Agile Record

The Magazine for Agile Developers and Agile Testers

April 2010

issue 2

www.agilerecord.com free digital version made in Germany

Meaningful Certification?

by Markus Gaertner

I hereby confess, I'm guilty of being certified. This may seem hard, but it's a relict from my life before software. Having spent multiple years on swimming sports, - a bit more than a decade - I finished my trainer's license certification back in 1999. Before that, back in 1996, I also made my certification to become a referee helper on swimming competitions. This article shows the lessons I learned while getting certified as trainer and as referee helper, and how all this relates to most certifications in place in the fields of software, development, testing, and Agile methodologies. Finally, I will propose a meaningful way for certifications, and how to view them in the larger software culture.

The problem

The problem arises from the common body of knowledge. A while back, Anne-Marie Charrett wrote a blog entry about her dream of software testing[1] . Instead of arguing about the best way to test software, she pleaded to reach common ground and understanding. Instead of arguing about the best way to do software testing, the professionals should rather sit together to reach an agreed-upon common body of knowledge. Software testers argue that the time spent on a project not testing, i.e. documenting, test reporting, mission clarification, is all wasted time. Paradoxically, we keep on wasting our time arguing about how to test software.

So, the core problem of certifications lies in the agreement on what constitutes knowledge. Certification programs mention one or more of the following: theory and definitions (including history), techniques (principles and models) and tools. Among the several certification programs, each addresses another body of knowledge, which may be a combination picked from this list. And of course, all of them claim that they are based on the knowledge that is the most meaningful. Unfortunately, none of them actually says something about skills for the real world.

Janet Gregory pointed this out in her blog entry "About Learning".[2] Certifications in testing, as they stand today, do not test skills nor solve problems. What they are targetted at is memorization of the body of knowledge assumed to be meaningful according to the certification vendor. Neither do the certifications help the testers translate that knowledge from memorization to application, nor do they help testers to transform what they learned in their particular work place. So, what we end up with are testers, that can prove to have known something at a specific time, i.e. when the certification exam needed to be filled out.

Recently, Ron Jeffries wrote on the plans of the Scrum Alliance for certifications on programmers and professionals.[3] Ron spent time on these new plans to certify Scrum practitioners. According to him, the plans on the Certified Scrum Professional include to be working in a Scrum environment as well as *"an acceptable report convincing people that you have a clue"*. Despite the classroom certificates common in the software industry, this supposedly will raise the bar on certifications. Though, this raised profile may be necessary, it does not automatically mean that it might be sufficient in itself, either.

Reflecting back, I remember one example from my past, where this "certification" model proved to be just paper. We had a father at our swimming club who was willing to help us out as a trainer. Since he had no past experience with swimming and training, he volunteered to take a trainer licensing course. Such a course consists of seven weekends spent learning about swimming and the underlying principles. When he finished his trainer license, he got in touch with us and asked how he could help us.

So, we invited him to our regular "management meeting", which we held once each month. During this evening we interviewed him, on his experience from the course, and on what he would like to do in our local swimming club. He told us that a good trainer needs to wfeature the good ones and get rid of the bad ones. Since this was such a dramatic gap between his viewpoints and ours, we argued the whole evening. In the end, we eventually lost that father and his two kids during that evening, but our underlying ethics were meaningful enough to let them go.

Certifications in sports

Taking a closer look on certification in sports, there are two basic types of certifications I came across: referees and trainers. The learning mainly divides up into theoretical principles and rules, practical exercises, and tests or exams. Last, but not least, there is a prolongation period to renew each license.

Principles and rules

For swimming referees there are clear rules which have to be followed. Faced with a difficult situation at the edge of the pool, the sapient referee judges properly based on the knowledge at the time according to those rules. The situation is the same for other sports, too. There are a thousand different viewpoints in front of the television in major sports games, yet the referee has to decide based on what he saw and what he learned before the game.

