SOFTWARE TESTING

Yogesh Singh

CAMBRIDGE UNIVERSITY PRESS
1. Introduction

1.1 Some Software Failures
   1.1.1 The Explosion of the Ariane 5 Rocket
   1.1.2 The Y2K Problem
   1.1.3 The USA Star-Wars Program
   1.1.4 Failure of London Ambulance System
   1.1.5 USS Yorktown Incident
   1.1.6 Accounting Software Failures
   1.1.7 Experience of Windows XP

1.2 Testing Process
   1.2.1 What is Software Testing?
   1.2.2 Why Should We Test?
   1.2.3 Who Should We Do the Testing?
   1.2.4 What Should We Test?

1.3 Some Terminologies
   1.3.1 Program and Software
   1.3.2 Verification and Validation
   1.3.3 Fault, Error, Bug and Failure
   1.3.4 Test, Test Case and Test Suite
   1.3.5 Deliverables and Milestones
   1.3.6 Alpha, Beta and Acceptance Testing
   1.3.7 Quality and Reliability
1.3.8 Testing, Quality Assurance and Quality Control 23
1.3.9 Static and Dynamic Testing 23
1.3.10 Testing and Debugging 24

1.4 Limitations of Testing 24
1.4.1 Errors in the Software Requirement and Specification Document 24
1.4.2 Logical Bugs 24
1.4.3 Difficult to Measure the Progress of Testing 26

1.5 The V Shaped Software Life Cycle Model 26
1.5.1 Graphical Representation 27
1.5.2 Relationship of Development and Testing Parts 27

Multiple Choice Questions 28
Exercises 34
Further Reading 35

2. Functional Testing 37
2.1 Boundary Value Analysis 38
2.1.1 Robustness Testing 43
2.1.2 Worst-Case Testing 44
2.1.3 Robust Worst-Case Testing 46
2.1.4 Applicability 48

2.2 Equivalence Class Testing 63
2.2.1 Creation of Equivalence Classes 63
2.2.2 Applicability 65

2.3 Decision Table Based Testing 81
2.3.1 Parts of the Decision Table 81
2.3.2 Limited Entry and Extended Entry Decision Tables 82
2.3.3 ‘Do Not Care’ Conditions and Rule Count 82
2.3.4 Impossible Conditions 83
2.3.5 Applicability 83

2.4 Cause-Effect Graphing Technique 96
2.4.1 Identification of Causes and Effects 97
2.4.2 Design of Cause-Effect Graph 97
2.4.3 Use of Constraints in Cause-Effect Graph 97
2.4.4 Design of Limited Entry Decision Table 99
2.4.5 Writing of Test Cases 99
2.4.6 Applicability 99

Multiple Choice Questions 102
Exercises 105
Further Reading 108

3. Essentials of Graph Theory 110
3.1 What is a Graph? 110
3.1.1 Degree of a Node 112
3.1.2 Regular Graph 113
3.2 Matrix Representation of Graphs 113
  3.2.1 Incidence Matrix 114
  3.2.2 Adjacency Matrix 114
3.3 Paths and Independent Paths 116
  3.3.1 Cycles 117
  3.3.2 Connectedness of a Graph 117
3.4 Generation of a Graph from Program 123
  3.4.1 Program Graphs 124
  3.4.2 DD Path Graphs 127
3.5 Identification of Independent Paths 144
  3.5.1 Cyclomatic Complexity 144
  3.5.2 Graph Matrices 150

Multiple Choice Questions 159
Exercises 161
Further Reading 163

4. Structural Testing 165

4.1 Control Flow Testing 165
  4.1.1 Statement Coverage 166
  4.1.2 Branch Coverage 167
  4.1.3 Condition Coverage 167
  4.1.4 Path Coverage 167
4.2 Data Flow Testing 173
  4.2.1 Define/Reference Anomalies 174
  4.2.2 Definitions 174
  4.2.3 Identification of du and dc Paths 175
  4.2.4 Testing Strategies Using du-Paths 175
  4.2.5 Generation of Test Cases 176
4.3 Slice Based Testing 197
  4.3.1 Guidelines for Slicing 197
  4.3.2 Creation of Program Slices 198
  4.3.3 Generation of Test Cases 202
4.4 Mutation Testing 212
  4.4.1 Mutation and Mutants 212
  4.4.2 Mutation Operators 216
  4.4.3 Mutation Score 216

