

|                         |  |
|-------------------------|--|
| 1. Record Nr.           | UNINA9910144432303321  |
| Autore                  | Holcombe W. M. L (William Michael Lloyd), <1944->  |
| Titolo                  | Running an agile software development project [[electronic resource] /] / Mike Holcombe  |
| Pubbl/distr/stampa      | Hoboken, N.J., : Wiley, c2008  |
| ISBN                    | 1-282-13692-5<br>9786612136924<br>0-470-38588-X<br>0-470-38587-1   |
| Descrizione fisica      | 1 online resource (328 p.)   |
| Disciplina              | 005.1/1<br>005.11  |
| Soggetti                | Computer software - Development<br>Agile software development<br>eXtreme programming   |
| 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 (p. 305-308) and index.  |
| Nota di contenuto       | Running an Agile Software Development Project; Contents; Preface; 1. What Is an Agile Methodology?; 1.1 Rapid Business Change: The Ultimate Driver; 1.2 What Must Agile Methodologies be Able to Do?; 1.3 Agility: What Is It and How Do We Achieve It?; 1.4 Evolving Software: Obstacles and Possibilities; 1.5 The Quality Agenda; 1.6 Do We Really Need All This Mountain of Documentation?; 1.7 The Human Factor; 1.8 Some Agile Methodologies; 1.8.1 Dynamic Systems Development Method; 1.8.2 Feature-Driven Design; 1.8.3 Crystal; 1.8.4 Agile Modeling; 1.8.5 SCRUM; 1.8.6 Summary Table; 1.9 Review; Exercise ConundrumReferences; 2. Extreme Programming Outlined; 2.1 Some Guiding Principles; 2.2 The Five Values; 2.2.1 Communication; 2.2.2 Feedback; 2.2.3 Simplicity; 2.2.4 Courage; 2.2.5 Respect; 2.3 The 12 Basic Practices of XP; 2.3.1 Test-First Programming; 2.3.2 Pair Programming; 2.3.3 On-Site Customer; 2.3.4 The Planning Game; 2.3.5 System Metaphor; 2.3.6 Small, Frequent Releases; 2.3.7 Always Use the Simplest Solution That Adds Business Value; 2.3.8 Continuous Integration; 2.3.9 Coding Standards; 2.3.10 Collective Code Ownership; |

2.3.11 Refactoring; 2.3.12 Forty-Hour Week; 2.4 Can XP Work?  
2.5 The Evidence for XP2.5.1 Evidence for Test First; 2.5.2 Evidence for  
Pair Programming; 2.5.3 Evidence for XP; 2.6 Preparing to XP; Exercise;  
Conundrum; References; 3. Foundations: People and Teams Working  
Together; 3.1 Software Engineering in Teams; 3.2 Personalities and  
Team Success; 3.3 Observations of Team Behavior in XP Projects; 3.4  
Setting Up a Team; 3.5 Developing Team Skills; 3.6 Training Together;  
3.7 Finding and Keeping a Client for a University-Based Project or a  
Small Business Start-Up; 3.8 The Organizational Framework; 3.9  
Planning  
3.9.1 PERT (Program Evaluation and Review Technique)3.9.2 Gantt  
Charts; 3.10 Dealing with Problems; 3.10.1 Basic Strategies; 3.10.2  
When Things Go Really Wrong; 3.11 Risk Analysis; 3.12 Review;  
Exercises; Conundrum; References; 4. Starting an XP Project; 4.1 Project  
Beginnings; 4.1.1 Researching the Business Background; 4.1.2  
Exploring the Outline System Description; 4.2 The First Meetings with  
the Client; 4.3 Business Analysis and Problem Discovery; 4.4 The Initial  
Stages of Building a Requirements Document; 4.5 Techniques for  
Requirements Elicitation; 4.6 Putting Your Knowledge Together  
4.7 Getting Technical4.8 Developing the Requirements Documents; 4.9  
Specifying and Measuring the Quality Attributes of the System; 4.9.1  
Identifying Attributes; 4.9.2 Specifying the Acceptable Level of an  
Attribute; 4.9.3 User Characteristics and User Interface Characteristics;  
4.10 The Formal Requirements Document and System Metaphor; 4.10.1  
Commentary; 4.11 Contract Negotiation; 4.12 Case Study: The Impact  
of Organizational Politics; 4.13 Review; Conundrum; References; 5.  
Identifying Stories and Preparing to Build; 5.1 Looking at the User  
Stories; 5.2 Collections of Stories  
5.2.1 Pharmacovigilance

---

## Sommario/riassunto

A Practical Approach To Building Small To Medium Software Systems For Real Business Clients Based on more than 100 actual commercial projects, this book clearly explains how to run an agile software development project that delivers high-quality, high-value solutions to business clients. It concentrates on the practical, social, business, and management aspects as well as the technical issues involved. Professor Holcombe successfully connects readers with the wave of "Agile 2.0" concepts that take the techniques of agile development and place them in the service of business goals. Since

---