rockynook
> COMPUTING

2nd Edition

Jamie L. Mitchell · Rex Black

Advanced Software Testing Vol. 3

Guide to the ISTQB Advanced Certification as an Advanced Technical Test Analyst

Jamie L. Mitchell · Rex Black

Advanced Software Testing—Vol. 3

Guide to the ISTQB Advanced Certification as an Advanced Technical Test Analyst

2nd Edition

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
TRUNG TÂM THÔNG TIN THƯ VIỆN
02 - 07
00185

rockynook

Table of Contents

Jamie Mitchell's Acknowledgements .... **v**

Rex Black's Acknowledgements .... **vii**

Introduction .... **xxi**

1 The Technical Test Analyst's Tasks in Risk-Based Testing **1**

1.1 Introduction 2

1.2 Risk Identification 4

1.3 Risk Assessment 7

1.4 Risk Mitigation or Risk Control 10

1.5 An Example of Risk Identification and Assessment Results 14

1.6 Risk-Aware Testing Standard 16

1.7 Sample Exam Questions 18

2 Structure-Based Testing **21**

2.1 Introduction 22

2.1.1 Control Flow Testing Theory 24

2.1.2 Building Control Flow Graphs 25

2.1.3 Statement Coverage 29

2.1.4 Decision Coverage 33

2.1.5 Loop Coverage 37

2.1.6 Hexadecimal Converter Exercise 41
2.1.7 Hexadecimal Converter Exercise Debrief 43
2.2 Condition Coverage 43
2.3 Decision Condition Coverage 47
2.4 Modified Condition/Decision Coverage (MC/DC) 48
2.4.1 Complicating Issues: Short-Circuiting 55
2.4.2 Complicating Issues: Coupling 57
2.5 Multiple Condition Coverage 58
2.5.1 Control Flow Exercise 65
2.5.2 Control Flow Exercise Debrief 66
2.6 Path Testing 70
2.6.1 Path Testing via Flow Graphs 71
2.6.2 Basis Path Testing 77
2.6.3 Cyclomatic Complexity Exercise 82
2.6.4 Cyclomatic Complexity Exercise Debrief 83
2.7 API Testing 84
2.8 Selecting a Structure-Based Technique 93
2.8.1 Structure-Based Testing Exercise Debrief 101
2.9 A Final Word on Structural Testing 107
2.10 Sample Exam Questions 108

3 Analytical Techniques 115

3.1 Introduction 115
3.2 Static Analysis 116
3.2.1 Control Flow Analysis 116
3.2.2 Data Flow Analysis 127
3.2.2.1 Define-Use Pairs 130
3.2.2.2 Define-Use Pair Example 133

3.2.2.3 Data Flow Exercise 139
3.2.2.4 Data Flow Exercise Debrief 139
3.2.2.5 A Data Flow Strategy 140
3.2.3 Static Analysis to Improve Maintainability 144
3.2.3.1 Code Parsing Tools 144
3.2.3.2 Standards and Guidelines 147
3.2.4 Call Graphs 149
3.2.4.1 Call-Graph-Based Integration Testing 150
3.2.4.2 McCabe's Design Predicate Approach to Integration 153
3.2.4.3 Hex Converter Example 156
3.2.4.4 McCabe Design Predicate Exercise 161
3.2.4.5 McCabe Design Predicate Exercise Debrief 162
3.3 Dynamic Analysis 163
3.3.1 Memory Leak Detection 165
3.3.2 Wild Pointer Detection 167
3.3.3 Dynamic Analysis Exercise 168
3.3.4 Dynamic Analysis Exercise Debrief 168
3.4 Sample Exam Questions 169

4 Quality Characteristics for Technical Testing 177

4.1 Introduction 178
4.2 Security Testing 181
4.2.1 Security Issues 182
4.2.1.1 Piracy 182
4.2.1.2 Buffer Overflow 183
4.2.1.3 Denial of Service 184
4.2.1.4 Data Transfer Interception 185
4.2.1.5 Breaking Encryption 185
4.2.1.6 Logic Bombs/Viruses/Worms 186
4.2.1.7 Cross-Site Scripting 187

4.2.1.8 Timely Information 188
4.2.1.9 Internal Security Metrics 192
4.2.1.10 External Security Metrics 194
4.2.1.11 Exercise: Security 195
4.2.1.12 Exercise: Security Debrief 195

4.3 Reliability Testing 196

4.3.1 Maturity 201
4.3.1.1 Internal Maturity Metrics 202
4.3.1.2 External Maturity Metrics 203

4.3.2 Fault Tolerance 207
4.3.2.1 Internal Fault Tolerance Metrics 208
4.3.2.2 External Fault Tolerance Metrics 209

4.3.3 Recoverability 211
4.3.3.1 Internal Recoverability Metrics 213
4.3.3.2 External Recoverability Metrics 214

4.3.4 Compliance 216
4.3.4.1 Internal Compliance Metrics 216
4.3.4.2 External Compliance Metrics 216

4.3.5 An Example of Good Reliability Testing 217

4.3.6 Exercise: Reliability Testing 219

4.3.7 Exercise: Reliability Testing Debrief 219

4.4 Efficiency Testing 220

4.4.1 Multiple Flavors of Efficiency Testing 221

4.4.2 Modeling the System 225
4.4.2.1 Identify the Test Environment 227
4.4.2.2 Identify the Performance Acceptance Criteria 227
4.4.2.3 Plan and Design Tests 228
4.4.2.4 Configure the Test Environment 228
4.4.2.5 Implement the Test Design 229
4.4.2.6 Execute the Test 229
4.4.2.7 Analyze the Results, Tune and Retest 230