Multiple Choice Questions 223
Exercises 226
Further Reading 228

5. Software Verification 230

5.1 Verification Methods 230
  5.1.1 Peer Reviews 231
5.1.2 Walkthroughs 231
5.1.3 Inspections 231
5.1.4 Applications 232

5.2 Software Requirements Specification (SRS) Document Verification 233
5.2.1 Nature of the SRS Document 233
5.2.2 Characteristics and Organization of the SRS Document 233
5.2.3 SRS Document Checklist 235

5.3 Software Design Description (SDD) Document Verification 238
5.3.1 Organization of the SDD Document 239
5.3.2 The SDD Document Checklist 239

5.4 Source Code Reviews 241
5.4.1 Issues Related to Source Code Reviews 241
5.4.2 Checklist of Source Code Reviews 242

5.5 User Documentation Verification 243
5.5.1 Review Process Issues 244
5.5.2 User Documentation Checklist 244

5.6 Software Project Audit 245
5.6.1 Relevance Scale 246
5.6.2 Theory and Practice Scale 246
5.6.3 Project Audit and Review Checklist 246

5.7 Case Study 257

6. Creating Test Cases from Requirements and Use Cases 285
6.1 Use Case Diagram and Use Cases 285
6.1.1 Identification of Actors 286
6.1.2 Identification of Use Cases 287
6.1.3 Drawing of Use Case Diagram 288
6.1.4 Writing of Use Case Description 290

6.2 Generation of Test Cases from Use Cases 292
6.2.1 Generation of Scenario Diagrams 293
6.2.2 Creation of Use Case Scenario Matrix 294
6.2.3 Identification of Variables in a Use Case 295
6.2.4 Identification of Different Input States of a Variable 296
6.2.5 Design of Test Case Matrix 296
6.2.6 Assigning Actual Values to Variables 296

6.3 Guidelines for generating validity checks 316
6.3.1 Data Type 316
6.3.2 Data Range 316
6.3.3 Special Data Conditions 316
6.3.4 Mandatory Data Inputs 316
6.3.5 Domain Specific Checks 316

6.4 Strategies for Data Validity 317
6.4.1 Accept Only Known Valid Data 317
6.4.2 Reject Known Bad Data 322
6.4.3 Sanitize All Data 322
6.5 Database Testing 326

Multiple Choice Questions 331
Exercises 333
Further Reading 334

7. Selection, Minimization and Prioritization of Test Cases for Regression Testing 335

7.1 What is Regression Testing? 335
   7.1.1 Regression Testing Process 336
   7.1.2 Selection of Test Cases 337
7.2 Regression Test Cases Selection 339
   7.2.1 Select All Test Cases 339
   7.2.2 Select Test Cases Randomly 339
   7.2.3 Select Modification Traversing Test Cases 339
7.3 Reducing the Number of Test Cases 340
   7.3.1 Minimization of Test Cases 340
   7.3.2 Prioritization of Test Cases 341
7.4 Risk Analysis 342
   7.4.1 What is Risk? 342
   7.4.2 Risk Matrix 343
7.5 Code Coverage Prioritization Technique 346
   7.5.1 Test Cases Selection Criteria 347
   7.5.2 Modification Algorithm 347
   7.5.3 Deletion Algorithm 352

Multiple Choice Questions 363
Exercises 364
Further Reading 365

8. Software Testing Activities 368

8.1 Levels of Testing 368
   8.1.1 Unit Testing 369
   8.1.2 Integration Testing 370
   8.1.3 System Testing 373
   8.1.4 Acceptance Testing 373
8.2 Debugging 374
   8.2.1 Why Debugging is so Difficult? 374
   8.2.2 Debugging Process 375
   8.2.3 Debugging Approaches 377
   8.2.4 Debugging Tools 378
8.3 Software Testing Tools 379
   8.3.1 Static Software Testing Tools 379
9. Object Oriented Testing

