

How To Save Time and Lower Costs While Raising Quality



Implementing Automated Software Testing

BUSINESS CASE

STRATEGY

PROCESSES

DESIGN

METRICS

ROI

ELFRIEDE DUSTIN | THOM GARRETT | BERNIE GAUF

Forewords by **Admiral Edmund P. Giambastiani, Jr., U.S. Navy (Retired)**
and **Dr. William C. Nylin, Jr.**

Contents

<i>Foreword by Admiral Edmund P. Giambastiani, Jr.</i>	xv
<i>Foreword by Dr. William Nylin, Jr.</i>	xvii
<i>Preface</i>	xix
<i>Acknowledgments</i>	xxiii
<i>About the Authors</i>	xxv
I. What Is Automated Software Testing and Why Should We Automate?	1
1. What Is Effective Automated Software Testing (AST)?	3
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
2. Why Automate?	23
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

Summary	68
Notes	68
4. Why Automated Software Testing Fails and Pitfalls to Avoid	69
4.1 R&D Does Not Generally Focus on Automated or Manual Testing Efforts	71
4.2 AST Myths and Realities	74
<i>Automatic Test Plan Generation</i>	74
<i>Test Tool Fits All</i>	75
<i>Immediate Test Effort Reduction</i>	76
<i>Immediate Reduction in Schedule</i>	77
<i>Tool Ease of Use</i>	77
<i>Universal Application of AST</i>	77
<i>100% Test Coverage</i>	79
<i>Equating Capture/Playback to AST</i>	80
<i>AST Is a Manual Tester Activity</i>	81
<i>Losing Sight of the Testing Goal: Finding Defects</i>	82
<i>Focusing on System Test Automation and Not Automating Unit Tests</i>	82
4.3 Lack of Software Development Considerations for AST	83
<i>Build Testability into the Application</i>	84
<i>Adhere to Open Architecture Standards</i>	86
<i>Adhere to Standard Documentation Format</i>	87
<i>Document Test Cases in a Standard Way</i>	88
<i>Adhere to Coding Standards</i>	89
<i>Use of OMG's IDL</i>	89
<i>GUI Testing Recommendations</i>	89
<i>GUI Object Naming Standards</i>	91
<i>Library Concept of Code Reuse</i>	91
4.4 The Forest for the Trees—Not Knowing Which Tool to Pick	91
<i>How to Evaluate and Choose a Tool</i>	92
4.5 Lack of Automation Standards across Tool Vendors	94
<i>Sample Automated Test Tool Standards</i>	95
4.6 Lack of Business Case	97
Summary	97
Notes	97
II. How to Automate: Top Six Keys for Automation Payoff	99
5. Key 1: Know Your Requirements	101
5.1 Understand the AST Support Requirements	102
<i>AUT or SUT Requirements</i>	103
<i>AST Framework (ASTF) and Tool Requirements</i>	105

<i>AST Data Requirements</i>	105
<i>Test Environment Requirements</i>	108
<i>AST Process Requirements</i>	112
5.2 Additional Information in Support of AST Requirements	114
<i>Manual Testing Procedure</i>	114
<i>Design Documents</i>	115
<i>Prototypes</i>	115
5.3 When Information Is Not Available	116
<i>Conducting Interviews</i>	118
<i>Further Increasing Your Knowledge Base</i>	120
<i>Developing Requirements Based on a Legacy Application or Legacy System</i>	121
5.4 Start Implementing Your Requirements Traceability Matrix (RTM)	124
<i>Information in an RTM</i>	125
<i>Example RTM</i>	125
Summary	128
Notes	128
6. Key 2: Develop the Automated Test Strategy	129
6.1 The AST Strategy Document	131
6.2 Scope and Automated Test Objectives	132
<i>Deciding Which Tests to Automate</i>	133
<i>Prioritize—Base Automation Requirements on a Prioritized Feature Schedule3</i>	137
<i>Defining the Test Objectives: An Example</i>	138
6.3 Identify the Approach	139
<i>Designing and Developing the Test Cases</i>	139
6.4 Automated Software Test Framework (ASTF)	146
6.5 AST Environment/Configuration	150
<i>Test Configurations</i>	151
<i>Other Test Environment Automated Testing Requirements</i>	153
<i>Automating the Test Environment Management—Automated CM</i>	153
6.6 Automating the RTM	159
<i>Require Standard Test Case Templates That Are Usable for Your Automation Framework</i>	160
<i>Hyperlink the Test Cases</i>	160
<i>Update the Test Case Steps to Include Pass/Fail Results</i>	161
<i>Update the RTM to Include Pass/Fail Results</i>	161
6.7 Automated Defect Tracking	164
Summary	164
Notes	165

