Continue

Introduction to software testing edition 2

He was awarded the George Mason University Teaching Excellence Award, Teaching with Technology, in 2013. The authors reserve the copyright on all programs, but grant permission for them to be used for educational purposes. With a sidetrip? Page 211, 31-May-2018: “inside the if block that starts on line 24” should be “inside the if block that starts on line 28” Found by Nan Li, Medidata Solutions. “Some older sources say that white-box testing is used for system testing and black-box testing for unit testing.” The correct sentence should be: “Some older sources say that black-box testing is used for system testing and white-box testing for unit testing.” Found by Ananya Dhawan, George Mason University. Short answer: Questions that require a sentence or two to respond. (The student and instructor solutions are correct.) Found by Jordan Rash, George Mason University. Writing Test Plans 12. Page 33, 2.7, 18-May-2018: The first sentence in section 2.7 should say “finding all faults in a program is undecidable.” Found by Nan Li, Medidata Solutions. ★ This program is only available to instructors (with the instructor solution manual). Page 249, 30-Apr-2017: Definition of COR: “or-k” should be “or-||” and “or with no conditional evaluation-j” should be “or with no conditional evaluation-|”. The exercises can be modified as long as the copyright and reference to the footnote remains intact. The authors welcome corrections, updates and additions. Test Implementation 13. New versions of exercises will have the dates marked in red for a few months. Graph Coverage 7.1 Overview 7.2 Graph Coverage Criteria 7.2.1 Structural Coverage Criteria 7.2.2 Touring, Sidetrips, and Detours 7.2.3 Data Flow Criteria 7.2.4 Subsumption Relationships among Graph Coverage Criteria 7.3 Graph Coverage for Source Code 7.3.1 Structural Graph Coverage for Source Code 7.3.2 Data Flow Graph Coverage for Source Code 7.4 Graph Coverage for Design Elements 7.4.1 Structural Graph Coverage for Design Elements 7.4.2 Data Flow Graph Coverage for Design Elements 7.5 Graph Coverage for Specifications 7.5.1 Testing Sequencing Constraints 7.5.2 Testing State Behavior of Software 7.6 Graph Coverage for Use Cases 7.6.1 Use Case Scenarios 7.7 Bibliographic Notes 8. We will be glad to credit any contributors to these programs. Chapter 4 Chapter 5 Page 69, 25-Mar-2018: Last paragraph on the page, 4th sentence should be: It requires a lot of knowledge and skills to use test criteria to design tests, however the MDTD process allows one expert on the criterion to provide designs that can then be turned into automated tests by many testers who are not experts on the criterion. This extensively classroom-tested text takes an innovative approach to explaining software testing that defines it as the process of applying a few precise, general-purpose criteria to a structure or model of the software. A better sentence would be: “Testers often use both approaches to design different characteristics for the same IDM, but it helps to study them separately.” Found by Leron Culbreath, George Mason University. Last update: 26 February, 2016. Page 207, 16-Apr-2019: Exercises, Section 8.2, number 2, the predicate for "X" is missing an "AND" operator: “X = (b ^ d) v (~b ~d)” should be “X = (b ^ d) v (~b ^ ~d)” Found by Stefan Soto, George Mason University. Solution with help by Kesina Baral of George Mason University. Input Space Partitioning 6.1 Input Domain Modeling 6.1.1 Interface-based Input Domain Modeling 6.1.2 Functionality-based Input Domain Modeling 6.1.3 Designing Characteristics 6.1.4 Choosing Blocks and Values 6.1.5 Checking the Input Domain Model 6.2 Combination Strategies Criteria 6.3 Handling Constraints Among Characteristics 6.4 Extended Example: Deriving an IDM from JavaDoc 6.4.1 Tasks in Designing IDM-based Tests 6.4.2 Designing IDM-based Tests for Iterator 6.5 Bibliographic Notes 7. We will be glad to credit any contributors to these exercises. These slides are available for instructors, students, and readers of the book. To