9.1 What is Object Orientation?
9.1.1 Classes and Objects
9.1.2 Inheritance
9.1.3 Messages, Methods, Responsibility, Abstraction
9.1.4 Polymorphism
9.1.5 Encapsulation

9.2 What is Object Oriented Testing?
9.2.1 What is a Unit?
9.2.2 Levels of Testing

9.3 Path Testing
9.3.1 Activity Diagram
9.3.2 Calculation of Cyclomatic Complexity
9.3.3 Generation of Test Cases

9.4 State Based Testing
9.4.1 What is a State Machine?
9.4.2 State Chart Diagram
9.4.3 State Transition Tables
9.4.4 Generation of Test Cases

9.5 Class Testing
9.5.1 How Should We Test a Class?
9.5.2 Issues Related to Class Testing
9.5.3 Generating Test Cases

Multiple Choice Questions
Exercises
Further Reading

10. Metrics and Models in Software Testing

10.1 Software Metrics
10.1.1 Measure, Measurement and Metrics
10.1.2 Applications

10.2 Categories of Metrics
10.2.1 Product Metrics for Testing
10.2.2 Process Metrics for Testing

10.3 Object Oriented Metrics Used in Testing
## Contents

10.3.1 Coupling Metrics 424
10.3.2 Cohesion Metrics 424
10.3.3 Inheritance Metrics 425
10.3.4 Size Metrics 426

10.4 What Should We Measure During Testing? 427
10.4.1 Time 427
10.4.2 Quality of Source Code 428
10.4.3 Source Code Coverage 429
10.4.4 Test Case Defect Density 429
10.4.5 Review Efficiency 429

10.5 Software Quality Attributes Prediction Models 430
10.5.1 Reliability Models 430
10.5.2 An Example of Fault Prediction Model in Practice 437
10.5.3 Maintenance Effort Prediction Model 442

Multiple Choice Questions 446
Exercises 449
Further Reading 451

### 11. Testing Web Applications 453

11.1 What is Web Testing? 453
11.1.1 Web Application versus Client Server Application 453
11.1.2 Key Areas in Testing Web Applications 455

11.2 Functional Testing 456

11.3 User Interface Testing 458
11.3.1 Navigation Testing 458
11.3.2 Form Based Testing 459
11.3.3 User Interface Testing Checklist 461

11.4 Usability Testing 463
11.4.1 What is Usability and Usability Testing? 463
11.4.2 Identification of Participants 464
11.4.3 Development of Usability Testing Questionnaire 465
11.4.4 Setting up Environment for Conducting Test 468
11.4.5 Conducting the Test 469
11.4.6 Analyze the Results and Observations 469

11.5 Configuration and Compatibility Testing 469
11.5.1 Browser Testing 470
11.5.2 Guidelines and Checklist for Configuration and Compatibility Testing 470

11.6 Security Testing 471

11.7 Performance Testing 476
11.7.1 Load Testing 476
11.7.2 Stress Testing 479
11.7.3 Performance Testing Checklist 479

11.8 Database Testing 480

11.9 Post-Deployment Testing 482
12. Automated Test Data Generation

12.1 What is Automated Test Data Generation?
12.1.1 Test Adequacy Criteria
12.1.2 Static and Dynamic Test Data Generation
12.2 Approaches to Test Data Generation
12.2.1 Random Testing
12.2.2 Symbolic Execution
12.2.3 Dynamic Test Data Generation
12.3 Test Data Generation using Genetic Algorithm
12.3.1 Initial Population
12.3.2 Crossover and Mutation
12.3.3 Fitness Function
12.3.4 Selection
12.3.5 Algorithm for Generating Test Data
12.4 Test Data Generation Tools

Multiple Choice Questions
Exercises
Further Reading

Appendix I
Appendix II
Appendix III
References
Answers to Multiple Choice Questions
Index