7. Key 3: Test the Automated Software Test Framework (ASTF)	167
7.1 Verify That the ASTF Meets Specified Requirements and That Features Behave As Expected	169
<i>Verify That Effective ASTF Development Processes Have Been Applied</i>	169
7.2 Peer-Review All ASTF-Related Artifacts, Including Design, Development, and Test Cases	170
<i>Peer Review As Part of the Software Development Lifecycle</i>	170
<i>Evaluate All ASTF Components</i>	173
<i>Review Test Cases</i>	173
<i>Review Test Logic</i>	175
<i>Review Test Data</i>	176
<i>Review Automated Test Code</i>	178
7.3 Verify Requirements and Coverage	182
<i>Traceability</i>	182
<i>Coverage</i>	183
7.4 Hold a Customer Review	183
Summary	184
Notes	184
8. Key 4: Continuously Track Progress—and Adjust Accordingly	187
8.1 AST Program Tracking and Defect Prevention	188
<i>Conduct Technical Interchanges and Walk-throughs</i>	189
<i>Conduct Internal Inspections</i>	189
<i>Examine Constraints and Associated Risks</i>	189
<i>Implement Risk Mitigation Strategies</i>	190
<i>Safeguard the Integrity of the AST Process and Environments</i>	190
<i>Define, Communicate, and Track Schedules and Costs</i>	191
<i>Track Actions, Issues, and Defects</i>	192
8.2 AST Metrics	192
<i>Percent Automatable or Automation Index</i>	196
<i>Automation Progress</i>	198
<i>Test Progress</i>	199
<i>Percent of Automated Test Coverage</i>	199
<i>Defect Density</i>	201
<i>Defect Trend Analysis</i>	202
<i>Defect Removal Efficiency</i>	203
<i>Automated Software Testing ROI</i>	204
<i>Other Software Testing Metrics</i>	204
8.3 Root Cause Analysis	205
Summary	206
Notes	207

9. Key 5: Implement AST Processes	209
9.1 AST Phases and Milestones	211
9.2 AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs	212
9.3 AST Phase 2: Test Case Design and Development	215
9.4 AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development	216
9.5 AST Phase 4: Automated Test Execution and Results Reporting	217
9.6 AST Phase 5: Program Review and Assessment	218
9.7 Virtual Quality Gates	219
9.8 Process Measurement	220
Summary	221
Notes	222
 10. Key 6: Put the Right People on the Project—Know the Skill Sets Required	 223
<i>Soft Skills</i>	231
10.1 Program Management	232
<i>AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs</i>	233
<i>AST Phase 2: Test Case Design and Development</i>	234
<i>AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development</i>	234
<i>AST Phase 4: Automated Test Execution and Results Reporting</i>	235
<i>AST Phase 5: Program Review and Assessment</i>	235
10.2 Systems Engineering	236
<i>AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs</i>	236
<i>AST Phase 2: Test Case Design and Development</i>	237
<i>AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development</i>	238
<i>AST Phase 4: Automated Test Execution and Results Reporting</i>	238
<i>AST Phase 5: Program Review and Assessment</i>	238
10.3 Software Development	239
<i>AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs</i>	239
<i>AST Phase 2: Test Case Design and Development</i>	240
<i>AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development</i>	240
<i>AST Phase 4: Automated Test Execution and Results Reporting</i>	241
<i>AST Phase 5: Program Review and Assessment</i>	241
10.4 Configuration Management	242
<i>AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs</i>	242

