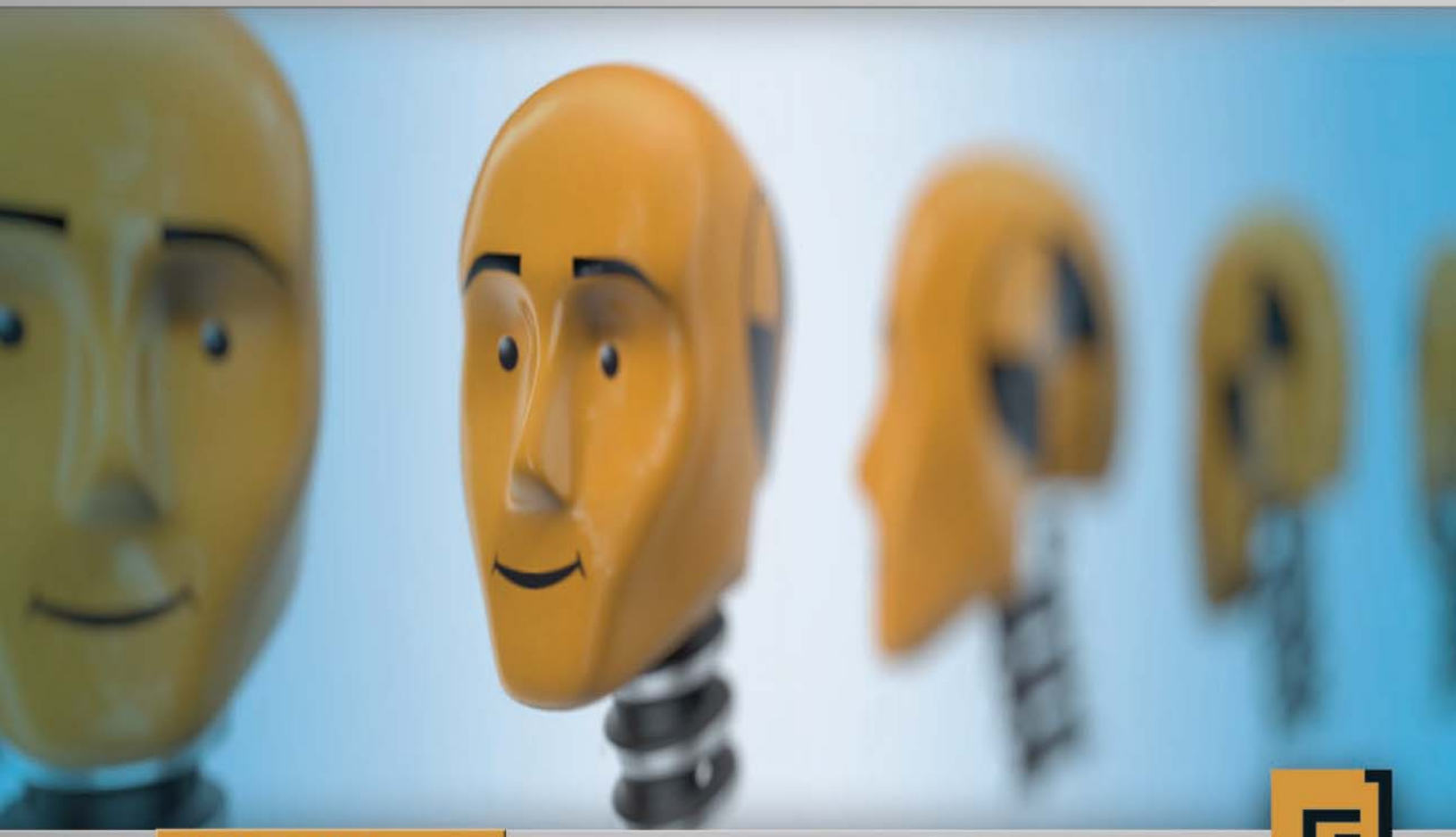
dpunkt.verlag



5th Edition

Andreas Spillner · Tilo Linz

Software Testing Foundations

A Study Guide for the Certified Tester Exam

- Foundation Level
- ISTQB® Compliant

About the Authors

Andreas Spillner is emeritus professor of computer science at the University of Applied Sciences Bremen. During the 1990s and early 2000s he spent 10 years as spokesman for the *TAV* (Test, Analysis, and Verification) group at the Gesellschaft für Informatik (German Computer Science Society) that he also helped to found. He is a founder member of the *German Testing Board* and was made an honorary member in 2009. He was made a fellow of the *Gesellschaft für Informatik* in 2007. His software specialty areas are technology, quality assurance, and testing.

Tilo Linz is co-founder and a board member of imbus AG, a leading software testing solution provider. He has been deeply involved in software testing and

quality assurance for more than 25 years. As a founding member and chairman of the *German Testing Board* and a founding member of the *International Software Testing Qualifications Board*, he has played a major role in shaping and advancing education and training in this specialist area both nationally and internationally. Tilo is the author of *Testing in Scrum* (published by Rocky Nook), which covers testing in agile projects based on the foundations presented in this book.

Andreas Spillner · Tilo Linz

Software Testing Foundations

A Study Guide for the Certified Tester Exam

- **Foundation Level**
- **ISTQB® Compliant**

5th, revised and updated Edition

dpunkt.verlag

Andreas Spillner · *andreas.spillner@hs-bremen.de*
Tilo Linz · *tilo.linz@imbus.de*

Editor: Dr. Michael Barabas / Christa Preisendanz
Translation and Copyediting: Jeremy Cloot
Layout and Type: Josef Hegele
Production Editor: Stefanie Weidner
Cover Design: Helmut Kraus, *www.exclam.de*
Printing and Binding: mediaprint solutions GmbH, 33100 Paderborn, and Lightning Source®, Ingram Content Group.

