Preface

This is a book for students and practitioners of software engineering. Software engineers carry out knowledge-intensive tasks in software development, project management, or software quality. With this book, they will make better use of their personal knowledge and experience – and also of the knowledge in their groups or companies. In many organizations, there are initiatives to foster knowledge management or experiential learning. Some of them target software departments. Everyone involved in such an initiative will benefit from this book: managers, software engineers, and knowledge managers.

Knowledge engineers may find this book interesting as a view of knowledge management from the perspective of an application domain: software engineering.

At the intersection of software engineering and knowledge management. Software has become the most fascinating discipline in our society. Software controls cars, airplanes, and factories. Travel agencies and the military, banks and games depend on software. And software depends on knowledgeable experts. The ever-increasing demand for software calls for a broader and more explicit application of knowledge and experience. Software engineering is now a knowledge discipline, combining knowledge from computer science, engineering, and the application domains a particular software project is working for. Many individuals working in the software area increase their market value by improving use of knowledge and experience. However, companies should no longer rely on their employees' individual commitment to learning. Many software companies have identified knowledge as a catalyst for success. Knowledge management deals with the creation, management, and dissemination of knowledge in general.

Knowledge management has roots in philosophy, epistemology, and in several other disciplines – and it extends to formal languages and computer support. To make effective use of the techniques and tools, many software engineers believe they need a good overview of related concepts without having to dig into too many details. This book is an attempt to select key issues on several levels and provide an overview of the intersection of software engineering and knowledge management.

There are clear advantages in focusing on this audience. The familiarity of software engineers with computers and programming languages provides a better starting point for knowledge and experience management. Therefore, examples are almost exclusively taken from the software realm. Software engineering imposes a view on

knowledge management that differs from the perspective of business or sociology. This book wants to support software engineers better by adopting their perspective.

Taking experience seriously. This book puts an exceptional emphasis on experience. Experience is indispensable around software. However, this general assertion is rarely followed by any concrete consequence. When it comes to planning and optimization, experience is treated as "soft stuff" that resides inside individuals and is not accessible to a team or a company. This attitude gives away many opportunities for improvement. Of course, experience is a delicate material, and it takes dedicated techniques to handle it well. They are similar but not identical to the approaches used in knowledge management. Some techniques are explained in this book.

Neither knowledge nor experience can be collected, stored, and shipped like material goods. Knowledge and experience are effective only in the brain and consciousness of people. Knowledge and experience management can establish links and support learning on individual and organizational levels. Knowledge has been called a company asset. However, this is only part of the truth: software engineers and other knowledgeable *people* should be regarded the assets, as they will make knowledge effective. This insight leads to various consequences: Semiformal and formal descriptions can be well-suited although they are difficult to support by a computer. In certain situations, more formal notations and ontologies are applicable. It is the challenge and art of good knowledge and experience management to identify the best techniques in a given situation and to support smooth transitions between them.

Textbook and resource for self-directed learning. This book can be used as a classic textbook. It is organized in self-contained chapters. Nevertheless, it will be best to read all chapters in sequence. Readers less interested in the formalisms of Web-based languages and ontologies may want to skip the corresponding chapter. They will still be able to follow subsequent chapters on experience and application scenarios.

The book contains several problems as exercises for each chapter. They repeat some of the material presented, and they challenge the reader to reflect on some of the issues. There are solutions to all problems at the back of the book.

A reader studying the full material in a rigorous way, solving the problems and following the interactive examples, will gain the most in-depth insights.

Just for curiosity. Others might read the book with a different goal in mind. Knowledge managers can learn more about software engineering as a particular application domain by looking through a software engineer's eyes. Software experts with a genuine interest in improving their professional working style may pick the chapters and sections they are most interested in. They will be able to gain an overview of an emerging field. Assumptions and challenges are presented as well as opportunities and working lightweight techniques. A selection of application scenarios puts the techniques in context.

Benefits for Readers

This book contains problem sections and solutions. Various learning objectives are pursued. When readers follow the text and solve the problems, they will increase their capabilities and opportunities within a software organization.

Learning objectives of this book. Readers will gain a good overview of knowledge management aspects. As a result, they will be able to make judgments and well-informed decisions.

In particular:

- They will understand which kind of knowledge is important in software engineering, and why knowledge needs to be managed.
- They will know the terminology used in experience and knowledge management.
- They will understand the difference between knowledge and experience in the context of software engineering.
- They will be able to evaluate and discuss a given knowledge management initiative planned in a company.
- They will be able to suggest improvements on knowledge management initiatives.
- They will understand and distinguish management and developer perspectives on knowledge and experience management – and their potential conflicts.
- They will understand why information needs to be structured and reorganized ("engineered") for reuse in software projects.
- They will be aware of the fact that knowledge and experience is deeply related with people – it is not just dead material.

Readers will also be enabled to contribute to improvement initiatives actively:

- Readers will know different techniques and tools for structuring knowledge in software engineering.
- They will understand the meaning and potential of patterns for reuse.
- They will know how to start effective experience exchange in a team or business unit.
- Readers will know and understand recurring problems and misunderstandings that challenge the reuse of experiences and knowledge – and options to overcome them.
- They will know the visions of experience factory, organizational memory, and Web 2.0 approaches like Wikis or blogs, which have been drivers for the development of novel experience and knowledge management techniques.
- They can avoid common pitfalls associated with the introduction of new experience management tools.
- They will be able to recall practical examples of introducing and applying experience and knowledge management in companies for discussions and planning.

Concrete impact on software engineering work. According to Eraut and Hirsh [37], there are a number of concrete benefits associated with improved learning abilities.

Applying knowledge management and experiences to software engineering can have the following impacts:

- Carrying out software engineering activities faster.
- Improving the quality of the process.
- Improving communications around complex software engineering tasks.
- Becoming more independent and needing less supervision, which is important for new employees.
- Helping others learn to preform knowledge-intensive tasks, thus freeing resources for development activities.
- Combining tasks more effectively.
- Recognizing possible problems faster.
- Expanding the range of project situations in which one can perform competently.
- Increasing ability to handle difficult tasks and taking on tasks of greater complexity.
- Dealing with more difficult or more important cases, clients, customers, suppliers, or colleagues.

This list of value-adding abilities shows the potential of focusing on experience and knowledge management – as a learning approach for software engineers.

Hannover, Germany Kurt Schneider

http://www.springer.com/978-3-540-95879-6

Experience and Knowledge Management in Software Engineering

Schneider, K.

2009, XVI, 235 p., Hardcover
ISBN: 978-3-540-95879-6