<i>AST Phase 2: Test Case Design and Development</i>	243
<i>AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development</i>	243
<i>AST Phase 4: Automated Test Execution and Results Reporting</i>	243
<i>AST Phase 5: Program Review and Assessment</i>	243
10.5 Quality Assurance	244
<i>AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs</i>	244
<i>AST Phase 2: Test Case Design and Development</i>	244
<i>AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development</i>	245
<i>AST Phase 4: Automated Test Execution and Results Reporting</i>	245
<i>AST Phase 5: Program Review and Assessment</i>	245
10.6 Subject Matter Experts (SMEs)	246
<i>Training</i>	246
Summary	247
Notes	248
Appendices	249
A. Process Checklist	251
A.1 AST Phase 1: Requirements Gathering—Analyze Automated Testing Needs	252
<i>Checklist</i>	252
<i>Products</i>	252
A.2 AST Phase 2: Test Case Design and Development	253
<i>Checklist</i>	253
<i>Products</i>	254
A.3 AST Phase 3: Automated Software Testing Framework (ASTF) and Test Script Development	254
<i>Checklist</i>	254
<i>Products</i>	255
A.4 AST Phase 4: Automated Test Execution and Results Reporting	255
<i>Checklist</i>	255
<i>Products</i>	256
A.5 AST Phase 5: Program Review and Assessment	256
<i>Checklist</i>	256
<i>Products</i>	256
B. AST Applied to Various Testing Types	257
B.1 Security Testing	257
B.2 Soak Testing	261
B.3 Concurrency Testing	263

B.4	Performance Testing	265
B.5	Code Coverage Testing	269
B.6	Unit Testing	271
	Notes	274
C.	The Right Tool for the Job	275
C.1	Requirements Management (RM)	276
	<i>Automating Requirements Management</i>	278
C.2	Unit Testing Frameworks—Example Evaluation	281
	<i>How to Evaluate and Choose a Framework</i>	281
C.3	Configuration Management—Example Evaluation	284
	<i>Benefits of Software Configuration Management (SCM)</i>	285
	<i>The SCM Tools Evaluated</i>	286
C.4	Defect Tracking—Example Evaluation	292
	<i>How to Evaluate and Choose a Defect Tracking Tool</i>	292
	<i>Bugzilla versus Trac</i>	296
	<i>Conclusion: Security Is Job 1</i>	298
C.5	Security Testing	299
	<i>Static Analysis versus Dynamic Analysis</i>	300
	<i>Source Analysis versus Binary Analysis</i>	301
	<i>Application Footprinting</i>	302
	<i>Fuzz Testing or Penetration Testing</i>	302
	<i>Threat Modeling—Prioritizing Security Testing with Threat Modeling</i>	303
	<i>Automated Regression Testing</i>	303
	<i>Wireless Security Assessment Tools</i>	303
C.6	Automated Software Testing Framework (ASTF)—	
	Example Evaluation	306
	<i>Test Case Development</i>	307
	<i>High-Level Languages</i>	307
	<i>Platform Support</i>	307
	<i>Open-Source</i>	307
	<i>Cost</i>	308
	<i>Multiple Process Management</i>	309
	<i>Capturing Test Case Results</i>	309
	<i>Support of Distributed Environment</i>	309
	<i>Future Execution Time</i>	310
	<i>Scalable</i>	310
	<i>Footprint on SUT</i>	310
	<i>E-mail</i>	311
	<i>Conclusion for ASTF</i>	311
	<i>Automated Test Tool Example: Testplant's Eggplant</i>	312

C.7	Other STL Support Tools	316
	<i>Self-testable or Autonomic Computing</i>	318
	Notes	319
D.	Case Study: An Automated Software Testing Framework	
	(ASTF) Example	321
D.1	Key Design Features	323
D.2	Test Manager	325
D.3	More on Automated Test Case and Test Code Generation	326
D.4	Results Reporting	328
D.5	Automated Defect Reporting	328
	Notes	329
	Contributing Authors	331
	Index	333