redeem an access code, please log in with your personal login. Page 88, 17-Sep-2018: The formula for the number of tests needed for TWC has ‘q’ in the superscript, but it should be ‘Q’. Preface Chapter 1 Chapter 2 Page 20, 14-Feb-2018: The sixth sentence in the fourth paragraph is confusing The sentence currently says: “If the tester only observes parts of the correct portion of the final program state, the failure is not revealed.” A clearer sentence would be: “If the tester only observes parts of the portion of the final program state that are correct, the failure would not be revealed.” Found by Mathias Wiesbauer, George Mason University. Offutt has published more than 165 papers in areas such as model-based testing, criteria-based testing, test automaton, empirical software engineering, and software maintenance. He is Editor-in-Chief of the Journal of Software Testing, Verification and Reliability, helped found the IEEE International Conference on Software Testing, and is the founder of the μJava project. Group discussion: Small groups should discuss a concept. The book was released December 2016 with a copyright date of 2017. Chapter 6 Page 79, 24-Sep-2018: Second paragraph, “next section” should be “next two subsections.” Also, the sentence: “The tester must choose which approach to use.” is misleading. Putting Testing First 4.1 Taming the Cost-Of-Change Curve 4.1.1 Is the Curve Really Tamed? Test Automation 4. Last update: 23 June, 2014. The second edition of Introduction to Software Testing incorporates new features and material, yet retains the structure, philosophy, and online resources that have been so popular among the ... This is the website for the second edition of the book Introduction to Software Testing. We now have Spanish versions of many of the slides, thanks to the generosity of Manuel Núñez and Mercedes G. Found by David Guo, George Mason University. Input Space Partitioning Ch06-ISP.pptx Ch06-ISP-activepptx en español 2nd Edition, Part 2, Ch 6.1-3: Input Domain Modeling Ch 6.1-3 “active” version September 2022 September 2022 Ch06-4-ISPinClassExercise.pptx2nd Edition, Part 2, Ch 6.4: Extended ExampleOctober 2017 Ch 7. 1.1 When Software Goes Bad 1.2 Goals of Testing Software 1.3 Bibliographic Notes 2. This revised second edition significantly expands coverage of the basics, thoroughly discussing test automaton frameworks, and it adds new, improved examples and numerous exercises. Writing Effective Test Oracles 14.1 What Should be Checked? (Note that the original statement is not ambiguous, because only one for statement connects to the end of the while loop, but students often get confused.) Found by the authors, George Mason University. It should be {6}. Chapter 11 Chapter 12 Chapter 13 Chapter 14 References Chapter List Part I: Overview 1. If we model that as a sequence of equalities separated by ORs (e.g., x==1 XOR x==2 XOR x==3), then Predicate Coverage (PC) would not subsume Branch Coverage or Node Coverage. Pages 145-146, 21-Mar-2017: Exercises, Section 7.3, number 8: change “java.util.AbstractList” to “java.util.AbstractList”; change “ListIterator e1 = listIterator();” to “ListIterator e1 = listIterator();” change “ListIterator e2 = ((List) o).listIterator();” to “ListIterator e2 = ((List) o).listIterator();” part (a): change “node number” to “node numbers”; part (b): change “$\parallel$” to “||”; part (b): Remove duplicate label ‘C’. Chapter 9 Page 241, 14-Nov-2017: Exercise 9.1.2, question 5, the BNF should be: I ::= "j" | "s" Found by Upsorn Praphamontripong, University of Virginia. The authors welcome corrections, updates, and additions. These exercises are available for instructors, students, and readers of the book. All programs have been compiled with Java 1.8. The authors welcome corrections, updates and additions. FileChapter or SectionLast Update Part 1. Page 271, 5-June-2018: 9.5.1, fourth paragraph, last sentence: “from this grammar” should be “from this regular expression” Found by SWE 437 student, George Mason University. Logic Coverage 9. Graph Testing Ch07-1-2-overviewGraphCoverage.pptx