

# Basic training for software testers must change

Posted by: *Huib Schoots*

**Date:** Jun 18, 2012

**Category:** Columns



On my blog I recently wrote about my meeting with James Bach with the provocative title: [“What they teach us in TMap Class and why it is wrong”](#). Mid July I go to San Jose for the [CAST conference](#). During the weekend preceding I participate in [Test Coach Camp](#). The title of the post is the title of a proposal that I submitted to discuss at Test Coach Camp.

In the past I have been a trainer for quite a few ISTQB and TMAP courses. The groups attending the training were often a mix of inexperienced and experienced testers. The courses cover topics like: the reason for testing, what is testing, the (fundamental) processes, the products that testers create, test levels, test techniques, etc. In these three-day courses all exercises are done on paper. Throughout the whole training not once actual software is tested! I wonder if courses for developers exist where no single line of code is written.

In San Jose at Test Coach Camp I want to discuss the approach of these courses with my peers. How can we improve them? I feel these courses are not designed to prepare testers to test well. Let alone to encourage testers to become excellent in their craft.

During my dinner with James, I asked him what he would do if he would train novices to become good testers. He replied that he would let them test some software from the start. He would certainly not start with lectures on processes, test definitions and vocabulary. During a session the student will (unknowingly) use several techniques that will be named and can be further explained when stumbled upon. A beautiful exploratory approach I would like to try myself: learning by doing! But there are many more opportunities to improve testing courses. People learn by making mistakes, by trying new things. Testing is much more about skills than about knowledge. Imagine a carpenter doing a basic training. His training will mainly consist of exercises! My neighbor is doing a course to become furniture maker. She is learning the craft by many hours of practice creating work pieces. Practice is the biggest part of her training!

One of the comments on my blog opposed to the suggestion by James Bach. [Peter says](#): I have been both a tester and trainer in ISTQB and TMap. Yes we can make testing fun but without a method that testing has no structure and more importantly has no measurable completion. How will those new people on “more practical” course know when they have finished? What tests did they do? What did they forget? What defect types did they target? Which ones did they not look for? What is the risk to the system? My view after 40 years as a developer and tester is that this idea might be fun but is not just WRONG but so dangerously wrong that I am sad that no one else has seen it.

What do you think?

---

Huib Schoots sees himself as a context-driven tester. Currently he works as team manager testing at Rabobank International and he 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 [www.magnifiant.com](http://www.magnifiant.com)