

1. Record Nr.	UNINA9911019378303321
Autore	Homès Bernard
Titolo	Fundamentals of Software Testing
Pubbl/distr/stampa	Newark : , : John Wiley & Sons, Incorporated, , 2024 ©2024
ISBN	9781394298976 1394298978 9781394298952 1394298951
Edizione	[2nd ed.]
Descrizione fisica	1 online resource (388 pages)
Collana	Computer engineering series
Disciplina	005.1
Soggetti	Computer software - Testing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Cover -- Title Page -- Copyright Page -- Contents -- Preface -- Glossary -- Chapter 1. Fundamentals of Testing -- 1.1. What is testing? -- 1.1.1. Software and systems context -- 1.1.2. Causes of software defects -- 1.1.3. Role of testing in software development, maintenance and operations -- 1.1.4. Tests and quality -- 1.1.5. Terminology. -- 1.2. What is testing? -- 1.2.1. Origin of defects -- 1.2.2. Common goals of testing -- 1.2.3. Examples of objectives for testing -- 1.2.4. Test and debugging -- 1.3. Paradoxes and main principles -- 1.3.1. Testing identifies the presence of defects -- 1.3.2. Exhaustive testing is impossible -- 1.3.3. Early testing -- 1.3.4. Defect clustering -- 1.3.5. Pesticide paradox -- 1.3.6. Testing is context dependent -- 1.3.7. Absence-of-errors fallacy -- 1.4. Test activities, testware and test roles -- 1.4.1. Planning -- 1.4.2. Monitoring and control -- 1.4.3. Test analysis and design -- 1.4.4. Test implementation -- 1.4.5. Test execution -- 1.4.6. Reporting -- 1.4.7. Test completion activities -- 1.4.8. The value of traceability -- 1.4.9. Impact of context on the test process -- 1.5. Roles in testing -- 1.6. Essential skills and "good practices" in testing -- 1.6.1. Generic skills -- 1.6.2. Specific skills -- 1.6.3. Whole team approach -- 1.6.4. Independence of testing -- 1.6.5. Levels of independence -- 1.6.6. Adapt to objectives -- 1.6.7.

Destructive or constructive? -- 1.6.8. People skills -- 1.6.9. Change of perspective -- 1.7. Testers and code of ethics (FL 1.6) -- 1.7.1. Public -- 1.7.2. Client and employer -- 1.7.3. Product -- 1.7.4. Judgment -- 1.7.5. Management -- 1.7.6. Profession -- 1.7.7. Colleagues -- 1.7.8. Self -- 1.8. Sample exam questions -- Chapter 2. Testing Throughout the Software Life Cycle -- 2.1. Testing through the software development life cycle -- 2.1.1. Sequential models. 2.1.2. Iterative models -- 2.1.3. Incremental model -- 2.1.4. RAD -- 2.1.5. Agile models -- 2.1.6. Selection of a development model -- 2.1.7. Positioning tests -- 2.1.8. Test-first and shift-left approaches -- 2.2. Test levels and test types -- 2.2.1. Component-level testing or component tests -- 2.2.2. Integration-level testing or integration tests -- 2.2.3. System tests -- 2.2.4. Acceptance tests -- 2.2.5. Other levels -- 2.3. Types of tests -- 2.3.1. Functional tests -- 2.3.2. Nonfunctional tests -- 2.3.4. Tests associated with changes -- 2.3.5. Comparisons and examples -- 2.4. Test and maintenance -- 2.4.1. Maintenance context -- 2.4.2. Evolutive maintenance -- 2.4.3. Corrective maintenance -- 2.4.4. Retirement and replacement -- 2.4.5. Regression test policies -- 2.4.6. SLA validation and acceptance -- 2.5. Oracles -- 2.5.1. Problems with oracles -- 2.5.2. Sources of oracles -- 2.5.3. Oracle usage -- 2.6. Process improvements -- 2.6.1. Objectives -- 2.6.2. Measurements -- 2.6.3. Retrospectives and improvements -- 2.7. Specific cases -- 2.7.1. Performance tests -- 2.7.2. Maintainability tests -- 2.8. Sample exam questions -- Chapter 3. Static Testing -- 3.1. Static techniques and the test process -- 3.2. Review process -- 3.2.1. Types of reviews -- 3.2.2. Roles and responsibilities during reviews -- 3.2.3. Phases of reviews -- 3.2.4. Success factors for reviews -- 3.2.5. Comparison of the types of reviews -- 3.3. Static analysis by tools -- 3.3.1. Types of static analysis -- 3.3.2. Types of defects that can be identified -- 3.3.3. Data flow analysis -- 3.4. Added value of static activities -- 3.5. Sample exam questions -- Chapter 4. Test Design Techniques -- 4.1. The test development process -- 4.1.1. Terminology -- 4.1.2. Traceability -- 4.2. Categories of test design techniques -- 4.2.1. Black box, white box or gray box. 4.2.2. Experience-based techniques -- 4.2.3. Test characteristics -- 4.2.4. Limitations and assumptions -- 4.3. Black-box techniques -- 4.3.1. Equivalence partitioning -- 4.3.2. Boundary value analysis -- 4.3.3. Decision tables -- 4.3.4. Other combinational techniques -- 4.3.5. State transition testing -- 4.3.6. Use case testing -- 4.3.7. Limitations and assumptions -- 4.4. Structure-based techniques -- 4.4.1. Statement testing and coverage -- 4.4.2. Decision testing and coverage -- 4.4.3. Other structure-based techniques -- 4.4.4. MC/DC coverage -- 4.4.5. Limitations and assumptions of structure-based testing -- 4.4.6. Coverage level and exit criteria -- 4.5. Experience-based technique -- 4.5.1. Attacks -- 4.5.2. Defect taxonomies -- 4.5.3. Error guessing and ad hoc testing -- 4.5.4. Exploratory testing -- 4.5.5. Limitations and assumptions -- 4.6. Collaboration-based test approaches -- 4.6.1. Collaborative user stories -- 4.6.2. Acceptance criteria -- 4.6.3. Acceptance test-driven development -- 4.7. Choosing test techniques -- 4.8. Sample exam questions -- Chapter 5. Test Management -- 5.1. Test organization -- 5.1.1. Independence levels -- 5.1.2. Roles and responsibilities -- 5.1.3. Human and contractual aspects -- 5.2. Test planning and estimation -- 5.2.1. Planning and evaluation activities -- 5.2.2. Test planning activities -- 5.2.3. Test documentation -- 5.2.4. Entry and exit criteria for test activities -- 5.3. Test progress monitoring and control (FL 5.3) -- 5.4. Reporting -- 5.4.1. What to report, to whom and how? -- 5.4.2. Statistics and graphs -- 5.5. Transverse processes and activities -- 5.5.1. Test data