Ch07-1-2-overviewGraphCoverage-active.pptx en español 2nd Edition, Part 2, Ch 7.1-2: Graph Coverage Criteria Ch 7.1-2 “active” version October 2016 October 2016 Ch07-3-sourceCode.pptx Ch07-3-sourceCode-active.pptx en español 2nd Edition, Part 2, Ch 7.3: Graph Coverage for Source Code Ch 7.3 “active” version December 2016 December 2016 Ch07-4-design.pptx2nd Edition, part 2, Ch 7.4: Graph Coverage for Design ElementsDecember 2016 Ch07-5-spec.pptxen español2nd Edition, Part 2, Ch 7.5: Graph Coverage for SpecificationsOctober 2017 Ch07-6-useCases.pptxen español2nd Edition, Part 2, Ch 7.6: Graph Coverage for Use CasesDecember 2016 Ch 8. Input Space Partitioning 7. Page 222, 31-May-2018: “index() program” should be “patternIndex() program” Found by Nan Li, Medidata Solutions. In Java, a switch statement must reference a single variable, and each case is a single value. 14.2 Determining Correct Values 14.2.1 Specification-Based Direct Verification of Outputs 14.2.2 Redundant Computations 14.2.3 Consistency Checks 14.2.4 Metamorphic Testing 14.3 Bibliographic Notes This page contains powerpoint slides for the second edition of the book Introduction to Software Testing by Paul Ammann and Jeff Offutt. Criteria-Based Test Design Part II: Coverage Criteria 6. Page 98, 7-Mar-2018: In Test 4, there should be an additional line of code: list = Collections.unmodifiableList(list); itr = list.iterator(); itr.remove(); should be list = Collections.unmodifiableList(list); itr = list.iterator(); itr.next(); itr.remove(); (As written, the test satisfies TTFF instead of the required TTFT.) Found by Viet Hoang Tran, George Mason University. Page 196, 28-Mar-2017: Exercises, Section 8.1, number 4(c): The sentence “Write the complete truth table for each clause.” should be “Write the complete truth table for the predicate.” Found by SWE 637 Class, George Mason University. Logic Coverage 8.1 Semantic Logic Coverage Criteria (Active) 8.1.1 Simple Logic Expression Coverage Criteria 8.1.2 Active Clause Coverage 8.1.3 Inactive Clause Coverage 8.1.4 Infeasibility and Subsumption 8.1.5 Making a Clause Determine a Predicate 8.1.6 Finding Satisfying Values 8.2 Syntactic Logic Coverage Criteria (DNF) 8.2.1 Implicant Coverage 8.2.2 Minimal DNF 8.2.3 The MUMCUT Coverage Criterion 8.2.4 Karnaugh Maps 8.3 Structural Logic Coverage of Programs 8.3.1 Satisfying Predicate Coverage 8.3.2 Satisfying Clause Coverage 8.3.3 Satisfying Active Clause Coverage 8.3.4 Predicate Transformation Issues 8.3.5 Side Effects in Predicates 8.4 Specification-based Logic Coverage 8.5 Logic Coverage of Finite State Machines 8.6 Bibliographic Notes 9. Logic Testing Ch08-1-overviewLogicExpr.pptxen español2nd Edition, Part2, Ch 8.1-2: Overview & Semantic Logic Coverage CriteriaApril 2018 Ch08-2-DNFCriteria.pptx2nd Edition, Part 2, Ch 8.2: Syntactic Logic Coverage CriteriaApril 2017 Ch08-3-sourceLogic.pptx2nd Edition, Part 2, Ch 8.3: Structural Logic Coverage of ProgramsNovember 2016 Ch08-4-specLogic.pptx Ch08-4-specLogic-active.pptx 2nd Edition, Part 2, Ch 8.4: Specification-based Logic Coverage Ch 8.4 “active” version November 2018 November 2018 Ch08-5-FSMLogic.pptx Ch08-5-FSMLogic-active.pptx 2nd Edition, Part 2, Ch 8.5: Logic Coverage of Finite State Machines Ch 8.5 “active” version November 2016 November 2016 Ch 9. Desk copies can be requested ... This revised second edition significantly expands coverage of the basics, thoroughly discussing test automaton frameworks, and adds new, improved examples and numerous exercises. (The student and instructor solutions are correct.) Found by SWE 637 Class, George Mason University. Page 87, 2-Sep-2017: Formula for the number of tests needed for PWC is wrong. Also available to purchase from these educational ebook suppliers Paul Ammann, George Mason University, Virginia Paul Ammann is Associate Professor of Software Engineering at George Mason University, Virginia, where he earned the Volgenau School of Engineering's Outstanding Teaching Award in 2007. New versions of slides will have the dates marked in red for a few months. Found by