1. Record Nr. UNINA9910300462203321

Autore	Crookshanks Edward
Titolo	Practical Software Development Techniques : Tools and Techniques for Building Enterprise Software / / by Edward Crookshanks
Pubbl/distr/stampa	Berkeley, CA : , : Apress : , : Imprint : Apress, , 2014
ISBN	9781484207284 1484207289
Edizione	[1st ed. 2014.]
Descrizione fisica	1 online resource (201 p.)
Collana	Expert's Voice in Software Development
Disciplina	005.1
Soggetti	Application software Software engineering Computer and Information Systems Applications Software Engineering
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	Contents at a Glance; Contents; About the Author; Acknowledgments; Introduction; Chapter 1: Version Control; Theory; Software Demonstration; Resolving Conflicts; Tagging and Branching; Retrieving a Previous Version; What to keep in the repository; IDE Integration; Distributed Version Control; Version Control Summary; Chapter 2: Unit Testing and Test Driven Development; Theory; Unit Testing Frameworks; JUnit; NUnit; Test Driven Development (TDD); Unit Testing Summary; Chapter 3: Refactoring; Theory; Software Demonstration Setup; A note About Tools; Refactoring the Code; Summary Chapter 4: Build Tools and Continuous IntegrationMake; Ant; NAnt/MSBuild; Maven; Continuous Integration (CI) Tools; Simple Example; Deploying to Environments; Summary; Chapter 5: Debugging; Breakpoints; Stepping; Stack Trace; Logging; Summary; Chapter 6: Development Methodologies and SDLC; Waterfall; Agile; Extreme Programming; Distributed Teams; Distributed Version Control; Summary; Chapter 7: Design Patterns and Architecture; Pattern Examples; The Observer Pattern (Behavioral); The Facade Pattern (Structural); The Singleton Pattern (Creational) Enterprise Patterns: MVC and Inversion of ControlModel-View-

Controller; Inversion of Control; Manual Example; Configured Example; Refactoring using Patterns; Factory Method Pattern; Strategy Pattern; Example; Architecture Pattern: N- Tier; Summary; Chapter 8: Software Requirements; Business Requirements; Functional Design; Technical Design; Change Control; Summary; Chapter 9: Just Enough SQL; A Note about the Server and Client Tools; Minimal Database Design; SQL Statement Basics; Filtering and Sorting; More Advanced SQL; Programming Frameworks; Basic ADO.NET; Basic JDBC Object-Relational Mapping - Methods and ToolsSummary; Appendix A: Enterprise Considerations and Other Topics; Number and Location of Team Members and/or Users; System Integration; Separation of Duties and Environmental Limitations; Software Political Statements- Which Language/Platform is Better?; Software Libraries and Frameworks - Use Third Party or Write In-House?; Domain Knowledge; Continuing Education; Contractor or Full Time Employee?; Summary; Appendix B: Discussion Questions; Chapter 1: Version Control; Chapter 2: Unit Testing and Test Driven Development; Chapter 3: Refactoring Chapter 4: Build Tools and Continuous IntegrationChapter 5: Debugging; Chapter 6: Development Methodologies; Chapter 7: Design Patterns and Architecture; Chapter 8: Software Requirements; Chapter 9: Just Enough SQL; Appendix C: Database Details; Appendix D: Bibliography; Books; Web Sites; Other Suggested Reading; SQL Additional Reading/Resources; Index

Sommario/riassunto

This book provides an overview of tools and techniques used in enterprise software development, many of which are not taught in academic programs or learned on the job. This is an ideal resource containing lots of practical information and code examples that you need to master as a member of an enterprise development team. This book aggregates many of these "on the job" tools and techniques into a concise format and presents them as both discussion topics and with code examples. The reader will not only get an overview of these tools and techniques, but also several discussions concerning operational aspects of enterprise software development and how it differs from smaller development efforts. For example, in the chapter on Design Patterns and Architecture, the author describes the basics of design patterns but only highlights those that are more important in enterprise applications due to separation of duties, enterprise security, etc. The architecture discussion revolves has a similar emphasis – different teams may manage different aspects of the application’s components with little or no access to the developer. This aspect of restricted access is also mentioned in the section on logging. Theory of logging and discussions of what to log are briefly mentioned, the configuration of the logging tools is demonstrated along with a discussion of why it’s very important in an enterprise environment.