

Title of Paper

Pragmatic Test Case Generation

Presenter

Dr. Anne Kramer, sepp.med gmbh

Instructional Level

Introductory Intermediate Advanced

Target Group

test managers, test engineers, QA manager

Keywords

- model-based test design
 - test case generation
 - risk-based approach
-

Abstract

The interest in model-based test design from both industrial and scientific communities is steadily growing. Test case generators deriving test cases from test design models are supposed to provide a flexible approach to reusable and maintainable test specifications. The ideal everybody seems to be striving for is a fully automated way from model to test case.

In this paper we present a more pragmatic approach to test case generation. We will demonstrate that test design models can be written in very different manners, depending on the results that shall be obtained. We state that it is not necessary to work with a fully automated tool chain. Also, not all the stakeholders have to be familiar with modeling techniques. We use model-based test design internally, even if the deliverable shall be a paper-based test specification. This allows us to react rapidly on changing requirements and specific customer needs. Also, the amount of errors is considerably reduced due to the automatisms.

Depending on the situation, the focus of testing activities can lie on different aspects. During verification, testing must provide a high degree of confidence that the system works as specified. Thus, test specifications are mainly based on functional requirements. Unit-Test design models will be very similar to the corresponding software design models. Its state diagrams will reflect the exact behavior of the system, plus specify testing aspects such as error situations or boundary tests.

For validation and customer acceptance tests, it is more difficult to write a test design model purely based on the requirements specification. User requirements tend to be less precise since they have been defined at a very early stage of the project. Therefore, test design models are better based on workflows.

While these two cases differ in their degree of detail, they have in common that the tests shall fully cover the requirements. These requirements are usually linked to the entire use case and not to specific states within this use case.

In some domains a risk-based approach to testing is adopted. Instead of verifying the implementation of requirements, the tests are defined either to test implemented measures against identified risks, or directly as a measure against the risk. Thus, test cases are focused on risks rather than on requirements.

For test design models, this represents a difficulty. On the one hand, it is still necessary to model the system behavior entirely for better understanding. On the other hand, this introduces unwanted additional steps and verification points in the test specification, i.e. test steps that are not related to any risk. Therefore, the test design model will be less concentrated on the system and more on the resulting test specification. In other words, states and transitions might be omitted or ignored, even if they are relevant for the functional behavior, simply because they do not represent any risk.

Writing test design models in a risk-based approach requires a certain degree of pragmatism. The model is used as a graphical way to write understandable and maintainable test specifications, rather than to represent a full description of the system. The risks are associated to the states, and not to the test use cases.

As a consultant, we must be able to generate test specifications in diverse formats depending on the customer's tools and templates. For standard test management tools, test case generators offer the corresponding interfaces, but for paper-based test specifications, the situation is more difficult. Therefore, we generate test cases in a semi-automated way.

First, the test design model is written in UML using a standard modeling tool. The various test steps and verification points are defined as incoming and outgoing events within a state diagram. The risk that is addressed is referenced as particular test step just before the corresponding actions. Preconditions (e.g. system configuration, required privileges) and other required global information are attached to the use case as entities.

Second, test cases are generated using our proprietary test case generator `.getmore`. Depending on the test strategy, we can choose between full path and/or transition coverage. For risk-based models, test case generation with user-guided paths proved to be best, especially if the number of test steps shall be as small as possible. "User-guided" means that the user may select manually for every state transition which way to go. The choice of the path depends on the prioritization and may influence the model itself. Transitions might be added to the model just to find a rapid end for a test case.

Test steps (and possibly other attributes) are then automatically exported by `.getmore` and imported into the specific application used by the customer. These can be standard test management tools, test automation tools or simply MS Office templates. The import is done more or less automated, depending on the project. That way, we fully keep the advantages of model-based test design even if the resulting review object, i.e. the test specification is still paper-based. The customer must not necessarily be familiar with the method, as the model is discussed on only if they ask for it. However, experience shows that the graphical representation is easily understood even by untrained persons and that it is a good basis for discussions, leading to improved specifications.

We successfully used the pragmatic approach in different projects. It allows us to keep a high degree of flexibility and to reduce test design efforts. For example, it may be necessary to modify the selected paths through the model considerably after discussions with the customer. We can also easily increase (or decrease) the degree of detail in parts of the model. The modified test cases can then be generated again with few manual steps.

Biography

Anne Kramer was born in 1967 in Bremen (Germany). She studied Physics at the University of Hamburg where she obtained her diploma in 1992. In 1995 she graduated at the Université Joseph Fourier in Grenoble (France).

Immediately afterwards she started working for Schlumberger Systems in Paris – first as software developer for smart card tools, than as project manager for point of sales terminals. In 2001 she be-

came technical project leader at sepp.med (near Erlangen, Germany).

The sepp.med gmbh has specialized on IT solutions with integrated quality assurance in complex, safety-relevant domains. A main aspect of the consultant activity lies in introduction and support of processes, methods and tool-based automatisms.

Contact information of Presenter

Full Postal Address: Dr. Anne Kramer, Gewerbering 9, D-91341 Röttenbach

Affiliation: sepp.med GmbH

E-mail Address: anne.kramer@seppmed.de

Phone Number: +49 (0)9195 / 931-207

Fax Number: +49 (0)9195 / 931-300