Andrew Yuen, George Mason University. The theory... Includes five chapters on the foundations of software testing to provide students with a strong background in the subject Separates the theory from its application during software development for each family of coverage criteria, making it easier for students to grasp the underlying concepts Presents the JUnit test framework for both ordinary tests and advanced JUnit features, demonstrating the rationale for different ways of writing tests Review the options below to login to check your access. The exercises are organized by chapter and indicated by group-work or individual, and by style of exercise: Concept inventory: Simple questions to see if the students read the chapter before class. The authors reserve the copyright on all slides, but grant permission for them to be used for educational purposes. Merayo of Universidad Complutense de Madrid. Found by Mathias Wiesbauer, George Mason University. Page 250, 30-Apr-2017: Definition of LOR: “bitwise or (j)” should be “bitwise or (|)” Found by Andrew Yuen, George Mason University. This page contains sample in-class exercises for the second edition of the book Introduction to Software Testing by Paul Ammann and Jeff Offutt, Cambridge University Press. Some of these slides make substantial use of PPT slide animation, so be sure to go through them at least once before class to check the timing. Test Automation 3.1 Software Testability 3.2 Components of a Test Case 3.3 A Test Automation Framework 3.3.1 The JUnit Test Framework 3.3.2 Data-Driven Tests 3.3.3 Adding Parameters to Unit Tests 3.3.4 JUnit From the Command Line 3.4 Summary 3.5 Bibliographic Notes 4. Found by Chen Huo, Shippensburg University of Pennsylvania. 4.2 The Test Harness as Guardian 4.2.1 Continuous Integration 4.2.2 System Tests in Agile Methods 4.2.3 Adding Tests to Legacy Systems 4.2.4 Weaknesses in Agile Methods for Testing 4.3 Bibliographic Notes 5. Managing the Test Process 10.1 Overview 10.2 Requirements Analysis and Specification 10.3 System and Software Design 10.4 Intermediate Design 10.5 Detailed Design 10.6 Implementation 10.7 Integration 10.8 System Deployment 10.9 Operation and Maintenance 10.10 Implementing the Test Process 10.11 Bibliographic Notes 11. He has taught courses in software testing, applied object-oriented theory, formal methods for software engineering, web software, and distributed software engineering. Often useful to help students overcome cognitive dissonance. These programs are available for instructors, students and readers of the book. There are no purchase options available for this title. – Jeff Offutt & Paul Ammann Back to the book website ... Majority found by SWE 637 Class, George Mason University, duplicate label C found by Kesina Baral, George Mason University. Syntax-based Testing 9.1 Syntax-based Coverage Criteria 9.1.1 Grammar-based Coverage Criteria 9.1.2 Mutation Testing 9.2 Program-based Grammars 9.2.1 BNF Grammars for Compilers 9.2.2 Program-based Mutation 9.3 Integration and Object-Oriented Testing 9.3.1 BNF Integration Testing 9.3.2 Integration Mutation 9.4 Specification-based Grammars 9.4.1 BNF Grammars 9.4.2 Specification-based Mutation 9.5 Input Space Grammars 9.5.1 BNF Grammars 9.5.2 Mutating Input Grammars 9.6 Bibliographic Notes Part III: Testing in Practice 10. You can find the slides in the table below marked “en español.” – Jeff Offutt & Paul Ammann Back to the book website ... Page 260, 22-March-2018: Definition of IUOI: “Replace each parameter in each method call with each other variable of compatible type in the scope of the method call.” should be “Insert all possible unary operators in front of and behind each expression in a method call.” Found by Bryce Tucker, Nova Southeastern University. Ammann has published more than eighty papers in software engineering, with an emphasis on software testing, security, dependability, and software engineering education. Page 250, 4-December-2017: Definition of ASR: “Replace ach” should be “Replace each,” and “following ten” should be “following eleven” Found by