16

December 2011

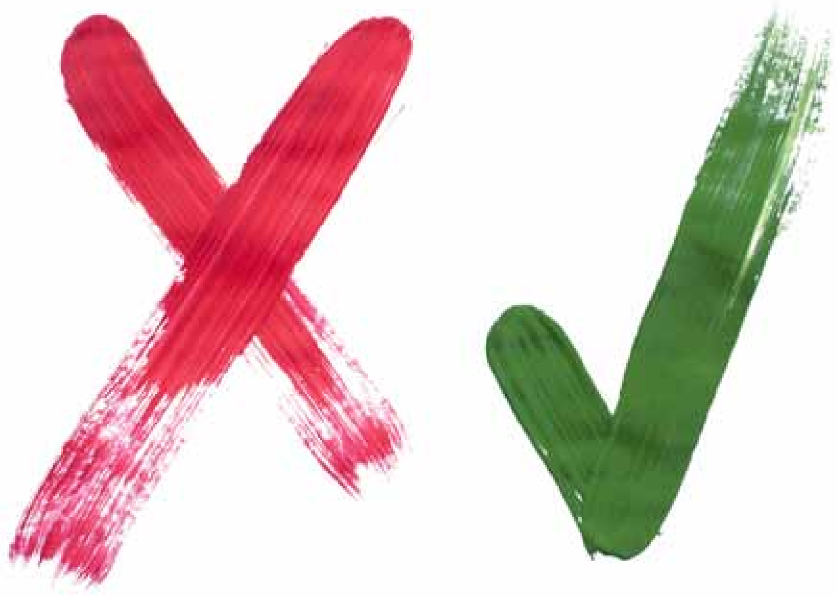
# testing experience

The Magazine for Professional Testers

ISSN 1866-5705 www.testingexperience.com free digital version print version 8,00 € printed in Germany

## The Future of Testing

© iStockphoto.com/pidjoe



# All About Quality

*by Leo Smits*

Software testing is all about quality. As Kaner states [1]: 'Software testing is an empirical technical investigation conducted to provide stakeholders with information about the quality of the product or service under test.' Stakeholders need to have insight in quality, software testers provide the required information for this. The essential questions then are: What is quality of a product or service under test exactly? How can it be determined? What information about quality is exactly needed?

### *Quality and Decisions*

There are many different definitions of quality. Garvin distinguishes five distinct approaches to the definition of quality [2]: the transcendent approach, the product-based approach, the user-based approach, the manufacturing-based approach and the value-based approach. Each approach differs significantly from the other approaches and this is reflected in the various definitions of quality. All approaches offer valuable definitions of quality for software testers.

Here however, a different perspective is presented. This perspective centers on 'decision'. The definition of quality is derived from the role it has in the context of decisions. This leads to a definition that significantly differs from the definitions of the other five approaches. On the other hand, there are common concepts. Especially the user-based approach does overlap.

First take a look at 'decision'. Individuals make decisions constantly. In many cases the decision is as trivial as 'Do I want to drink coffee or tea?' and 'What shall I take home for dinner from the supermarket?'. Sometimes individuals are faced with more difficult decisions like 'Do I purchase this car, or not?'. The types of decisions and their complexity may differ, they all basically share the same activities: identification of the available options; ascribing values to all options; order the options by value and pick the best option (the one with the highest value). An individual does not make perfect decisions. He possibly does not identify all options. Also, the values he ascribes to the various options are based on his capability to do so. But he will carry out these activities. (Note that not making a decision implicitly equals a decision.) Otherwise he cannot act.

Hence it can be concluded for decisions that:

- options must be identified;
- values must be ascribed to the options.

Value is a term that will be discussed in more detail later, but for now it is important to highlight two aspects:

- Value is subjective. It is the value as perceived by the individual. Individual A may ascribe a different value to an option than individual B. This matches the user-based approach which has the premise that quality 'lies in the eye of the beholder' [2].
- Value depends on the context of the decision to be made. Example: An individual prefers the taste of coffee to the taste of tea, and at the same time he does not want his sleep to be disturbed by too much caffeine. He will then ascribe a higher value to coffee than to tea at 9:00 AM but he will do the other way round at 9:00 PM.

What does this mean for 'quality'? From the perspective of decision making, quality has no other meaning than the value of an option as compared to the values of other options. And like 'value' to which quality is related it depends on the individual and the context of the decision to be made. There is no absolute quality in an object, idea, etc. itself. So quality can only be defined as a comparative term.

