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.

Case Study: VirtualShowRoom VSR-II

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

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.

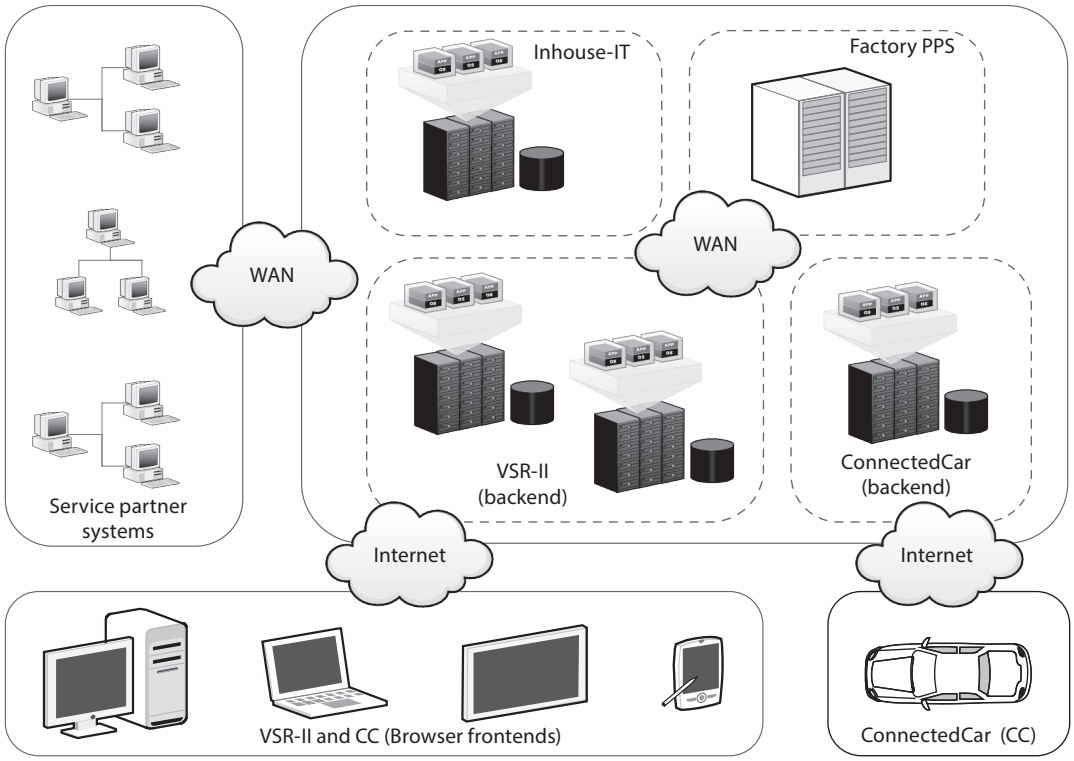

Fig. 1-1 *VSR-II overview*

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.

Certified Tester syllabus and exam

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.

The book's website

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:

Web-based Training System vsr.testbench.com

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.