4.4.3 Time Behavior 230
4.4.3.1 Internal Time Behavior Metrics 235
4.4.3.2 External Time Behavior Metrics 236
4.4.4 Resource Utilization 238
4.4.4.1 Internal Resource Utilization Metrics 238
4.4.4.2 External Resource Utilization Metrics 239
4.4.5 Compliance 242
4.4.5.1 Internal Compliance Metric 242
4.4.5.2 External Compliance Metric 242
4.4.6 Exercise: Efficiency Testing 242
4.4.7 Exercise: Efficiency Testing Debrief 243
4.5 Maintainability Testing 244
4.5.1 Analyzability 248
4.5.1.1 Internal Analyzability Metrics 250
4.5.1.2 External Analyzability Metrics 250
4.5.2 Changeability 252
4.5.2.1 Internal Changeability Metrics 255
4.5.2.2 External Changeability Metrics 255
4.5.3 Stability 256
4.5.3.1 Internal Stability Metrics 257
4.5.3.2 External Stability Metrics 258
4.5.4 Testability 258
4.5.4.1 Internal Testability Metrics 259
4.5.4.2 External Testability Metrics 260
4.5.5 Compliance 261
4.5.5.1 Internal Compliance Metric 261
4.5.5.2 External Compliance Metric 261
4.5.6 Exercise: Maintainability Testing 261
4.5.7 Exercise: Maintainability Testing Debrief 262

4.6 Portability Testing 263
4.6.1 Adaptability 264
4.6.1.1 Internal Adaptability Metrics 266
4.6.1.2 External Adaptability Metrics 267
4.6.2 Replaceability 268
4.6.2.1 Internal Replaceability Metrics 270
4.6.2.2 External Replaceability Metrics 270
4.6.3 Installability 271
4.6.3.1 Internal Installability Metrics 274
4.6.3.2 External Installability Metrics 275
4.6.4 Coexistence 275
4.6.4.1 Internal Coexistence Metrics 276
4.6.4.2 External Coexistence Metrics 277
4.6.5 Compliance 278
4.6.5.1 Internal Compliance Metrics 278
4.6.5.2 External Compliance Metrics 278
4.6.6 Exercise: Portability Testing 278
4.6.7 Exercise: Portability Testing Debrief 278
4.7 General Planning Issues 280
4.8 Sample Exam Questions 285
5 Reviews 287
5.1 Introduction 287
5.2 Using Checklists in Reviews 292
5.2.1 Some General Checklist Items for Design and Architecture Reviews 296
5.2.2 Deutsch's Design Review Checklist 299
5.2.3 Some General Checklist Items for Code Reviews 301

5.2.4 Marick's Code Review Checklist 304
5.2.5 The Open Laszlo Code Review Checklist 306
5.3 Deutsch Checklist Review Exercise 308
5.4 Deutsch Checklist Review Exercise Debrief 308
5.5 Code Review Exercise 310
5.6 Code Review Exercise Debrief 311
5.7 Sample Exam Questions 316

6 Test Tools and Automation 319

6.1 Integration and Information Interchange between Tools 319
6.2 Defining the Test Automation Project 323
6.2.1 Preparing for a Test Automation Project 324
6.2.2 Why Automation Projects Fail 329
6.2.3 Automation Architectures (Data Driven vs. Keyword Driven) 335
6.2.3.1 The Capture/Replay Architecture 337
6.2.3.2 Evolving from Capture/Replay 341
6.2.3.3 The Simple Framework Architecture 343
6.2.3.4 The Data-Driven Architecture 346
6.2.3.5 The Keyword-Driven Architecture 348
6.2.4 Creating Keyword-Driven Tables 350
6.2.4.1 Building an Intelligent Front End 352
6.2.4.2 Keywords and the Virtual Machine 354
6.2.4.3 Benefits of the Keyword Architecture 356
6.2.4.4 Creating Keyword-Driven Tables Exercise 359
6.2.4.5 Keyword-Driven Exercise Debrief 360
6.3 Specific Test Tools 362
6.3.1 Fault Seeding and Fault Injection Tools 362
6.3.2 Performance Testing and Monitoring Tools 363
6.3.2.1 Data for Performance Testing 365

6.3.2.2 Building Scripts 366
6.3.2.3 Measurement Tools 367
6.3.2.4 Performing the Testing 368
6.3.2.5 Performance Testing Exercise 370
6.3.2.6 Performance Testing Exercise Debrief 370
6.3.3 Tools for Web Testing 373
6.3.4 Model-Based Testing Tools 374
6.3.5 Tools to Support Component Testing and Build Process 380
6.4 Sample Exam Questions 383

