Preface

As Component-Based Software Development (CBSD) starts to be effectively used, some software vendors have started to successfully sell and license commercial off-the-shelf (COTS) components. CBSD advocates the use of prefabricated pieces, perhaps developed at different times, by different people, and possibly with different uses in mind. The goal, once again, is the reduction of development times, costs, and efforts, while improving the flexibility, reliability, and maintainability of the final application due to the (re)use of software components already developed, tested and validated.

So far, most of the efforts from the Software Engineering community have concentrated on the functional aspects of CBSD, but leaving aside the (difficult) treatment of the quality issues and extra-functional properties of both components and component-based systems. However, this kind of property deserves special attention, since such properties may have even more importance than the other technical issues when building any commercial or industrial application.

The particular nature of COTS components (black-box binary entities developed by third parties, and independently deployable in unforeseen contexts) require that specific quality mechanisms are put in place for their effective integration in the development life-cycle of software applications. First, components need to be properly documented and offer measurement mechanisms in order to be assessed and evaluated for selection. Second, both the quality and some of the extra-functional properties of the final system heavily depend on the individual properties of its constituent components, and therefore some traceability between the quality attributes at these two levels is required. Finally, the use of third-party COTS components may also introduce risks, such as potentially harmful side-effects for the system, or quality degradation of the final product. COTS component testing and certification may offer partial solutions to these problems.

Our knowledge about several research and industrial initiatives trying to address some of the quality issues involved in CBSD motivated us to edit the present volume. In order to gather as many initiatives as possible, a "Call for Chapters" was issued in June 2002, requesting novel works and experiences related to software component quality, both at the individual component and composed system levels. As stated in the Call for Chapters, the main objective of the book was to provide an overview of the state-of-the-art in Component-Based Software Quality (CBSQ), discuss the main techniques and methods currently used in industry and academia, and analyze the most critical aspects related to CBSQ, such as component assessment, selection, quality evaluation and monitoring, testing, etc.

As a response to this public call, 21 chapter proposals were received from the most relevant research groups in this emerging discipline. After a thorough

peer-review process, 16 proposals were finally accepted, those included in this volume.

A first chapter written by the editors serves as an introduction to the topics covered by this book, and aims at describing the best practices in software components assessment. The goal of this introductory chapter is to highlight the specific nature of COTS and of COTS-components-based systems, introduce the basic concepts and common vocabulary used in the different approaches and practices, and identify the main problems that challenge CBSQ. This chapter also provides a basis for discussing the different approaches presented in this book.

The other chapters are organized into five parts, corresponding to the five main dimensions of quality assessment and CBSQ: COTS selection, testing and certification, quality models, formal approaches to quality assessment, and component-based systems quality management.

Four chapters comprise the first dimension, COTS selection. First, C. Alves discusses the new challenges that arise during the requirements engineering process for COTS-based systems. She presents an overview of a COTS selection process specially designed to help with the decision-making process. Then, H. Leung et al. present a domain-based COTS-product selection method, which uses domain models for capturing relevant features to be considered for analysis. The chapter by D. Kunda discusses the STACE framework, which recommends decomposition of high-level requirements into a hierarchy of social and technical criteria comprising functionality characteristics, technology factors, product quality characteristics, and social and economical factors. Finally, to address the issue of COTS selection tools, N. Maiden et al. describe the SCARLET Process Advisor, a Web-enabled workflow software tool that may guide software engineers to select software components and COTS packages.

The second dimension covers testing and certification. The chapter by A. Vincenzi et al. introduces the basic concepts of software testing, focusing on the state-of-the-art and state-of-the-practice in the context of CBSD. Then, J. Morris et al. discuss the implications of standardized testing on certification, presenting an XML-based test specification format and a system for executing these specifications. Finally, C. Atkinson, H. Gross, and F. Barbier outline the principles behind built-in contract testing that derive from built-in test (BIT), and describe how built-in testing can be naturally integrated into component-based development.

The third dimension focuses on software component quality models. First, R. Simão et al. try to identify and organize the most relevant quality characteristics of software components based on the ISO/IEC 9126 standard. The proposed quality model has been empirically validated through field research, and the results are analyzed by using a fuzzy model for software quality evaluation. Second, C. Atkinson, C. Bunse, and J. Würst present a model-driven approach for component-based development, introducing some strategies for deriving quality-related information from UML models, and then illustrating how the measurement of UML structural properties can help drive the quality assur-

ance activities of CBSD. To address particular quality model cases, the chapter by P. Botella et al. proposes the adoption of quality models as a means for structuring the description of the capabilities of ERP systems. The ISO/IEC 9126-1 quality standard is chosen as a framework, and a methodology for tailoring it to this specific domain is described. Finally, as another particular case, J. Bosch introduces a first attempt to classify software product-line maturity levels. Based on the conceptual framework presented in his chapter, he shows how an organization can identify the different levels of intra-organizational reuse that can be accomplished, and how to embark on improvement initiatives at each level.

The fourth part deals with formal approaches to CBSQ. In the first place, the chapter by H.Y. Kim et al. presents a framework for assessing component properties (such as completeness and consistency of requirement specifications) using Z and statecharts. The chapter also presents an approach for verifying properties such as reliability, using two different stochastic models. Then, R. Reussner et al. introduce a method based on RADL (Rich Architecture Definition Language) as another approach for determining the reliability of component-based software architectures. The method uses the notion of parameterized contracts to analyze the effects of component interactions on the final system's reliability. Contracts involve finite state machines that allow software architectures to define how a component's reliability will react to a given deployment environment. Finally, the chapter by R. Díaz et al. discusses some efficiency issues of iterative and incremental life-cycles, which involve balancing requirements. In order to improve the consistency-checking process, they propose reusing formal verification information – previously obtained by model-checking algorithms – to reduce the amount of verifications required.

The last dimension is concerned with CBSD management. First, L. Rose introduces risk management by presenting those risks that are specific to the development and maintenance of COTS-based systems. The chapter discusses some risk mitigation techniques that can be applied to ensure successful deployment and operation of this kind of system. And, then, the chapter by S. Sedigh-Ali et al. presents a framework that aims at reducing risks by using software metrics to accurately quantify factors contributing to the overall quality of a component-based system. The framework helps guide quality and risk management by identifying and eliminating the sources of risk. Tradeoffs between cost and quality in a component-based system are also discussed, as well as analytical techniques and formal models for taking into account both cost and quality during decision-making processes.

Every chapter presents a set of issues and problems commonly encountered when researching and conducting CBSD. In all of them, the authors share their vision about the importance of quality assessment, and how quality has a strong effect on system development and deployment. We hope that the insights and experiences described in this book can provide the reader with new research directions and valuable guidelines for conducting CBSD.

We want to express our gratitude to all individuals and parties who helped us produce this volume. In the first place, we want to thank Springer-Verlag for

believing in our project and for giving us the opportunity to publish this book. Special thanks to Ralf Gerstner, LNCS editor, for his kindliness, assistance, and continuous support. Many thanks to all those experts who submitted chapters, whether finally selected or not, and particularly to the contributing authors. Our gratitude also goes to the reviewers who helped in choosing and improving the selected chapters.

February 2003   Alejandra Cechich
                                                                                Mario Piattini
                                                                                Antonio Vallecillo