficiencies and effectiveness.”

In Chapter 2, Why Automate?, we address this question that is asked so often. Here we discuss the challenges of software testing today and how the time

and cost of software testing can be reduced. Reasons for why to automate, laying the foundation to help build the business case discussed step by step in Chapter 3, are presented here.

In Chapter 3, The Business Case, we define a step-by-step approach to defining the business case, which will cover the business need, the reasons for an automated software testing project, the business benefits (tangible and intangible), an analysis of the expected costs and timescales, an investment appraisal, and return on investment (ROI).

Chapter 4, Why Automated Software Testing Fails and Pitfalls to Avoid, clarifies some of the myths and realities surrounding automated software testing. The goal is for companies and organizations to review the lessons described here and not to repeat them during their automated software testing implementations.

## **Part II: How to Automate: Top Six Keys for Automation Payoff**

Once management has been convinced by the business case that was laid out in Part I of this book and understands the pitfalls to avoid and the realities of automated testing, the next step is to determine how to automate. Part II of the book addresses how to successfully implement the various automated software testing tasks. We have determined that successful automated software testing can be achieved by implementing six top keys, described next.

Chapter 5, Key 1: Know Your Requirements, covers the importance of understanding the requirements before developing an automated testing strategy. Here we discuss approaches to determining the problem we are trying to solve along with how to gather information when requirements are not available.

Chapter 6, Key 2: Develop the Automated Test Strategy, discusses developing an automated testing approach in detailed steps, including test environment considerations, configuration management for automated test scripts, and related artifacts, among others. Here we also discuss what to consider when deciding what to automate and the importance of choosing the right tool, whether open-source, vendor-provided, or in-house-developed.

Chapter 7, Key 3: Test the Automated Software Test Framework (ASTF), covers the importance of understanding testing techniques and documenting test cases as part of automated testing. Automators often forget that documentation is still a vital part of the automated test program. The test case documentation serves as the blueprint for the automated software testing efforts. This chapter describes the importance of tracing test cases back to requirements; the content of the test cases, such as needing to include inputs and expected results;

and how documented test cases become the basis for developing and implementing the automated tests.

Chapter 8, Key 4: Continuously Track Progress—and Adjust Accordingly, addresses the importance of tracking the goal that was set at the outset of the automation program. For example, during the discussion of business case development in Chapter 3 we explain the need for defining goals; in this chapter we discuss how peer reviews, inspections, and various automation and testing metrics can help measure and track progress against those goals.

Chapter 9, Key 5: Implement AST Processes, points out the need for a lightweight process. Some automated testing scripts can be implemented successfully without much process in place, but in order to effectively implement a large automated testing program a lightweight adaptable process should be in place. This chapter discusses a summary of this process, linking back to the details in various chapters.

Chapter 10, Key 6: Put the Right People on the Project—Know the Skill Sets Required, clarifies the skill sets needed for developing automated software testing, for instance, a skill set similar to that of the software development team, which includes requirements analysis, design, software development, and testing. Key 6 points out that although knowledge of testing techniques and analytical skills is important, effective automated software testing implementation requires software development skills. The skills described here parallel the automated testing process described in Chapter 9.

## **Audience**

The target audience of this book is software test professionals such as test managers, leads, and practitioners. It is also geared toward all quality assurance professionals, QA leads, and practitioners. Project managers and software developers looking to improve the effectiveness and quality of their software delivery will also benefit from this book.

# Acknowledgments

---

Thanks to all of the software professionals who have helped support the development of this book. Special thanks go to IDT employees Scott Bindas, Marcus Borch, and Vinny Vallarine, who are contributing authors to this book and whose bios are listed in the back of the book. Their valuable contributions to this effort as a whole have greatly added to the content, presentation, and overall quality of the material. Additional thanks go to IDT employees Pete Bria and Burt LeJune for their editorial contributions to sections of this book.

Additional thanks go to our reviewers who added to the quality and content of this book. They are Joe Strazzere and Jake Brake, both long-time and experienced moderators on the popular software quality assurance site [www.sqaforums.com](http://www.sqaforums.com), Jim Hazen, Rob Sabourin, Professor Jeff Offutt, Dr. Satyam Priyadarshy, and Jeff Rashka, PMP, co-author of *Automated Software Testing*, and *Quality Web Systems*. Their edits and contributions were invaluable.

Also we much appreciate the inputs and efforts of our foreword writers, Admiral Edmund P. Giambastiani, Jr., USN (retired), and Dr. William Nylin, Jr.

We also would like to thank the staff at Addison-Wesley, especially Peter Gordon and Kim Boedigheimer, for all their effective ideas and useful input; also Elizabeth Ryan, John Fuller, and Stephane Nakib for their efforts; and finally our copy editor, Barbara Wood, for her valuable suggestions to help make this book a quality product.