Similarly, a swimming trainer gets some body of knowledge during the educational courses. Yet, to come up with successful training, trainers need to adapt to the actual training situation and the particular student in the water. During the courses, the trainer learns mainly the underlying principles, like if the degree of lactate in the muscle is too high during training, then the student will get muscle ache the next day; or when this even might be the case. On this knowledge basis, the aspiring trainer learns techniques and practices to choose in order to achieve a certain goal of in training. In the end, the trainer is left alone after some tests and teaches people how they can master their sport.

The classroom material of the certification programs in place in software development build the set of rules and principles that your Agile methodology or testing course mainly focuses on. These ingredients are essential, since they include lessons on the overall picture. They help frame the discussion to a particular context. Unfortunately, though principles and rules are essential, they are not sufficient, as they may leave out relevant details which appear in the real world.

Practical exercises

In the swimming courses, we went through difficult exercises using practical learning parts. We got to know exercises by doing them ourselves in the water. We ate our own food, thereby getting to know whether or not an eight-year old may or may not have the power and the stamina for a particular exercise. Additionally, we were taught, that any new training exercise should be tried by the trainer himself before the kids do them. Thereby, the trainer knows how hard or easy the exercise in question is. During the referee courses, we also tackled practical evaluations. We practiced the business rules that we learned before-hand with direct exercises. These exercises referred directly to practical lessons from actual situations at swimming competitions. Therefore, they were meaningful and authentic.

Similarly, the course material from good certification programs today is filled with practical exercises on how to apply software test design, how to lead a daily Scrum or how the product backlog gets managed. By going through artificial software development projects, or dealing with a simple and easy to learn software, people are guided through their transition process to transform the theoretical knowledge into practical experience. The exercises take the application out of the particular context, and into a safe environment, where making mistakes does not blow up the next power plant (hopefully).

For the learning of the techniques these practical exercises are necessary to cross the gap between the right and the left brain, thereby transitioning the new knowledge into the long-term memory. In addition, for swimming courses there were some group exercises, where each individual group was asked to create a training program for the other participants of the course. After leading through the session, these groups got direct feedback from the other aspiring trainers. Also, group exercises resulted in an exchange of experiences in other clubs. This was a vital part of the knowledge that I was able to take home with me. A meaningful certification course has to include these exercises for collaboration, since the professional exchange is essential - for both sports and software development. Do you remember the most significant part of your last conference visit? Most likely this was the expert exchange between the sessions. The same can hold for certification programs.

Tests and exams

For the trainer license we went through multiple tests after conducting the theoretical and practical parts of the course. First, we needed to fill out a multiple choice test. The test also included questions which asked for fully written answers. After that I had to plan and exercise a training session with a given goal. At the beginning of the practical test, I got a tiny piece of paper with a goal for a training session. Then I had ten minutes to prepare that session. In another twenty minutes, I lead other participants through the training program I had just prepared. After the session I received direct feedback on my performance from both, the course leaders as well as the community of the aspiring swimming trainers who participated. If one of the aspirants failed either or both of the tests, he/she was asked to conduct a verbal interrogation afterwards with the course leads.

Similarly, new referees have an evaluation period during which they are escorted to real swimming competitions. Each new referee and referee helper thereby gets practical lessons at a swimming event. The new aspirants get an experienced referee or helper at their side, learning directly hands on from the more experienced colleagues. Open questions that may arise and that were not raised during the theoretical courses get clarified directly. Additionally, the new referee receives direct feedback on his actions and learns how to steer difficult decisions.

For software certifications we have just theoretical tests. You need to fill out some piece of paper with the knowledge you might have gained from the course itself or by self education from books, online material, or the course material itself. The theoretical tests simply exercise your memorization skills of that knowledge. Thus far, there are no exams showing that the course participants are

capable of transforming the theoretical knowledge into practice. A test manager who can manage people on paper may not be able to lead real people. Similarly, a ScrumMaster may know the terms "iteration" and "product backlog", but this does not say anything about his ability to remove impediments.