Bibliographic information published by the Deutsche Nationalbibliothek (DNB)
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data can be found on the Internet at *http://dnb.dnb.de*.

ISBN dpunkt.verlag:

Print	978-3-86490-834-7
PDF	978-3-96910-298-5
ePUB	978-3-96910-299-2
mobi	978-3-96910-300-5

ISBN Rocky Nook:

Print	978-1-68198-853-5
PDF	978-1-68198-854-2
ePUB	978-1-68198-855-9
mobi	978-1-68198-856-6

Wieblinger Weg 17
69123 Heidelberg

Title of the German Original: Basiswissen Softwaretest
Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB®-Standard
6., überarbeitete und aktualisierte Auflage 2019
ISBN 978-3-86490-583-4

Distributed in the UK and Europe by Publishers Group UK and dpunkt.verlag GmbH.

Distributed in the U.S. and all other territories by Ingram Publisher Services and Rocky Nook, Inc.

This book is printed on acid-free paper.
Printed in Germany and in the United States.

5 4 3 2 1 0

Preface to the 5th Edition

Bestseller

The first edition of the book was published in German at the end of 2002. Since then, *Basiswissen Softwaretest* has been the best-selling book on software testing in the German-speaking world.

This 5th edition in English has been comprehensively revised and updated. It is based on the latest (6th) edition of the German-language book and the current 2018 *ISTQB® Certified Tester – Foundation Level* syllabus.

The Certified Tester training scheme

The *Certified Tester* qualification scheme is extremely successful and is widely recognized and accepted within the IT industry. It has become the de facto global standard for software testing and quality assurance education. By the end of 2020 there were over 955,000 exams taken and more than 721,000 certifications issued in 129 countries around the world [URL: ISTQB]. Many IT employment ads for beginners and experienced workers reflect this, and certified training is often an obligatory requirement. The *Certified Tester* scheme is also part of the curriculum at many universities and technical colleges.

