

Becoming a Software Testing Expert – Part 1

Posted by: *Huib Schoots*

Date: Nov 10, 2011

Category: Columns



You can learn testing

This is the first in a series of three columns. The central theme of the columns is “how do I become a software testing expert?”.

To become great in your profession, you need to learn a lot. This seems obvious. Jos van Rooyen wrote an article in Dutch entitled “[Hoe goed zijn we als tester?](#)“ (How good are we as testers?). In this article he writes: “Many people call themselves professional tester without having a solid foundation. Yes, we follow the ISTQB Foundation, etc. and think that we are professional.” I fully agree with this. [Lesson 272](#) in the book *Lessons Learned in Software Testing*: “If you can get a black belt in only two weeks, avoid fights”. You have to learn and practice a lot to become good at your profession. But that goes for everything: just think of sports or music.

Jos draws the conclusion that testers on average are good. I do not agree with his conclusion (besides from the fact that mathematically any population is average). I think testers can do much better! On [my weblog](#) I once wrote: “A lot of them claim that they are great testers. But are they? I think a lot of testers maybe aren’t that great...”

Knowledge and skills

What makes a great tester? The skills that make a great tester, I will discuss in Part 2. Testing is a profession, that is something I don’t need to explain. And it’s obvious that you need a lot of different skills to be a true professional. And skills, the [learned](#) capacity to carry out pre-determined results often with the minimum outlay of [time](#), [energy](#), need to be trained. To become an expert in a particular skill you must practice a lot, and improve yourself continuously. To be an expert in a particular field, such as software testing, you need to become proficient in many skills.

So to become an expert, knowledge is important. Knowledge you can gain in many ways and you must never stop learning! “Stagnation means decline,” they say and especially in IT this is true for me. But applying this knowledge is often difficult. Experience is of great importance. James Bach said in his presentation [Becoming a Software Testing Expert](#) “Do not confuse experience with expertise.” You can have years of experience, but how do you know that you have gained the right experience? How do you know if you do it “right”? Because let’s face it: there is a lot to be improved in our projects. We have to do much better, but how do we achieve that? And how do we know what we can improve?

Learning

Looking at the [different stages](#) of competence in a learning process: you start unconsciously incompetent. So you need to find out where you can improve. Through feedback from others, but also by looking for new knowledge and experience, you find out what else you can learn. In Part 3, I will discuss where and how you can gain knowledge and experience. But we also learn by [making mistakes](#). Preferably in a [safe environment](#). We learn from feedback and evaluations. In the agile world, [retrospectives](#) are common and often used. In a retrospective the team identifies what went well and what can be improved.

Coaching

For juniors coaching is essential. But not only for newcomers, for anyone who wants to learn, who wants to develop, a coach has added value. Antony Marcano wrote a [nice article](#) in which he says: “One thing that I notice is that while the teams are being coached, they do amazing things. They are more happy, more productive, fast to improve as if there are no limits to what they can achieve”. In many organizations, I notice that coaching is not often used. Here Marcano says: “So, if you have a professional software team without a coach, consider, are you really helping your business save money by going it alone? Or, like the professional sports team, is having a professional development team without a coach another example of a false economy.”

I want to conclude with a [tweet](#) from Michael Bolton: “Great sports teams treat practice with the same seriousness as a game — and every game as practice and learning. Testers take note.”

Huib Schoots sees himself as a context-driven tester. Currently he works as team manager testing at Rabobank International and is board member of TestNet. He is a member of DEWT (Dutch Workshop on Exploratory Testing), student in the Miagi-Do School of Software Testing and maintains a blog on <http://www.magnifiant.com/>