Preface

Foreword to the First Edition

Today, software systems are generally complex products and the use of engineering techniques is essential if the systems are to be produced successfully. Over the last three decades, this finding, which is frequently quoted but is now more than 30 years old, has led to intensive work on languages, methods, and tools in the IT field of Software Engineering to support the software creation process. However, despite great advances, we must concede that in comparison with other much older engineering disciplines, many questions still remain unanswered and new questions are constantly arising.

For example, a superficial comparison with the field of construction quickly shows that in there, international standards have been set for creating models of buildings, analyzing the models, and then realizing the models in actual constructions. The distribution of roles and tasks is generally accepted and there are professional groups such as architects, structural engineers, as well as engineers for construction above and below ground.

This type of model-based approach is increasingly finding favor in software development. In recent years in particular, this has led to international attempts to define a generally accepted modeling language so that just like in construction, a model created by a software architect can be analyzed by a "software structural engineer" before it is implemented in executable programs by specialists responsible for the realization, i.e., programmers.

This standardized modeling language is the Unified Modeling Language and it is subject to continuous further development by an international consortium in a gradual process. Due to the wide range of interested parties in the standardization process, the current version 2.0 of UML has emerged as a language family with a great many open questions with regard to scope, semantic foundation, and methodological use.

Over the past few years, Professor Rumpe has dedicated himself to this problem in his scientific and practical work, the results of which are now

available to a wide audience in two books. In these books, Professor Rumpe focuses on the methodological process. In line with the current finding that lightweight, agile development processes offer great advantages particularly in smaller and medium-sized development projects, Professor Rumpe has developed techniques for an agile development process. On this basis, he has then defined a suitable modeling language by defining a language profile for UML. In this language profile, UML/P, Professor Rumpe has made UML leaner and rounded it off in some places to produce a manageable version of UML in particular for an agile development process.

Professor Rumpe has explained this language UML/P in detail in his previous book "Modeling with UML", which offers a significant basis for the current book (the content of the previous book is briefly summarized). The current book, "Agile Modeling with UML", is dedicated primarily to the methodological treatment of UML/P.

Professor Rumpe addresses three core topics of model-based software development. These are:

- Code generation, i.e., the automated transition from a model to an executable program
- Systematic testing of programs using a model-based, structured definition of test cases
- Further development of models using techniques for transformation and refactoring

Professor Rumpe initially examines all three core topics systematically and introduces the underlying concepts and techniques. For each topic, he then presents his approach based on the language UML/P. This division and clear separation between basic principles and applications make the presentation extremely easy to understand and also allows the reader to transfer this knowledge directly to other model-based approaches and languages.

Overall, this book is of great benefit to those who practice software development, for academic training in the field of Software Engineering, and for research in the area of model-based software development. Practitioners learn how to use modern model-based techniques to improve the production of code and thus significantly increase quality. Students are given both important scientific basics as well as direct applications of the basic techniques presented. And last but not least, the book gives scientists a comprehensive overview of the current status of development in the three core topics it covers.

The book therefore represents an important milestone in the development of concepts and techniques for a model-based and engineering-style software development and thus offers the basis for further work in the future. Practical experience of using the concepts will validate their stability. Scientific, conceptual work will provide further research on the topic of model transformation based on graph transformation in particular. It will also deepen the area of model analysis in the direction of structural model analysis.

This deeper understanding of the IT methods in model-based software development is a crucial prerequisite for a successful combination with other engineering-style methods, such as in the field of embedded systems or the area of intelligent, user-friendly products. The fact that the language UML/P is not specific to any domain also offers a lot of opportunities here.

Gregor Engels

Paderborn, September 2004

Preface to the Second Edition

As this is the second book on agile software development with UML, interested readers will probably be familiar with the first book [Rum16]. The preface in [Rum16] holds true for both books and here, therefore, I refer to the first book, in which the following aspects are discussed:

- Agile methods and model-based methods are both successful software development techniques.
- So far, the two approaches have not been harmonized or integrated.
- However, the basic idea of using models instead of programming languages provides the opportunity to do exactly that.
- This book contributes to this integration in the form of UML/P.
- In the second edition, UML/P has been updated and adapted to UML 2.3 and Java Version 6.

I hope you enjoy using this book and its contents.

Bernhard Rumpe

Aachen, Germany, March 2012

Preface to the English and 3rd Edition

Colleagues have asked when the English version of the two books would be published. The first one was finished in 2016 and now, here comes the second one. I wish all the readers, students, teachers, and developers fun and inspiration for their work.

I would like to thank all the people that helped me translating and quality checking this book, namely Tracey Duffy for the main translation, Sylvia Gunder and Gabi Heuschen for continuous support, Robert Eikermann for the Latex and continuous integration setup, Kai Adam (for reviewing Chapters 5,7 and 10), Vincent Bertram (8), Arvid Butting (3,8,10), Anabel Derlam (1), Katrina Engelbrecht (3,9,10), Robert Eikermann (3,8,9), Timo Greifenberg (6,7,11), Lars Hermerschmidt (11,), Steffen Hillemacher (3,7,11), Katrin Hölldobler (9,10), Oliver Kautz (3,8,10), Thomas Kurpick (2), Evgeny Kusmenko (2,5), Achim Lindt (1,2,9), Matthias Markthaler (7,9), Klaus Müller (4), Pedram Mir Seyed Nazari (1,5), Dimitri Plotnikov (1,4,5), Deni Raco (6,7,8), Alexander Roth (4), Christoph Schulze (6,8,11), Michael von Wenckstern (2,3,4,6), and Andreas Wortmann (1,11).

Bernhard Rumpe

Aachen, Germany, February 2017

Further material:

http://mbse.se-rwth.de

http://www.springer.com/978-3-319-58861-2

Agile Modeling with UML
Code Generation, Testing, Refactoring
Rumpe, B.
2017, XIII, 388 p. 176 illus., 101 illus. in color.,
Hardcover
ISBN: 978-3-319-58861-2