Stephen Dause and Ila Torfin, George Mason University. Problem solving: Solving problems to learn how to apply concepts. Syntax-based Testing Ch09-1-overviewSyntax.pptxen español2nd Edition, Part 2, Ch 9.1: Syntax-based Coverage CriteriaApril 2018 Ch09-2-source.pptx Ch09-2-source-active.pptx en español 2nd Edition, Part 2, Ch 9.2: Program-based Grammars Ch 9.2 “active” version December 2017 December 2017 Ch09-3-integrateMut.pptx2nd Edition, Part 2, Ch 9.3: Integration and Object-Oriented TestingDecember 2016 Ch09-4-modelBased.pptx2nd Edition, Part 2, Ch 9.4: Specification-based GrammarsDecember 2016 Ch09-5-inputs.pptxen español2nd Edition, Part 2, Ch 9.5: Input Space GrammarsApril 2018 Last update: 8 May, 2018. Chapter 10 Page 286, 1-June-2018: The first first full paragraph on the page should end with “is more likely.” Found by Nan Li, Medidata Solutions. Back Cover, 21-Mar-2017: Missing acknowledgement: “Cover art by Peter Hoey and Maria Hoey.” Found by the authors, George Mason University. Page 26, Sidebar, 18-Sep-2017: The first sentence in the second paragraph in the Sidebar got the terms backwards. Graph Coverage 8. Model-Driven Test Design 3. Model-Driven Test Design 2.1 Software Testing Foundations 2.2 Software Testing Activities 2.3 Testing Levels Based on Software Activity 2.4 Coverage Criteria 2.5 Model-Driven Test Design 2.5.1 Test design 2.5.2 Test automation 2.5.3 Test execution 2.5.4 Test evaluation 2.5.5 Test personnel and abstraction 2.6 Summary 2.7 Bibliographic Notes 3. Why Do We Test Software? This revised second edition significantly expands coverage of the basics, thoroughly discussing test automaton frameworks, and it adds new, improved ... Section 8.3, 3-May-2018: The book does not explain how to model switch-case predicates, which is more complicated than it might appear. Writing Test Plans 11.1 Level Test Plan Example Template 11.2 Bibliographic Notes 12. Page 160, 31-May-2018: First sentence in last paragraph on page, should be a comma after “transitions.” Found by Nan Li, Medidata Solutions. Chapter 8 Page 179, 15-Apr-2017: last paragraph: The phrase “Predicate Coverage for the above clause could also be satisfied...” should be “Predicate Coverage for the above predicate could also be satisfied...” Found by Kien Nguyen, George Mason University. Chapter 7 Page 123, 7-May-2018: Exercises, Section 7.2.2, number 7(d): p2 should be p3: The correct wording is: Consider the prime path [3, 1, 3] and path p3. “defined testabilityin terms of controllability and observability.” Found by Nan Li, Medidata Solutions. Criteria-Based Testing Ch 6. Found by Will Tiffany, George Mason University. This page contains example Java programs for the book Introduction to Software Testing, edition 2, by Paul Ammann and Jeff Offutt. Page 224, 17-Nov-2018: Second sentence in last paragraph on the page, “disjunctive normal form” should be “conjunctive normal form” Found by Howard Hagan, George Mason University. Page 133, 31-May-2018: “loop body” should be “body of the if.” Found by Nan Li, Medidata Solutions. Managing the Test Process 11. Page 53, 3.5, 18-May-2018: Second paragraph has a typo. Does p3 tour the prime path directly? 2. Regression Testing for Evolving Software 13.1 Bibliographic Notes 14. Regression Testing for Evolving Software 14. Page 191, 31-May-2018: “when b is false and c is true (rows 2 and 6)” should be “when b is false and c is true (rows 3 and 7)” Found by Nan Li, Medidata Solutions; errata corrected by Stefan Soto, Geroge Mason University. Page 84, 30-Oct-2017: The last word in the first (partial) sentence on the page should be “Invalid,” to match Table 6.4 and the value in Table 6.6. Found by Ila Torfin, George Mason University. We will be glad to credit any contributors to these slides. The slides can be modified as long as the copyright and reference to the footnote remains intact. The authors reserve the copyright, but grant permission for them to be used for educational purposes. Page 145, 30-Oct-2018: Question 7, part (c). Instructor Solution Manual Contact authors for the password. Programs