definition -- 5.5.2. Configuration management (FL 5.4) -- 5.5.3. Change management -- 5.6. Risk management (FL 5.2) -- 5.6.1. Principles of risk management -- 5.6.2. Project risks and product risks. 5.6.3. Introduction to risk management -- 5.7. Defect management (FL 5.5) -- 5.7.1. Introduction to defect management -- 5.7.2. Defect identification -- 5.7.3. Actions applied to defects -- 5.7.4. Defect disposition -- 5.8. Sample exam questions -- Chapter 6. Tools Support for Testing -- 6.1. Types of test tools -- 6.1.1. Test tool classification -- 6.1.2. Tools supporting test management -- 6.1.3. Tools supporting requirement management -- 6.1.4. Tools supporting static tests -- 6.1.5. Modeling tools -- 6.1.6. Tools supporting test design and test data creation -- 6.1.7. Tools supporting test execution -- 6.1.8. Tools supporting test environment management -- 6.1.9. Tools supporting test data comparison -- 6.1.10. Tools supporting test coverage measurement -- 6.1.11. Other test supporting tools -- 6.2. Assumptions and limitations of test tools -- 6.2.1. Advantages and risks of the tools -- 6.2.2. Specific considerations for some tools -- 6.3. Selecting and introducing tools in an organization -- 6.3.1. Main principles -- 6.3.2. Tool selection process -- 6.3.3. Test tool implementation -- 6.3.4. To build or to buy test tools? -- 6.4. Sample exam questions -- Chapter 7. Mock Exam -- Chapter 8. Templates and Models -- 8.1. Master test plan -- 8.2. Test plan -- 8.2.1. Test plan as per IEEE 829-1998 -- 8.2.2. Test plan as per IEEE 829-2008 -- 8.3. Test design document -- 8.3.1. Test design specifications as per IEEE 829-1998 -- 8.3.2. Test design document as per IEEE 829-2008 -- 8.4. Test case -- 8.4.1. Test case document as per IEEE 829-1998 -- 8.4.2. Test case document as per IEEE 829-2008 -- 8.5. Test procedure -- 8.5.1. Test procedure document as per IEEE 829-1998 -- 8.5.2. Test procedure document as per IEEE 829-2008 -- 8.6. Test log -- 8.6.1. Test log as per IEEE 829-1998 -- 8.6.2. Test log as per IEEE 829-2008 -- 8.7. Defect report. 8.7.1. Defect report as per IEEE 829-1998 -- 8.7.2. Defect report as per IEEE 829-2008 -- 8.8. Test report -- 8.8.1. Test report as per IEEE 829-1998 -- 8.8.2. Interim test report as per IEEE 829-2008 -- 8.8.3. Level test report as per IEEE 829-2008 -- 8.8.4. Master test report as per IEEE 829-2008 -- Chapter 9. Answers to the Questions -- 9.1. Answers to the end-of-chapter questions -- 9.2. Correct answers to the sample paper questions -- References -- Index -- Other titles from ISTE in Computer Engineering -- EULA.

Sommario/riassunto

Software testing has greatly evolved since the first edition of this book in 2011. Testers are now required to work in "agile" teams and focus on automating test cases. It has thus been necessary to update this work, in order to provide fundamental knowledge that testers should have to be effective and efficient in today's world. This book describes the fundamental aspects of testing in the different lifecycles, and how to implement and benefit from reviews and static analysis. Multiple other techniques are approached, such as equivalence partitioning, boundary value analysis, use case testing, decision tables and state transitions. This second edition also covers test management, test progress monitoring and incident management, in order to ensure that the testing information is correctly provided to the stakeholders. This book provides detailed course-study material for the 2023 version of the ISTQB Foundation level syllabus, including sample questions to help prepare for exams.