7 Preparing for the Exam 387

Appendix

A Bibliography 401

Advanced Syllabus Referenced Standards 401
Advanced Syllabus Referenced Books 401
Advanced Syllabus Other References 402
Other Referenced Books 403
Other References 404
Referenced Standards 407

B HELLOCARMS The Next Generation of Home Equity Lending System Requirements Document 409

I Table of Contents 411

II Versioning 413

III Glossary 415

000 Introduction 417

001 Informal Use Case 418

003 Scope 419

004 System Business Benefits 420

010 Functional System Requirements 421

020 Reliability System Requirements 425

030 Usability System Requirements 426

040 Efficiency System Requirements 427

050 Maintainability System Requirements 429

060 Portability System Requirements 430

A Acknowledgement 431

C Answers to Sample Questions 433

Index 435

Introduction

This is a book on advanced software testing for technical test analysts. By that we mean that we address topics that a technical practitioner who has chosen software testing as a career should know. We focus on those skills and techniques related to test analysis, test design, test tools and automation, test execution, and test results evaluation. We take these topics in a more technical direction than in the volume for test analysts, since we focus on test design using structural techniques and details about the use of dynamic analysis to monitor the current internal state of the system, as well as how it interfaces with the OS and other external entities. We assume that you know the basic concepts of test engineering, test design, test tools, testing in the software development life cycle, and test management. You are ready to mature your level of understanding of these concepts and to apply these advanced concepts to your daily work as a test professional.

This book follows the International Software Testing Qualifications Board's (ISTQB) Advanced Technical Test Analyst syllabus 2012, with a focus on the material and learning objectives for the advanced technical test analyst. As such, this book can help you prepare for ISTQB Advanced Level Technical Test Analyst exam. You can use this book to self-study for this exam or as part of an e-learning or instructor-led course on the topics covered in those exams. If you are taking an ISTQB-accredited Advanced Level Technical Test Analyst training course, this book is an ideal companion text for that course.

However, even if you are not interested in the ISTQB exams, you will find this book useful to prepare yourself for advanced work in software testing. If you are a test manager, test director, test analyst, technical test analyst, automation test engineer, manual test engineer, programmer, or in any other field where a sophisticated understanding of software testing is needed, especially an understanding of the particularly technical aspects of testing such as white-box testing and test automation, then this book is for you.

This book focuses on technical test analysis. It consists of seven chapters, addressing the following material:

1. The Technical Test Analyst's Tasks in Risk-Based Testing
2. Structure-Based Testing
3. Analytical Techniques
4. Quality Characteristics for Technical Testing
5. Reviews
6. Test Tools and Automation
7. Preparing for the Exam

What should a technical test analyst be able to do? Or, to ask the question another way, what should you have learned to do—or learned to do better—by the time you finish this book?

- Recognize and classify typical risks associated with performance, security, reliability, portability and maintainability of software systems.
- Create test plans detailing planning, design, and execution of tests to mitigate performance, security, reliability, portability, and maintainability risks.
- Select and apply appropriate structural design techniques so tests provide adequate levels of confidence, via code coverage and design coverage.
- Effectively participate in technical reviews with developers and architects, applying knowledge of typical mistakes made in code and architecture.
- Recognize risks in code and architecture and create test plan elements to mitigate those risks via dynamic analysis.
- Propose improvements to security, maintainability, and testability of code via static analysis.
- Outline costs and benefits expected from types of test automation.
- Select appropriate tools to automate technical testing tasks.
- Understand technical issues and concepts in test automation.

In this book, we focus on these main concepts. We suggest that you keep these high-level outcomes in mind as we proceed through the material in each of the following chapters.

In writing this book, we've kept foremost in our minds the question of how to make this material useful to you. If you are using this book to prepare for an ISTQB Advanced Level Technical Test Analyst exam, then we recommend that you read Chapter 7 first, then read the other six chapters in order. If you are using this book to expand your overall understanding of testing to an advanced and highly technical level but do not intend to take an ISTQB Advanced Level

Technical Test Analyst exam, then we recommend that you read Chapters 1 through 6 only. If you are using this book as a reference, then feel free to read only those chapters that are of specific interest to you.

Each of the first six chapters is divided into sections. For the most part, we have followed the organization of the ISTQB Advanced Syllabus to the point of section divisions, but subsection and sub-subsection divisions in the syllabus might not appear. You'll also notice that each section starts with a text box describing the learning objectives for the section. If you are curious about how to interpret those K2, K3, and K4 tags in front of each learning objective, and how learning objectives work within the ISTQB syllabus, read Chapter 7.

Software testing is in many ways similar to playing the piano, cooking a meal, or driving a car. How so? In each case, you can read books about these activities, but until you have practiced, you know very little about how to do it. So we've included practical, real-world exercises for the key concepts. We encourage you to practice these concepts with the exercises in the book. Then, make sure you take these concepts and apply them on your projects. You can become an advanced testing professional only by applying advanced test techniques to actual software testing.