

te testing experience

The Magazine for Professional Testers

printed in Germany

print version 8,00 €

Performance Testing

free digital version

www.testingexperience.com

ISSN 1866-5705



We turn 10!

Where are the non-functional requirements?

by Erik van Veenendaal

Column

Although this issue of Testing Experience focuses on performance testing specifically, I would like to look at non-functionals and their state-of-the-practice from a broader perspective. Let me start with a quote:

“... The increasing intensive use of software products in society as a whole, but especially in business environments, has made the IT user aware of the possibilities of software products. Users of software products have pushed quality to the forefront of their wishes. They can no longer be satisfied with software products that ‘only’ meet the defined functional requirements, that are delivered in time and at reasonable costs. Users of software products ask for clear, objective and quantifiable software quality.”

This quote from Dr. ir. Jos Trienekens (associate professor at the Eindhoven University of Technology) dates back to the early nineties (over 15 years ago!). What he basically stated is that quality is not only determined by the functionality of the product, but largely also by its non-functional characteristics, such as the reliability, usability or portability. I believe that in principle we all agree to this statement. However, what, if anything, has happened in the last 15 years regarding the non-functionals in our real-life projects?

Requirements perspective

One cannot test quality into a product. Quality starts with clear and unambiguous requirements. Most of the requirements specifications that I have seen, have very limited focus on non-functional requirements. Just on the last page they state “it shall be usable”. Good luck !! Maybe not so strange, if we consider that for instance the IREB Foundation syllabus for requirements engineering certification pays little attention on how to specify measurable non-functional requirements. And the well-known standard ISO 9126 was to be expanded to also define how to write requirements based on the non-functional quality characteristic terminology defined by the standard. But so far there have been no results.

Engineering perspective

Of course translating non-functional requirements into design and code is a major challenge. What engineering measures need to be taken to enhance the portability or reliability of a product? There is a whole community built around reliability engineering, and also for other non-functionals, such as maintainability, there are supporting techniques, such as code metrics and coding standards. In fact, I believe there is a lot available to support engineers in doing a good job provided they are given the time by management to do so. They tend to focus on more technically oriented non-functionals though. However, why should they worry, if no or only a few vague non-functional requirements are defined?

Testing perspective

How many testers actually spend more than 5% of their time testing non-functionals? How many testers know (which of course does not mean they can apply!) more than one non-functional technique for each non-functional characteristic? [On the other hand: Sit down with your test team and try to identify three different test design techniques for reliability, maintainability and usability. Would you succeed?] It is a good thing that non-functional characteristics are included in the ISTQB Advanced Level syllabus, especially within the Technical Test Analyst module, and that security has been identified as an ISTQB Expert Level topic. Testers need much more knowledge and skills on this. Test managers should understand how to define non-functional acceptance criteria, perform a product risk analysis for non-functionals, etc. Some, e.g. within TMMi, state that one first has to master functional testing before starting to build knowledge and skills on non-functional testing. If a calculation is wrong, who cares about the performance and usability of the calculation function? But again, why should we worry to build those skills if there are no or only few vague non-functional requirements defined?

Business perspective

In the end we (requirements analysts, software engineers and testers) are all just providing a service to the business. If they don't care, why should we? The business should also take their responsibility in asking for non-functional requirements. One of the problems regarding non-functionals is that they are often too technical to discuss with the business. It is just not on their patch (or so they think). Perhaps it is our responsibility to educate the business representatives and explain the importance to them in a business language. In this context, new development methodologies, such as Agile/SCRUM, do not help either. Is it not great as a user to get new functionality delivered fast, e.g. every 4 weeks? However, how much non-functional engineering and testing can you do in a four-week sprint?

Food for thought

So what's next? I certainly do not have the answer, but I do believe it is a shared responsibility and each discipline will have to play their part. We are probably still in a software crisis, and only focusing on functionality being delivered faster does not always help. It sometimes pays off to deliver less functionality, but to deliver it inside more reliable software. It should not be about making business tomorrow, but also about the years to come. Ask the automotive industry; they may have recently learned this lesson. Software quality, specifically non-functional quality, is still largely underestimated, and there is a lot of work to do!

What are you doing on non-functional requirements, engineering and testing in your projects.....?



Erik van Veenendaal is a leading international consultant and trainer, and recognized expert in the area of software testing and quality management. He is the director of Improve Quality Services BV. At EuroStar 1999, 2002 and 2005, he was awarded the best tutorial presentation. In 2007 he received the European Testing Excellence Award for his contribution to the testing profession over the years. He has been working as a test manager and consultant in software quality for almost 20 years.

He has written numerous papers and a number of books, including “The Testing Practitioner”, “ISTQB Foundations of Software Testing” and “Testing according to TMap”. Erik is also a former part-time senior lecturer at the Eindhoven University of Technology, the vice-president of the International Software Testing Qualifications Board and the vice chair of the TMMi Foundation.