Grass-roots knowledge required in the IT world

In spite of this rapid development, there is a lot of the grass-roots knowledge in the field of computer science that

doesn’t change very much over the years. We take the *Foundations* part of our book title seriously and don’t discuss topics that have yet to be proven in everyday practice. Specialist topics such as web app or embedded system testing are not part of these foundations.

What’s new?

This 5th edition of *Software Testing Foundations* has been comprehensively revised and extended, and its content brought completely up to date.

Side notes are not part of the official syllabus

The latest revision of the ISTQB® syllabus has seen some test techniques shifted to higher training levels, so these are no longer part of the *Foundations* syllabus. However, we have kept the corresponding sections in the book and have highlighted them as side notes. If you are using the book exclusively for exam preparation you can simply skip the side note sections.

New test techniques included

Many readers have told us that they use the book for reference in their everyday work scenarios. This is why we have included a number of additional test techniques that do not appear in the *Foundations* syllabus. These include techniques such as pair-wise testing that weren’t covered in previous editions.

The case study that illustrates the implementation of the test techniques has been adapted and comprehensively updated.

We have revised the lists of standards to reflect the changes made by the introduction of ISO 29119, and all the URLs referenced in the text have been updated too.

Online resources

Any future changes to the syllabus and the glossary that affect the book text can be found on our website [URL: Softwaretest Knowledge], where you will also find exercises that relate to the individual chapters in the book. Any necessary corrections or additions to the book text are also made available at the website.

Thanks

For a book like this, success is rarely down to the authors alone, and we would like to thank all our colleagues at the German Testing Board and the *International Software Testing Qualifications Board*, without whom the *Certified Tester* program would never have achieved the global success that it enjoys. Many thanks also to Hans Schaefer, our co-author of the previous four editions of the book, for his constructive cooperation.

We would further like to thank our readers for their many comments and reviews, which have encouraged us during our work and motivated us to keep getting better. Heartfelt thanks also go to our editor Christa Preisendanz and the entire team at dpunkt.verlag for years of successful cooperation.

We wish all our readers success in the practical implementation of the testing approaches described in the book and—if you are using the book to prepare for the *Certified Tester Foundation Level* exam—we wish you every success in answering the exam questions.

Andreas Spillner and *Tilo Linz*
May 2021

Foreword by Yaron Tsubery

The software systems industry continues to grow rapidly and, especially over the last two decades, exponentially. Market requirements and a growing appetite for exciting new challenges have fuelled the development of new software technologies. These new opportunities affect almost everyone on our planet and reach us primarily via the internet and, subsequently, via smart devices and technologies.

The need for software that is easy to create and maintain has caused many key industries—such as health, automotive, defense, and finance— to open up and become visible to the world via applications and/or web interfaces. Alongside these traditional domains, new types of services (such as social media and e-commerce) have appeared and thrived on the global market. The rapid growth and enormous demands involved in introducing new software-based products that greatly impact our lifestyles and our wellbeing require new and faster ways of producing software solutions.

This situation has created a market in which multiple companies compete for market share with extremely similar products. Such competition is beneficial to consumers (i.e., software users) and, as a result, software-based products have started to become commoditized. Software manufacturers have begun to think more economically, generating increased revenues using fewer

resources (i.e., doing more with less). Continual introduction of new products into our daily lives has given rise to the “agile” design and production ethos—driving a cultural change in the tradition software development life cycle, as well as pushing forward the necessity of more and early automatic tests (e.g. as driven by the DevOps movement)—that is increasingly commonplace in today’s software industry, while the business leaders behind software-based products have understood that the world is becoming smaller and that competition is getting fiercer all the time. An increasingly short time to market is essential not only for generating revenue, but also simply to survive in today’s market. Successful and innovative companies understand that they need to put the customer first if they want to maintain product quality, generate brand loyalty, and increase their market share. In other words, the software industry has understood the importance of the customer to the overall product life cycle.

