1. Record Nr. UNINA9910458600303321

Autore	Utting Mark
Titolo	Practical model-based testing [[electronic resource]] : a tools approach / / Mark Utting, Bruno Legeard
Pubbl/distr/stampa	San Francisco, CA, : Morgan Kaufmann Publishers, 2006
ISBN	1-280-72897-3 9786610728978 0-08-046648-6
Descrizione fisica	1 online resource (455 p.)
Altri autori (Persone)	LegeardBruno
Disciplina	005.3028/7
Soggetti	Computer software - Testing Computer software - Testing - Automation Electronic books.
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Description based upon print version of record.
Nota di bibliografia	Includes bibliographical references and index.
Nota di contenuto	Front cover; Title page; Copyright page; Table of Contents; Preface; Acknowledgements; About the authors; Chapter 1 The challenge; 1.1 What Do We Mean by Testing?; 1.2 What Is Model-Based Testing?; 1.3 A Smart Card Example; 1.4 Summary; 1.5 Further Reading; Chapter 2 The pain and the gain; 2.1 Classic Testing Processes; 2.2 The Model-Based Testing Process; 2.3 Models: Build or Borrow?; 2.4 Your Maturity Level; 2.5 Hypothetical Case: Total Testing Hours; 2.6 Model-Based Testing Experience Reports; 2.7 Benefits of Model-Based Testing; 2.8 Limitations of Model-Based Testing; 2.9 Summary 2.10 Further ReadingChapter 3 A model of your system; 3.1 How to Model Your System; 3.2 A Case Study; 3.3 Transition-Based Models; 3.4 Pre/Post Models in B; 3.5 Summary; 3.6 Further Reading; Chapter 4 Selecting your tests; 4.1 Structural Model Coverage; 4.2 Data Coverage Criteria; 4.3 Fault-Based Criteria; 4.4 Requirements-Based Criteria; 4.5 Explicit Test Case Specifications; 4.6 Statistical Test Generation Methods; 4.7 Combining Test Selection Criteria; 4.8 Summary; 4.9 Further Reading; Chapter 5 Testing from finite state machines; 5.1 Testing Qui-Donc with a Simple FSM 5.2 EFSMs and the ModelJUnit Library5.3 Unit Testing ZLive with EFSMs;

5.4 Labeled Transition Systems Models; 5.5 Summary; 5.6 Further Reading; Chapter 6 Testing from pre/post models; 6.1 How to Write Pre/Post Models for Testing; 6.2 The System Process Scheduler Example; 6.3 The Triangle Example; 6.4 Robustness Testing from a Pre/Post Model; 6.5 Testing a Chat System with Spec Explorer; 6.6 Summary; 6.7 Further Reading; Chapter 7 Testing from UML transition-based models; 7.1 UML Modeling Notations; 7.2 Testing an eTheater with LTG/UML; 7.3 Testing a Protocol with Qtronic; 7.4 Summary
7.5 Further ReadingChapter 8 Making tests executable; 8.1 Principles of Test Adaptation; 8.2 Example: The eTheater System; 8.3 Summary; 8.4 Further Reading; Chapter 9 The gsm 11.11 case study; 9.1 Overview of the GSM 11.11 Standard; 9.2 Modeling GSM 11.11 in B; 9.3 Validation and Verification of the B Model; 9.4 Generating Tests with LTG/B; 9.5 Generating Executable Scripts; 9.6 Test Execution; 9.7 Summary; 9.8 Further Reading; Chapter 10 The atm case study; 10.1 Overview of the ATM System; 10.2 Modeling the ATM System in UML; 10.3 Generating Test Cases
10.4 Generating Executable Test Scripts10.5 Executing the Tests; 10.6 Summary; 10.7 Further Reading; Chapter 11 Putting it into practice; 11.1 Prerequisites for Model-Based Testing; 11.2 Selecting a Model-Based Testing Approach; 11.3 People, Roles, and Training; 11.4 Model-Based Testing and Agile Methods; 11.5 Model-Based Testing and the Unified Process; 11.6 Epilogue; Appendix A Summary of B abstract machine notation; Appendix B Summary of common OCL constructs; Appendix C Commercial tools; Glossary; Bibliography; Index

Sommario/riassunto

This book gives a practical introduction to model-based testing, showing how to write models for testing purposes and how to use model-based testing tools to generate test suites. It is aimed at testers and software developers who wish to use model-based testing, rather than at tool-developers or academics.The book focuses on the mainstream practice of functional black-box testing and covers different styles of models, especially transition-based models (UML state machines) and pre/post models (UML/OCL specifications and B notation). The steps of applying model-based testing are demons