In contrast to this, practical tests are in wide use during job interviews. New testers get a simple application. In order to apply for the job position, they need to show their talent and their skills at learning a new program and showing relevant experience on their job. The same holds for Agile positions. Beyond certifications that the interviewees might already have, these practical exercises show the interviewer whether the applicant is suitable for the open position or not.

Prolongation

For my swimming licenses there is a fixed prolongation period during which I can renew them. For example, the trainer license can be prolonged within four years by attending a course over a single weekend. After four years, it expires and the renewal of the license requires more effort. This renewal process ensures that I still show interest as well as relevant practice in the field. Once a license was earned by spending the several weekends on the courses, it states something meaningful. But if you don't put this knowledge into practice over a longer period of time, it may diminish as you forget some of the lectures that you initially learned.

This also holds for software certifications. Interestingly there are just few which call out for a prolongation. For referees there are yearly changes to the rules as the federations find out new things, e.g. on swimming clothes. During the prolongation session, participants are also informed about these changes. Interestingly, considering the flux of the software world, there is no comparable effort put into keeping certified software team members up to date on the latest changes in Scrum, XP, or even software testing.

Meaningful certification

A meaningful certification program should provide the theoretical principles and so far they all do. But it shouldn't be the only thing to take away from a certification course. Besides a theoretical test, a practical test provides the teacher with meaningful insights whether the participant learned the lessons and can start taking actions from the material directly or not. When failing one or both, a follow up verbal test may reveal further insights on whether or not to grant the certificate. Last, but not least, failing the course after having failed all three tests is a consequential outcome.

As shown from the sports analogy, a meaningful certification course would include theoretical knowledge put directly into practical exercises. This includes running simulations of a project in the classroom, artificial training exercises on simple programs, which are easy to learn, as well as mini projects with mini iterations. In order to show that each participant has the ability to lead a Scrum or a test team, a written test alone is necessary, but not sufficient. Showing practical relevance to the teacher proves meaningfully that the knowledge has found its way into your brain and that you can apply it at your work. In addition, escorting new or unsure participants during their first days back at work also helps to foster the knowledge and put it in practice in the right context. Last, but not least, a prolongation course may transfer meaningful insights from the last few years accompanied with the professional exchange from other participants and practioners.

Finally, the corporate culture needs to get aware of the flaws of certifications. Certification alone means nothing. As the example from the misled father showed, you need to interview your certified tester or ScrumMaster and watch out if they fit into your working place. If they don't, either intensify their training or get rid of them. As shown in the example, this heavily relies on the support from the surrounding corporate culture. If there is no support, this may be the overall problem as Robert Sutton points out in his book *"The No Asshole Rule"*.[4] Maybe we need to create a software development culture, which is vitally suspect about certifications about certifications alone. Certification expresses that you may know something, but it does not express what you have made out of it and whether this fits into your workplace. ■

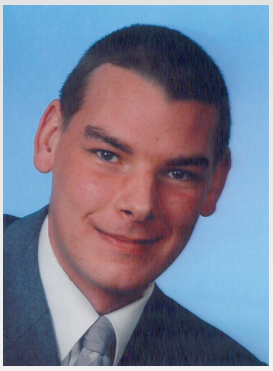
[1] Anne Marie Charrett, I have a dream, a software testing dream http://maverictester.com/i have a dreama software testing dream

[2] Janet Gregory, About learning http://janetgregory.blogspot.com/2010/02/about learning.html

[3] Ron Jeffries, Developer Quality! ... and Certification? http://xprogramming.com/xpmag/developer quality and certification/

[4] Robert I. Sutton, The No Asshole Rule Little, Brown Book Group, 2007

> About the author



Markus Gaertner

is a software tester for Orga Systems in Paderborn, Germany. Personally committed to Agile methods, he believes in continuous improvement in software testing through skills. Markus co-founded the European Weekend Testing chapter, blogs at http://blog.shino.de and is a black-belt in the Miagi-Do school of software testing.