We in the software testing business have always known the importance of quality to the customer, because part of our job is to represent the customer’s point of view. The challenges we face have grown with the complexity of software products, and we sometimes still find ourselves having to justify the necessity for software testing, even if it has become a largely standard practice within the software industry. Recently, the rise of software-based artificial intelligence (AI)—such as software enhancement in robots and autonomous devices—has created a whole new set of challenges.

Software testing is an extremely important factor in the industry. Alongside controlling costs and quality, the main issue is customer focus. Preserving a healthy balance between cost and quality is an essential customer requirement, making it critical to have well-trained and

highly professional people assigned to quality and software testing roles. Recruiting skilled professionals is the key to success. The primary factors we look for when recruiting are related to a person's knowledge and skills. We look at the degree to which a person is aligned with the software testing profession, and with the required technology and industry domain (such as web, mobile, medical devices, finance, automotive, and so on). We also have to ask ourselves whether a person is suited to work in the product domain itself (for example, when candidates come from competitors). Communications and soft skills that fit in with the team/group/company are important too. In the case of industry newcomers, we have to consider how much potential a person has. This book teaches the fundamentals of software testing and provides a solid basis for enhancing your knowledge and experience through constant learning from external sources, your own personal experience, and from others.

When reading an educational book, I expect it to be sequentially structured and easy to understand. This book is based on the *Certified Tester Foundation Level (CTFL)* syllabus, which is part of the ISTQB® (International Software Testing Qualifications Board) education program. The ISTQB® has created a well-organized and systematic training program that is designed to teach and qualify software testers in a variety of roles and domains. One of the primary objectives of the ISTQB® program is to create professional and internationally accepted terminology based on knowledge and experience. The chapters in the book are designed to take you on that journey and provide you with the established and cutting-edge fundamentals necessary to becoming a successful tester. They combine comprehensive theory with detailed practical examples and side notes that will enhance and broaden your view of

software systems and how to test them. This book provides a great way to learn more about software testing for anyone who is studying the subject, thinking about joining the software testing profession, or for newcomers to the field.

For those who already have a role in software testing, the practical examples provided (based on a case study and corresponding side notes) are sure to help you learn. They provide a great basis for comparison with and application to your own real-world projects. This book contains a wealth of great ideas that will help you to build and improve your own software testing skills. The new, revised edition is based on the latest (2018) ISTQB® CTFL, which has been updated to cover agile processes and experience gained from changes that have taken place within the industry over the last few years. It also includes references to the other syllabi and professional content upon which it is based, and an updated version of the case study introduced in earlier editions. The case study is based on a multilayer solution that includes both specific and general technical aspects of software system architecture. The case study in this edition is based on a new-generation version of the system detailed in previous editions, thus enabling you to learn from a practical, project-based viewpoint.

The world is changing fast every day. Some of the technologies that we use today will become obsolete within a few years and the products we build will probably become obsolete even sooner. Software is an integral and essential part of virtually all the technology that surrounds us. Along with growth and expansion in the artificial intelligence (AI) arena and other new technologies that have yet to be introduced, this continual change offers new and exciting opportunities for the software testing profession. We are sure to find ourselves tuning our

knowledge and experience in various ways, and we may even find ourselves teaching and coaching not only humans but also machines and systems that test products for us.

The fundamental knowledge, grass-roots experience, and practical examples provided by this book will prepare you for the ever-changing world and will shape your knowledge to enable you to test better and, in the future, perhaps pass on your knowledge to others.

I wish you satisfying and fruitful reading.

Yaron Tsubery
Former ISTQB® President
President ITCB®

Overview

Appendices

Contents

1 Introduction

Software is everywhere! Nowadays there are virtually no devices, machines, or systems that are not partially or entirely controlled by software. Important functionality in cars—such as engine or gear control—have long been software-based, and these are now being complemented by increasingly smart software-based driver assist systems, anti-lock brake systems, parking aids, lane departure systems and, perhaps most importantly, autonomous driving systems. Software and software quality therefore not only govern how large parts of our lives function, they are also increasingly important factors in our everyday safety and wellbeing.

