Sami Beydeda · Volker Gruhn (Eds.)

Testing Commercial-off-the-Shelf Components and Systems

Sami Beydeda · Volker Gruhn (Eds.)

Testing Commercial-off-the-Shelf Components and Systems

With 115 Figures and 38 Tables

Springer

Sami Beydeda
Federal Finance Office (Bundesamt für Finanzen)
Friedhofstraße 1
53225 Bonn, Germany
e-mail: sami.beydeda@bff.bund.de

Volker Gruhn
University of Leipzig
Applied Telematics / e-Business
Klostergasse 3
04109 Leipzig, Germany
e-mail: gruhn@ebus.informatik.uni-leipzig.de

Library of Congress Control Number: 2004115149

ACM Computing Classification (1998): D.2.4, D.2.5

ISBN 3-540-21871-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: By the authors
Cover design: KünkelLopka, Heidelberg
Production: LE-TEX Jelonek, Schmidt & Vöckler GbR, Leipzig
Printed on acid-free paper 45/3142/YL - 5 4 3 2 1 0

Preface

Component-based development has an intuitive underlying idea. Instead of developing a system by programming it entirely from scratch, develop it by using preexisting building blocks, *components*, and plug them together as required to the target system. The benefits are obvious. Parts of a system which do not have to be programmed do not entail costs and do not face the project manager with the risks typically encountered in software projects. It was expected that using components will also contribute to an increase in system quality. However, this expectation is fulfilled only holds in certain circumstances; quality assurance techniques, particularly testing, are still required.

The topic of this book is quality assurance and testing in component-based development in particular. In software construction using preexisting components, various subjects of testing can be distinguished; we also have several roles, each of which is concerned with quality assurance. Each subject of testing and each role has its own requirements of testing techniques. The chapters of this book present testing techniques which can be applied in the various situations.

In component-based development, testing can firstly be classified according to the subject under test. The subject under test can on the one hand be a component and on the other hand be a system consisting of components. We do not consider in this book the testing of component frameworks, such as application servers, or technologies for specific purposes, such as middleware. The testing of components can be further distinguished according to the assumptions made and information available. When testing a component, tests can be carried out without assuming an explicit context in which the component will be embedded. Such type of testing is usually conducted by the component provider, who does not necessarily know the specific environment in which the component will later operate. Tests can also be carried out within a specific system, particularly for integration testing purposes. Such type of testing is often employed by the component user prior to integration into a certain system. As discussed before, the significant difference of these

two types of testing is generally the availability of information, particularly source code.

This book consists of three parts, each reflects one of the above types of testing in component-based development.

Part I: Testing Components Context-Independently

This part of the book contains chapters covering the testing of components without assuming an explicit context in which the component will be embedded. This form of testing is usually conducted by the component provider.

Testing Polymorphic Behavior of Framework Components. The first chapter in this part deals with a specific type of component. Tyler and Soundarajan describe the testing of framework components with particular consideration to problems resulting due to polymorphism used for customization.

COTS Component Testing through Built-In Test. Barbier presents in this chapter an approach which employs the strategy of enhancing a component with the purpose of increasing testability. In particular, Barbier proposes to augment a component with metadata constructed on the basis of relevant aspects and present a technical class structure.

COTS Component Testing through Aspect-based Metadata. Cechich and Polo employ a similar approach. They explore aspect-based categorization of information for testing and augmenting a component with metadata for the generation of aspect-dependent test cases.

Automatic Testing of Exception Handling Code. In contrast to all approaches in this part, the approach of Fetzer, Högstedt, and Felber requires access to the source of the component under test. They describe a possibility of testing exception handling code vital, for instance, for recovery during runtime after a failure.

Part II: Testing Components in the Context of a System

This part of the book considers the testing of components, in contrast with the first part, in the context of the system in which the components will be integrated. This form of testing corresponds to that carried out by the component user and includes integration testing in particular.

A Process and Role-based Taxonomy of Techniques to Make Testable COTS Components. In this chapter, a survey of approaches and strategies for component user's testing is given by Memon.

Evaluating the integrability of COTS components – software product family viewpoint. Taulavuori, Niemelä, and Matinlassi address issues of component integrability with emphasis on integration aspects specific to software product families.

A User Oriented Framework for Component Deployment Testing. In this chapter Polini and Bertolino describe the generation of test cases from a system architecture specification for assessing suitability of components to a given environment.

Modeling and Implementation of Built-in Contract Tests. Gross, Schieferdecker, and Din also employ the strategy of enhancing a component for increasing testability. They explain how built-in testing can be used to enable components to test their environments.