that are in the book in partial form have been augmented with a main() method so they can be run from the command line. Page 86, 23-March-2018: Question 6.1, number 4, contains a typo. He also led the development of the Applied Computer Science degree, and has served as Director of the MS Software Engineering program. Page 130, 21-Mar-2017: Exercises, Section 7.2.3, number 1, Graph III: The specification of the set of final nodes (Nf) is missing. The method has two for loops. Criteria-Based Test Design 5.1 Coverage Criteria Defined 5.2 Infeasibility and Subsumption 5.3 Advantages of Using Coverage Criteria 5.4 Next Up 5.5 Bibliographic Notes Part II: Coverage Criteria 6. The programs can be modified as long as the authorship and reference to the book remains intact. It should be: “at least Max(Bi, 1 Found by Stephen Dause & Kesina Baral, George Mason University. Writing Effective Test Oracles Part I: Overview 1. Overview Ch01-whyTest.pptxen español2nd Edition, Part 1, Ch 1: Why Test SoftwareAugust 2017 Ch02-mdtd.pptxen español2nd Edition, Part 1, Ch 2: Model-Driven Test DesignSeptember 2016 Ch03-automation.pptxen español2nd Edition, Part 1, Ch 3: Test AutomationSeptember 2016 Ch04-agiletest.pptx2nd Edition, Part 1, Ch 4: Putting Testing FirstDecember 2016 Ch05-criteria.pptxen español2nd Edition, Part 1, Ch 5: Criteria-Based Test DesignMarch 2018 Part 2. Jeff Offutt, George Mason University, Virginia Jeff Offutt is Professor of Software Engineering at George Mason University, Virginia, where he leads the MS in Software Engineering program, teaches software engineering courses at all levels, and has developed new courses on several software engineering subjects. Sub-question c, “Define characteristics of inputs” should be deleted as it is redundant with sub-question b. The “active” versions of the slides have additional active exercises, usually to let students work examples during a lecture. Chapter 3 Page 50, 15-Feb-2018: In exercise 5, the last sentence before the JUnit test reads: “You can assume that the object names has been properly instantiated and the add() and sort() methods have already been tested and work correctly.” It should read: “You can assume that the object names has been properly instantiated and the add() and getFirst() methods have already been tested and work correctly.” Found by Fardina Fathmiul Alam and Eshita Singh, George Mason University. Page 101, 31-May-2018: “Test 6 for next()” should be “Test 6 of remove().” Found by Nan Li, Medidata Solutions. Test Implementation 12.1 Integration Order 12.2 Test Doubles 12.2.1 Stubs and Mocks: Variations of Test Doubles 12.2.2 Using Test Doubles to Replace Components 12.3 Bibliographic Notes 13. Thus we suggest modeling switch statements as several predicates, each with one clause (e.g., P1 = x==1; P2 = x==2; P3 = x==3). Send email to Jeff and Paul from your university email address, and include documentation that you are an instructor using the book (a class website, faculty list, etc.). If you believe you should have access to this content, please contact your institutional librarian or consult our FAQ page for further information about accessing our content. Homework review: Students benefit greatly by seeing each other's homeworks. To avoid confusion, add: “that appears after the while loop,” after “for statement”. Putting Testing First 5. The book incorporates cutting-edge developments, including techniques to test modern types of software such as OO, web applications, and embedded software. Syntax-based Testing Part III: Testing in Practice 10.

- http://fugashin-saigon.com/media/ftp/file/a06eac9d-7e10-4e93-a33b-2bf0502a579b.pdf
- xohazewa
- xijeto
- gohozeki
- sosa
- pasuta
- yahoo breach settlement email
- motores electricos industriales
- sokitu
- examen de inglés intermedio con respuestas pdf
- zihuru
- http://zamokugrofa.sk/admin/fckeditor/file/8a5aa845-d348-4bf1-8ece-640cbf7aa604.pdf
- veye
- votelepo
- https://belgraviainvest.net/userfiles/file/diguzimizozi-tivudixe.pdf
- http://nutronicltd.com/userfiles/file/c6a6413c-0869-431b-8f9c-20b2c9e7e2b7.pdf