Equally, the smooth running of countless companies today relies largely on the reliability of the software systems that control major processes or individual activities. Software therefore determines future competitiveness. For example, the speed at which an insurance company can introduce a new product, or even just a new tariff, depends on the speed at which the corresponding IT systems can be adapted or expanded.

High dependency on reliable software

Quality has therefore become a crucial factor for the success of products and companies in the fields of both technical and commercial software.

Most companies have recognized their dependence on software, whether relying on the functionality of existing systems or the introduction of new and better ones. Companies therefore constantly invest in their own development skills and improved system quality. One way to achieve these objectives is to introduce systematic software evaluation and testing procedures. Some companies already have comprehensive and strict testing procedures in place, but many projects still suffer from a lack of basic knowledge regarding the capacity and usefulness of software testing procedures.

Grass-roots knowledge of structured evaluation and testing

This book aims to provide the basic knowledge necessary to set up structured, systematic software evaluation and testing techniques that will help you improve overall software quality.

This book does not presume previous knowledge of software quality assurance. It is designed for reference but can also be used for self-study. The text includes a single, continuous case study that provides explanations and practical solutions for each of the topics covered.

This book is aimed at all software testers in all types of companies who want to develop a solid foundation for their work. It is also for programmers and developers who have taken over (or are about to take over) existing test scenarios, and it is also aimed at project managers who are responsible for budgeting and overall procedural improvement. Additionally, it offers support for career changers in IT-related fields and people involved in application approval, implementation, and development.

Especially in IT, lifelong learning is essential, and software testing courses are offered by a broad range of companies and individuals. Universities, too, are

increasingly offering testing courses, and this book is aimed at teachers and students alike.

Certification program for software testers

The ISTQB® *Certified Tester* program is today seen as the worldwide standard for software testing and quality assurance training. The ISTQB® (*International Software Testing Qualifications Board*) [URL: ISTQB] coordinates qualification activities in individual countries and ensures the global consistency and comparability of the syllabi and exam papers. National *Testing Boards* are responsible for publishing and maintaining local content as well as the organization and supervision of exams. They also approve courses and offer accreditation for training providers. Testing Boards therefore guarantee that courses are of a consistently high standard and that participants end up with an internationally recognized certificate. Members of the Testing Boards include training providers, testing experts from industrial and consulting firms, and university lecturers. They also include representatives from trade associations.

Three-stage training scheme

The *Certified Tester* training scheme is made up of units with three levels of qualification. For more details, see the ISTQB® [URL: ISTQB] website. The basics of software testing are described in the *Foundation Level* syllabus. You can then move on to take the *Advanced Level* exam, which offers a deeper understanding of evaluation and testing skills. The *Expert Level* certificate is aimed at experienced software testing professionals, and consists of a set of modules that cover various advanced topics (see also section 6.1.2). In addition, there are syllabi for agile software development (foundation and advanced level) as

well as special topics from the testing area (for example, Security Tester, Model-Based Tester, Automotive Software Tester).

This book covers the contents of the *Foundation Level* syllabus. You can use the book for self-study or in conjunction with an approved course.

Chapter overview

The topics covered in this book and the basic content of the F*oundation Certificate* course are as follows:

Software testing basics

Chapter 2 discusses the basics of software testing. Alongside the concepts of when to test, the objectives to aim for, and the required testing thoroughness, it also addresses the basic concepts of testing processes. We also talk about the psychological difficulties that can arise when you are looking for errors in your own work.

Lifecycle testing

Chapter 3 introduces common development lifecycle models (sequential, iterative, incremental, agile) and explains the role that testing plays in each. The various test types and test levels are explained, and we investigate the difference between functional and non-functional testing. We also look at regression testing.

Static testing

Static testing (i.e., tests during which the test object is not executed) are introduced in Chapter 4. Reviews and static tests are used successfully by many organizations, and we go into detail on the various approaches you can take.