Using a specification approach to facilitate component testing. Hybertson describes an approach to specification of components and system architectures for facilitating tests by the component user.

A Methodology of Component Integration Testing. Zhu and He consider integration testing formally; they propose a formal model for integration testing and a hierarchy of behavior observation schemes.

Part III: Testing Component-based Systems

This part shows the testing of entire component-based systems. Component-based systems possess certain properties, particularly at the architectural level, which other types of systems do not possess. This part shows how testing can be conducted when assuming such properties.

Modeling and Validation of Publish/Subscribe Architecture. Baresi, Ghezzi, and Zanolin focus on a specific type of architecture and show how a component-based system having such an architecture can be tested.

Performance Testing of Distributed Component Architectures. Denaro, Polini, and Emmerich show the use of testing techniques for performance analysis in early stages of development once the architectural decisions have been made.

A Generic Environment for COTS Testing and Quality Prediction. An approach for predicting quality of components, and thus a system consisting of these components, is presented by Cai, Lyu, and Wong.

Automatic Testing for Robustness Violations. Fetzer and Xiao explain an approach for detecting faults and security problems in third-party libraries without having access to their source codes with the application of fault injection techniques.

Testing Component-Based System Using FSMs. Beydeda and Gruhn present in this chapter an approach to black-box testing of component-based systems once the single constituents' specifications are given as finite state machines.

Acknowledgments

In July 2003, we invited the leading researchers in the area of testing components and component-based system to summarize their research results in the format of chapters. Fortunately, most of them accepted our invitation and contributed chapters to specific aspects of testing in component-based development. We are very grateful for this and would like to thank all authors for their effort.

Inviting the leading researchers guaranteed a high quality of the individual chapters and thus of the book. We, however, decided to conduct a review to further improve the chapters. The review was conducted with the support of a committee. The committee members reviewed the individual contributions and gave valuable remarks to revise and improve them. We want to thank the committee members for their support; without them the book would not have its quality.

The committee members were:

Franck Barbier
LIUPPA, Université de Pau
BP 1155
64013 Pau CEDEX, France

Luciano Baresi
Politecnico di Milano – Dipartimento di Elettronica e Informazione
Piazza L. da Vinci, 32 – I20133
Milano, Italy

Fevzi Belli
Universität Paderborn
Fakultät V (EIM-I)
Pohlweg 47-49 / Postfach 16 21
D-33095 Paderborn, Germany

Antonia Bertolino
Istituto di Scienza e Tecnologie dell'Informazione - "Alessandro Faedo"
Area di Ricerca del CNR di Pisa, Via Moruzzi 1
I-56124 Pisa, Italy

Jean-Michel Bruel
LIUPPA, Université de Pau
BP 1155
64013 Pau CEDEX, France

Teresa Cai
Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
Hong Kong, China

Xinyu Chen
Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
Hong Kong, China

Giovanni Denaro
Università di Milano-Bicocca, Dipartimento di Informatica Sistemistica e Comunicazione
via Bicocca degli Arcimboldi 8, I-20126
Milano, Italy

Anne Eerola
Department of Computer Science and Applied Mathematics
University of Kuopio P.O.Box 1627
70211 Kuopio, Finnland

Wolfgang Emmerich
University College London, Department of Computer Science
Gower Street
WC1E 6BT London, UK

Christof Fetzer
Dresden University of Technology
Dresden, Germany

Carlo Ghezzi
Politecnico di Milano – Dipartimento di Elettronica e Informazione
Piazza L. da Vinci, 32 – I20133
Milano, Italy

Engin Kirda
Distributed Systems Group, Technical University of Vienna
Argentinierstr. 8/184-I 1040
Vienna, Austria

Michael Lyu
Dept. of Computer Science and Engineering, The Chinese University of Hong Kong
Hong Kong, China

Atif Memon
Department of Computer Science and Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742, USA

Andrea Polini
Istituto di Scienza e Tecnologie dell'Informazione - "Alessandro Faedo"
Area di Ricerca del CNR di Pisa, Via Moruzzi 1
I-56124 Pisa, Italy

Neelam Soundarajan
Computer Science and Engineering
Ohio State University
Columbus, OH 43210, USA

Benjamin Tyler
Computer Science and Engineering
Ohio State University
Columbus, OH 43210, USA

Hong Zhu
Department of Computing
Oxford Brookes University
Wheatley campus
Oxford OX33 1HX, UK

Again, we want to thank all authors and committee members.

Leipzig,
April 2004

Sami Beydeda
Volker Gruhn

Contents