# About the Authors

---

**Elfriede Dustin, Thom Garrett, and Bernie Gauf** work together at Innovative Defense Technologies ([www.idtus.com](http://www.idtus.com)), which specializes in the design, development, and implementation of automated software testing solutions.

**Elfriede Dustin** has authored multiple software testing books and articles based on her many years of actual hands-on automated software testing experience. Elfriede leads IDT's efforts in automated software testing research programs.

**Thom Garrett** has experience in planning, testing, and deployment of complex systems for DoD and commercial applications for companies such as Public Broadcasting Service (PBS), Digital System Resources (DSR), Inc., and America Online (AOL). Thom received a master's degree from the University of San Francisco.

**Bernie Gauf** is the president of IDT. Bernie has been invited to participate in numerous DoD panels associated with the use of new technology, testing conferences, and as a guest speaker to share his insights on automated software testing.

# Contents

---

<i>Foreword by Admiral Edmund P. Giambastiani, Jr.</i>	· 15 ·
<i>Foreword by Dr. William Nylin, Jr.</i>	· 17 ·
<i>Preface</i>	· 19 ·
<i>Acknowledgments</i>	· 23 ·
<i>About the Authors</i>	· 24 ·
<b>I. What Is Automated Software Testing and Why Should We Automate?</b>	<b>1</b>
<b>1. What Is Effective Automated Software Testing (AST)?</b>	<b>3</b>
1.1 Automated Software Testing Definition	4
1.2 Automated Software Testing Recipes	5
1.3 Advances in AST Technologies	8
1.4 Automating Various Software Testing Types	11
1.5 Providing AST-Based Production Support	16
<i>Support Troubleshooting the Production Issue, As Needed</i>	16
<i>Support Identifying the Specific Program Components Affected by the Correction</i>	16
<i>Verify the Accuracy and Quality of the Program Correction</i>	17
<i>Support the Production STR Triage</i>	17
1.6 Automating Standards Assessments	18
Summary	20
Notes	21
<b>2. Why Automate?</b>	<b>23</b>
2.1 The Challenges of Testing Software Today	24
2.2 Reducing the Time and Cost of Software Testing	26
<i>Automated Test Planning and Development—Initial Test Effort Increase</i>	28

<i>Test Data Generation—Test Effort/Schedule Decrease</i>	28
<i>Test Execution—Test Effort/Schedule Decrease</i>	32
<i>Test Results Analysis—Test Effort/Schedule Decrease</i>	33
<i>Error Status/Correction Monitoring—Test Effort/Schedule Decrease</i>	35
<i>Report Creation—Test Effort/Schedule Decrease</i>	37
<i>Other Mitigating Factors to Consider</i>	37
2.3 Impacting Software Quality	38
2.4 Improvements to Your Software Test Program	42
<i>Improved Build Verification Testing (Smoke Test)</i>	43
<i>Improved Regression Testing</i>	43
<i>Multiplatform and Configuration Compatibility Testing</i>	44
<i>Improved Execution of Mundane Tests</i>	44
<i>Improved Focus on Advanced Test Issues</i>	44
<i>Testing What Manual Testing Can't Accomplish</i>	45
<i>Ability to Reproduce Software Defects</i>	45
<i>Enhancement of System Expertise</i>	45
<i>After-Hours "Lights-Out" Testing</i>	45
<i>Improved Requirements Definition</i>	46
<i>Improved Performance Testing</i>	46
<i>Improved Stress and Endurance Testing</i>	47
<i>Quality Measurements and Test Optimization</i>	47
<i>Improved System Development Lifecycle</i>	48
<i>Improved Documentation and Traceability</i>	48
<i>Distributed Workload and Concurrency Testing</i>	49
Summary	49
Notes	50
3. The Business Case	51
3.1 Definition of the Business Case	51
3.2 Identifying the Business Needs	53
<i>Need for Speeding Up the Testing Effort and Increasing Efficiency</i>	53
<i>Need for Decreasing the Testing Cost</i>	54
<i>Need for Applying Test Team Member Skills Most Effectively</i>	55
3.3 Justifying Automation in Terms of Cost and Benefits	55
<i>Estimating ROI</i>	55
<i>Overall Test Automation Savings</i>	57
<i>Test Environment Setup Time Savings</i>	57
<i>Test Development Time Savings</i>	59
<i>Test Execution Time Savings</i>	61
<i>Test Evaluation/Diagnostics Time Savings</i>	62
<i>Other ROI Considerations</i>	63
<i>More ROI Metrics</i>	65
3.4 Risks	65
3.5 Other Considerations	67