Dynamic testing

Chapter 5 addresses testing in a stricter sense and discusses “black-box” and “white-box” dynamic testing techniques. Various test techniques and methods are explained in detail for both. We wrap up this chapter by looking at when it makes sense to augment common testing techniques using experience-based or intuitive testing techniques.

Test management

Chapter 6 discusses the organizational skills and tasks that you need to consider when managing test processes. We also look at the requirements for defect and configuration management, and wind up with a look at the economics of testing.

Test tools

Testing software without the use of dedicated tools is time-consuming and extremely costly. Chapter 7 introduces various types of testing tools and discusses how to choose and implement the right tools for the job you are doing.

Most of the processes described in this book are illustrated using a case study based on the following scenario:

Case Study: Virtual-ShowRoom VSR-II

A car manufacturer has been running an electronic sales system called *VirtualShowRoom (VSR)* for over a decade. The system runs at all the company’s dealers worldwide:

- Customers can configure their own vehicle (model, color, extras, and so on) on a computer, either alone or assisted by a salesperson. The system displays the available options and immediately calculates the corresponding price. This functionality is performed by the *DreamCar* module.

- Once the customer has selected a configuration, he can then select optimal financing using the *EasyFinance* module, order the vehicle using the *JustInTime* module, and select appropriate insurance using the *NoRisk* module. The *FactBook* module manages all customer and contract data.

The manufacturer's sales and marketing department has decided to update the system and has defined the following objectives:

- *VSR* is a traditional client-server system. The new *VSR-II* system is to be web-based and needs to be accessible via a browser window on any type of device (desktop, tablet, or smartphone).
- The *DreamCar, EasyFinance, FactBook, JustInTime*, and *NoRisk* modules will be ported to the new technology base and, during the process, will be expanded to varying degrees.
- The new *ConnectedCar* module is to be integrated into the system. This module collects and manages status data for all vehicles sold, and communicates data relating to scheduled maintenance and repairs to the driver as well as to the dealership and/or service partner. It also provides the driver with various additional bookable services, such as a helpdesk and emergency services. Vehicle software can be updated and activated "over the air".
- Each of the five existing modules will be ported and developed by a dedicated team. An additional team will develop the new *ConnectedCar* module. The project employs a total of 60 developers and other specialists from internal company departments as well as a number of external software companies.

- The teams will work using the *Scrum* principles of agile development. This agile approach requires each module to be tested during each iteration. The system is to be delivered incrementally.
- In order to avoid complex repeat data comparisons between the old and new systems, *VSR-II* will only go live once it is able to duplicate the functionality provided by the original *VSR* system.

Within the scope of the project and the agile approach, most project participants will be confronted or entrusted with test tasks to varying degrees. This book provides the basic knowledge of the test techniques and processes required to perform these tasks. Figure 1-1 shows an overview of the planned *VSR-II* system.

