

Foreword

The service industry has emerged as the prominent industry, and the software industry is at the heart of this transformation. The service-based view of the firm describes the new way of conceptualizing organizations and has led to an increasingly Web-based service industry. It is reasonable to envision a world in which services will become the main way of specifying, developing and providing functionality.

The software engineering community will play a central role in this paradigmatic shift and contributions like this book are essential to inform the professional and academic software engineering community about the current and future challenges, trends and solutions.

The term software industrialization indicates that there is no intention to start from scratch in the process of designing software engineering processes and practices. Rather, established manufacturing practices form a widely appreciated point of reference. This includes especially the desire for standardization in the form of patterns, software reuse, modularity and advanced software configuration capabilities.

The discipline of developing, deploying and managing software is well-researched and a plethora of methodologies, frameworks, reference models, tools and techniques supports nowadays the software engineer. However, despite this matured supportive environment, a range of challenges demands substantial shifts in the current practices of software engineering.

First, software engineering demonstrates like hardly any other sector the full utilization of a flat world. Globally distributed development processes have become the established way to build software in large projects. This leads to new challenges in terms of project management and the overall coordination of globally distributed contributions, standards and quality assurance to name just a few. Globalization does not only comprise developing in a flat world but also for a flat world meaning that internationalized software has to cater for the requirements of different regional languages, legislations and user expectations among others.

Second, agility has become a main obligation for software engineering. Agile development ensures early interactions with the user base and leads to a fluent transition from requirements engineering to software engineering. The iterative nature of agile development and the demand to provide early demonstrators demand reconsiderations of classical, often sequential software development processes. Among others, it requires ongoing interactions with various stakeholders along the software engineering process facilitated by intuitive but expressive powerful conceptual models.

Third, software engineering is now in many cases an open process. Open source models have led to entire new levels of transparency and accessibility in this process. Open innovation and a long tail of software engineers are examples for how the wisdom of the crowds can increase the development capacity. It also demonstrates, however, the limitations of current software lifecycle and governance models.

Fourth, the center of attention of software engineering keeps on changing. The classical concentration on data, functions and processes is increasingly complemented with a focus on services. While the technical benefits of service-oriented software engineering are well accepted, a similar level of appreciation is still missing in the specification of business requirements.

Fifth and finally, engineering off-the-shelf-solutions has become a mainstream activity. Enterprise Resource Planning is the most prominent form of this type of software. In this type of software the actual finalization of the engineering process is done in the form of configuration and customization. However, the high demand for this type of software stands in sharp contrast to the still modest development of dedicated frameworks and processes for such software.

Globalized, agile, open, service-oriented and configurable are just a few of the attributes of the current software engineering practice. A list of challenges that is by no means complete.

This well choreographed book approaches these interrelated challenges in three sections.

Section 1 proposes processes and frameworks for integrating requirements and software engineering. It is not a surprise that agile development plays a central role in the related chapters.

Section 2 is concerned with the productivity of software engineering. Model-based engineering, advanced software testing and componentization are a few of the selected issues that are discussed here.

Section 3 is dedicated to the development of Enterprise Systems in a global context. The special requirements of small and medium-sized organizations and open source development are two of the challenges that are covered in this part.

In light of the significant and fast emerging challenges that software engineers face today, the editors of this book have done an outstanding job in selecting the contents of this book. I am confident that this book will provide an appreciated contribution to the software engineering community. It has the potential to become one of the main reference points for the years to come.

Michael Rosemann

Brisbane, December 2008

Michael Rosemann is a Professor for Information Systems and Co-Leader of the Business Process Management Group at Queensland University of Technology, Brisbane, Australia. Michael has experiences as the Chief Investigator of a number of research projects funded by competitive European and Australian schemas. He has been a member of the prestigious Australian Research Council College of Experts in 2006/07. He is the author/editor of six books, more than 140 refereed papers and Editorial Board member of eight international journals incl. the Business Process Management Journal. His papers have been published in journals such as MIS Quarterly, IEEE Transactions on Knowledge and Data Engineering, Information Systems, European Journal of Information Systems and Decision Support Systems. He also presented his work at all global major IS conferences. His papers won the best paper award at CAiSE (1999), PACIS (2004) and ACIS (2005) and two of his PhD students won national PhD awards. Michael is the co-inventor of seven US patent proposals and co-editor of the book 'Process Management' that is also available in German, Russian and Chinese. Michael has been the General Chair of the 5th International Business Process Management Conference in 2007 and was the invited keynote at BPM 2008. Dr Rosemann chairs the Australian BPM Community of Practice (<http://bpm-collaboration.com>) and established QUT as a main vendor of BPM training (www.bpm-training.com). Michael has intensive consulting experiences and provided advice to organisations from various industries including telecommunications, banking, retail, insurance, utility, logistics, public sector and film industry. He has been a speaker at events of organisations such as SAP, Oracle and Infosys.