The 'comparative' aspect of quality from another perspective: Imagine a person looking at a painting. This person has never seen a painting before in his whole life, or in other words: This is the first painting he has ever seen and thus the only painting he knows. When asked, what quality would he ascribe to that particular painting? How many 'stars' would he give the painting? The fact is he cannot reasonably answer this question about quality since he knows no other paintings to compare to.

Summarized, from the decision-perspective quality contains three characteristics:

1. Quality is the value of an option as compared to the values of other options
2. Quality is based on perceived values, differing per individual

3. Quality is determined in the context of the decision to be made

An object, idea, etc. does only contain quality in the context of a decision to be made (i.e. it has no absolute quality). Within that context the object, idea, etc. is an option. It is therefore more correct to speak of 'the quality of an option' than 'the quality of an object'. The definition for quality from a decision-perspective can consequently be assembled as follows. Quality of an option is the value of that option in comparison to the values of other options, as perceived by a person and within the context of a decision to be made.

### *Quality and Software Testing*

If quality of an option only exists in comparison to other items then what does that mean for software testing? What is the 'option' in the context of software testing and what are the other options it must be compared to?

To answer these questions it must first be clear what decisions have to be made. Let's assume that the information about quality that must be provided is intended to support a release decision for a system under test (SUT). A release decision in general has two basic options: 'Release the SUT' or 'Do not release the SUT'. There may also be other options, e.g. 'Release the SUT except for subsystem A'. Looking at the options for a release decision, options in this case are scenarios. The choice is of the kind 'What to do next?' Therefore in the context of software testing the word 'scenario' is used for 'option'.

The two (or more) scenarios can be compared. Scenarios are usable in the context of the decision to be made. The conclusion is that software testers need to identify the different scenarios for their project and then determine the value of all identified scenarios. In doing so, quality is determined in the context of a release decision and quality is determined as a comparative term.

It still leaves the question what 'value' is unanswered. It also raises the question how value can be calculated so that it covers the perception of a group of stakeholders. This challenge is caused by the fact that quality is based on perceived values of individuals whereas the release decision is not made by one individual.

### *Determining the Value of an Option*

The first step in making a decision is determining the values of all options (considering the identification of all options as completed). What unit of measure can be used for 'value' to make the comparison in the following step possible?

Let's first look at an example: A couple goes into town to have dinner in a restaurant. They have all evening to spend and this dinner itself is the reason for their visit to the town. There are two restaurants available, one expensive restaurant with a highly acclaimed cuisine, the other being a pizzeria with affordable cooking. For the purpose of their visit the couple will decide by asking themselves 'Which restaurant serves the best food?' not 'Which restaurant is the most inexpensive?'. So the expensive restaurant has the highest quality. A week later they visit the same town again. This time to go to the movies. They have one hour left before the movie starts and still need something to eat. In this situation the couple decides by asking 'Which restaurant is the most inexpensive?'. This time the pizzeria has the highest quality. The 'highest quality' more exactly: Highest quality for them, and in the context of the current decision to make.

This example shows that quality depends on the decision to be made and that subsequently 'value' must be related to that decision as well. In the example 'value' is simply based on one simple question. In practice decision making is often more complex, but it boils down to the same idea: The unit of measurement being used supports the decision to be made.

### *Values in Software Testing*

What unit of measurement supports the decision making for software products? What unit of measurement makes the best comparison for the scenarios?

To support the decision making process it is required that the type of value is accepted by all people involved and that it allows for a complete and still feasible comparison. This is possible when value is calculated in terms of financial value. Financial value eliminates the perceived value to a great extent. This is important because a release decision is not made by an individual but by a group. It is also in most cases relatively easy to express various aspects as a financial value. And even for difficult aspects like damage to reputation it is still possible to do so.

Looking a bit more in detail, the financial value of a scenario consists of two parts. First the SUT has benefits for the organization. Second it has costs that diminish the financial value: defects and unmitigated risks. The total value of the scenario is simply the subtraction of these two (value = benefits – costs). Neglecting the benefits (i.e. only reporting on defects and unmitigated risks) leads to an incomplete value, for example: If a choice is offered between two cars for the same price, one without damage and one with some exterior damage, then what would be the best pick? And what if the first car is a cheap small car that has run 200,000 km and the second car is a luxury car that has run only 50,000 km?

Conclusions: In software testing the value can be expressed as a financial value. The unit of measurement would be for example Euro. Also, it is important to look at both costs and benefits of the scenarios. Remember that quality only exists as a comparison of scenarios and that therefore the values of all scenarios must be calculated.