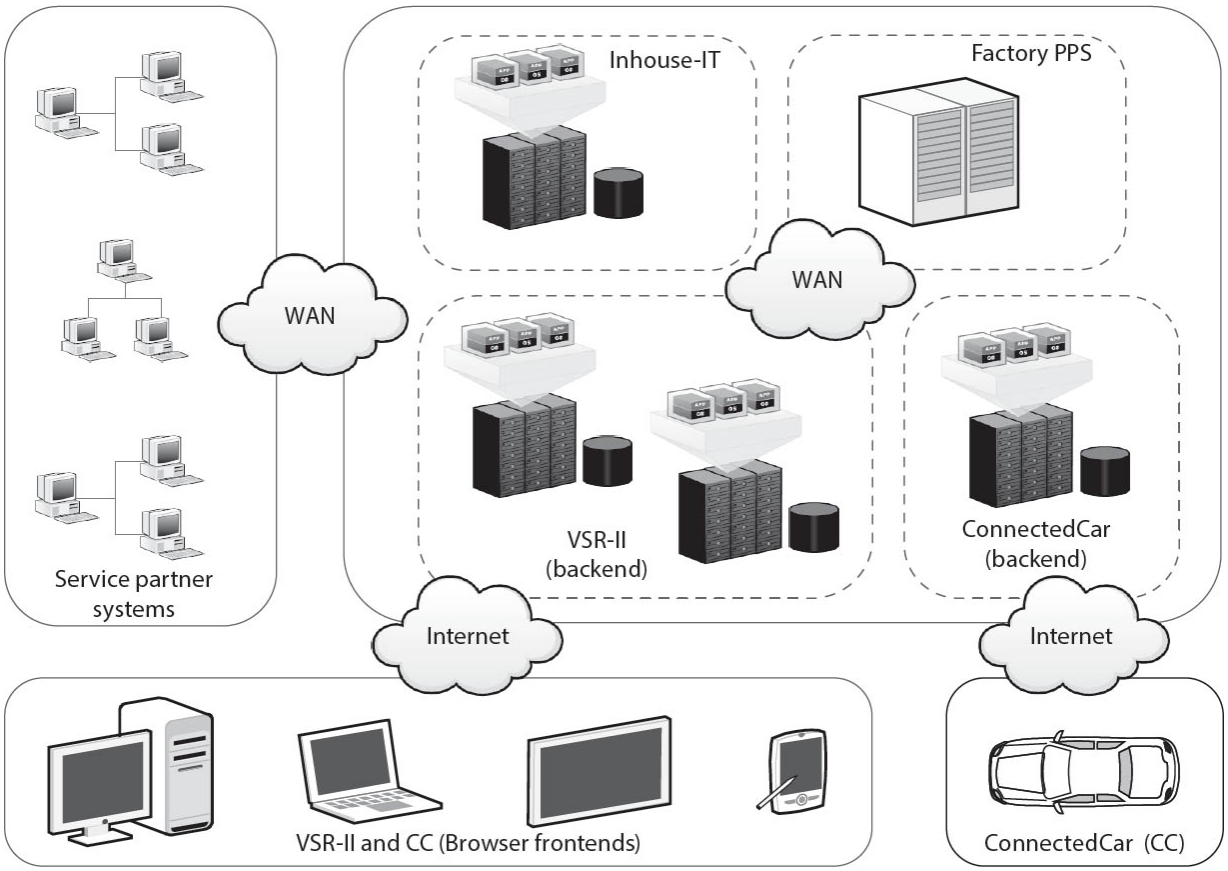


Fig. 1-1 *VSR-II overview*

Certified Tester syllabus and exam

The appendices at the end of the book include references to the syllabus and *Certified Tester* exam, a glossary, and a bibliography. Sections of the text that go beyond the scope of the syllabus are marked as side notes.

The book's website

The book's website [URL: Softwaretest Knowledge] includes sample exam questions relating to each chapter, updates and addenda to the text, and references to other books by authors whose work supports the *Certified Tester* training scheme.

Web-based Training System vsr.testbench.com

We have put a free implementation of *VSR-II* as a test object online for training purposes[1]. It reproduces the *VSR-II* examples included in the book on a realistic, executable system, so you can "test" live to find the software bugs hidden in *VSR-II* by applying the test techniques presented in the book. It takes just a few mouse clicks to get started:

1. Open your browser and load *vsr.testbench.com*
2. Create your personal *VSR-II* training workspace
3. Log into your *VSR-II* workspace and start

Fig. 1-2 *VSR-II Training System Login-Screen*

Also included in your registration for a *VSR-II* training workspace is a free basic license for the test management system *TestBench CS*, which includes the *VSR-II* test specification as a demo project and several of the *VSR-II* test cases presented in the book.

You can use *TestBench CS* not only for learning and training, but also for efficient testing of your own "real" software. A description of all features can be found at [URL: TestBench CS].

Many thanks to our colleagues at imbus Academy, imbus JumpStart and imbus TestBench CS Development Team for this awesome implementation of the VSR-II Case Study as a web-based training system.