Preface to the Third Edition

An introductory course in Software Engineering remains one of the hardest subjects to teach. Much of the difficulty stems from the fact that Software Engineering is a very wide field which includes a wide range of topics. Consequently, what should be the focus of an introductory course remains a challenge with many possible viewpoints.

This third edition of the book approaches the problem from the perspective of what skills a student should possess after the introductory course, particularly if it may be the only course on software engineering in the student's program. The goal of this third edition is to impart to the student knowledge and skills that are needed to successfully execute a project of a few person-months by employing proper practices and techniques. Incidently, a vast majority of the projects executed in the industry today are of this scope—executed by a small team over a few months. Another objective of the book is to lay the foundation for the student for advanced studies in Software Engineering.

Executing any software project requires skills in two key dimensions—engineering and project management. While engineering deals with issues of architecture, design, coding, testing, etc., project management deals with planning, monitoring, risk management, etc. Consequently, this book focuses on these two dimensions, and for key tasks in each, discusses concepts and techniques that can be applied effectively on projects.

The focus of the book remains as the first course in software engineering, and it retains its character of having a running case study with most of the outputs available. This edition draws upon my experience during my sabbaticals with two software companies—Infosys Technologies and Microsoft Corporation—and my practice-oriented book *Software Project Management in Practice* (Addison-Wesley, 2002) to bring, in addition to the concepts, more elements of how these concepts are actually applied in practice.

In this edition, new material has been added on current practices, outdated material has been removed, and discussion has been sharpened. The following key additions have been made:

- In “Software Process” a discussion on the timeboxing model for iterative development and on inspection process
- In “Requirements Analysis and Specification”a description of Use Cases
- A new chapter on “Software Architecture”
- In “Project Planning” some practical techniques for estimation, scheduling, tracking, risk management, etc.
- In “Object Oriented Design”, discussion on UML and on concepts like cohesion, coupling, and open-closed principle
- In “Coding” many additions have been made. These include refactoring, test driven development, and pair programming, as well as a discussion on common coding defects, coding standards, and some useful coding practices.
- In “Testing” a discussion on pair-wise testing as an approach for functional testing, defect tracking, and defect analysis and prevention

In addition to the old case study, a new case study has been added. Various work products of the case studies, including the SRS, architecture document, project plan, design document, code, and test plan, have been made available through the Web site.

A Web site has been created for this edition. In addition to outputs of the case studies, implementations of some of some of the examples are also available from the site. The site will soon include presentation slides for teaching, as well as other instructional material like examples and illustrative studies. The URL of the website is:

http://www.cse.iitk.ac.in/JaloteSEbook

I would like to express my gratitude to many people who helped me in preparing the case study. These include Kapil Narula, Ragesh Jaiswal, Vivek Pandey, Nilesh Lunawat, and Rajneesh Malviya. My special thanks to Vipindeep Vangala and Raghu Lingampally whose help in manuscript and Web site preparation allowed me to focus on the contents.

Pankaj Jalote