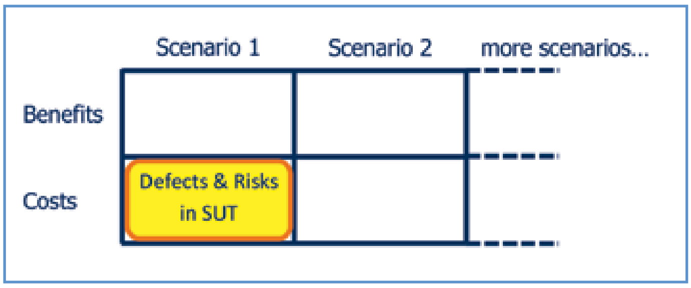


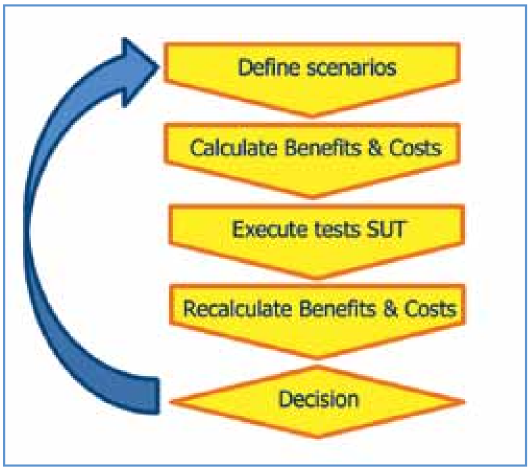
Figure 1 shows the values that must be calculated when software testing provides information about quality. It also shows that value consists of both benefits and costs. If information about quality is solely based on defects and unmitigated risks in the system under test (SUT), then only partial information is given. Also if quality is based only on the SUT and not on all scenarios, then partial information is given. This is shown as the yellow text box in the figure.

### *Implementation*

The approach from the perspective of decision has its implications on the work of software testers. It requires additional activities from the software testing team, in particular:

- Identification of scenarios
- Calculation of values for all scenarios

The easiest way to start with identification of scenarios is to take only two scenarios to start with. It is always possible to add other scenarios later. Calculation of the values for the scenarios can be done as soon as the scenarios have been identified, before test execution starts. It provides insight in the values at an early stage, even if this information is not fully complete. When all tests have been executed, the costs for the SUT are known and the calculations must be updated. At that point the software testing team is able to provide the information about quality to the stakeholders. They can make a decision, i.e. they can choose from the scenarios. If the SUT is released, then the process ends. Otherwise the whole sequence starts all over. Two remarks can be made here. First, it might be desirable to add or remove scenarios when going into a next iteration. Second, based on the values of the scenarios it is possible to consider which defects should be fixed based on the extra value this will deliver. Note that the 'Release the SUT' scenario in principle only needs to have a higher value than the other scenario(s). There is no need to maximize the value of 'Release the SUT'. Figure 2 shows the suggested process as described here.



Some suggestions that may aid the implementation:

- Much information on value may already be available. It provides an easy starting point.
- Do not calculate values in a very detailed manner from the start. First gain experience with the new process and build metrics to refine calculations later.
- Use standard costs for defects with low severity (e.g. 'Inconvenient' and 'Cosmetic').
- Keep in mind that costs of a scenario are those costs that are made when the SUT is running in a production environment. It does not include the costs of the project to develop and test the SUT.

## Conclusion

A definition of quality from the perspective of decision fits the goal of software testing well.

Following the implications of the new quality definition it appears that software testers must broaden the scope of their work. It will make work more complicated to a certain extent, but it results in providing complete information about quality. This in turn will allow decisions to be made more accurately. If software testers do not provide the complete information (i.e. leaving out benefits and scenarios), others will. That is not what should happen: Providing information about quality is the domain of software testers.

## References

[1] Cem Kaner, J.D., Ph.D.: Exploratory Testing - Keynote at QAI, November 17, 2006

[2] Garvin, David A.: What Does "Product Quality" Really Mean? in Sloan Management Review Volume 26 Number 1, Fall 1984

## › biography

**Leo Smits**
*has 10 years of experience in software testing and 15 years in ICT. He has worked both as a software tester and as a test manager. As a test manager he has led the testing for several major ICT projects. Apart from software testing he shows an interest in information analysis, requirements engineering and the Software Development Life Cycle. Leo is currently working as a test manager for a Dutch insurance company. He is the owner of Smits Haltna and lives in the Netherlands.*