

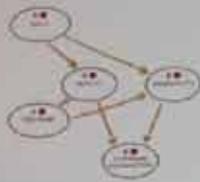


Agile Record

The Magazine for Agile Developers and Agile Testers



Complexity Thinking



View #1: Emergent Process
...ion, desires, understanding, relatedness

self-organizing teams, delegation

Management 3.0

Testing challenges for distributed team

February 2012

issue 9

www.agilerecord.com

free digital version

made in Germany

ISSN 2191-1320



An agile testing future

by Huib Schoots

In November 2011, I attended the Agile Testing Days in Potsdam. I had a great week, met a lot of interesting people and was very much inspired. However, I noticed something: only two of the nine keynotes were about testing. Not that they weren't interesting. They certainly were! But it's remarkable. Is this an evolution? Is testing not important in agile? I think it is at least underexposed. What does the future bring us?

The future of software testing

At the moment I'm writing a book with a group of enthusiastic testing colleagues for the Dutch testing association TestNet (*). The anniversary book by the jubilee book committee (JuBoCo) is about the future of software testing. There is nothing more difficult than predicting the future, and we don't have the illusion that we can. Yet writing a book with seven colleagues is a great experience: discussions with each other, investigating topics, the new insights, but also developing my own vision of the future. What will you do as a tester in the future? How do you prepare yourself for the future? What choices will you make?

For the tester it is slowly becoming a balancing act: does he opt for generalization by knowing a bit of all aspects, or does he choose to specialize and know everything there is to know of a certain aspect? And if a tester is going to specialize, which specialization to pick? Which choice should a tester make to be of value in about five years or later?

In our book we choose to use different personas to look into the future. A persona, which is a characterization of a particular type of user, is commonly used in IT. We use the personas in our book to describe various scenarios: they all make a different choice to prepare for the future.

André believes that testing will end up in the business: "I want to become a business analyst or product manager, so I'll learn all about how users think and work".

Dennis believes that testing will immerse in development: "All the innovations come from the development: ATTD, TDD, SCRUM are all development techniques".

Ilona says that testers are going to specialize in non-functional aspects such as security, performance, etc. "As the demand for testers is waning, I am still asked because I know more than others about these topics."

Yasmine thinks that if you are really good, you always have work: "I'm a jedi-tester and find errors earlier and better than anyone else. Scripted, Exploratory, technical or user level. I quickly understand each system and advise on the value of the system."

These four scenarios could become reality. Or maybe they already have become reality in some places? I like the fourth scenario best.

"Old school" testing?

Look at projects we do in a traditional setting with characteristics like Waterfall, Prince2 and the V-model. I almost wanted to say in former times, but in many cases we still do it. A designer consults the users and makes his design on paper. A programmer builds the system based on his interpretation of the design in a development environment. When the tester starts, the various components have to communicate with the infrastructure and with other systems. Often it is the first time the whole system is integrated. The tester is a spider in the web. When he does well, he communicates with all parties: technical experts, designers, developers, business analysts, environment coordinators, project managers, users, etc. But he still tests as much as possible himself, and testing is a separate phase.

Projects are becoming increasingly agile and use more and more agile elements: short iterations, stand-ups, whiteboards, user stories, burn-down charts, work items, etc. The tester's skills (the learned capacity to carry out pre-determined results often with the minimum outlay of time and energy) will have to change with this. Collaboration becomes more important and thus we see that the tester is becoming more assertive and pro-active.

Different role

The tester gets a different role. In agile projects everyone is responsible for the quality of the delivered products. And in the future, I think development will be even more in that direction. A tester is not alone in this, after all it's the whole team that tests. Testers will get a coaching role and will coach their teammates. They will teach and distribute their expertise over the team. A tester must be able to work with everyone and know that software development is a team sport. Software development is done with a team, and the tester should therefore actively look for ways to collaborate with the team. The tester knows that he is even more effective when he works together with developers

for example. Developers are complementary and can help him test faster and smarter. In agile projects test automation is very important. And again: cooperation between testing and development is essential to make this work.

You cannot hide

“In former times”, the tester was finding errors in the testing phase when the software was ready. “In the future”, testers will get involved much earlier. Agile means building a good product from the start. This will change the testing craft. In the past we saw a strong focus on finding errors and a procedural way of testing. Under the influence of agile and context-driven trends testing is becoming increasingly exploratory. This is necessary because of the way and the speed in which we do projects: iterative with short feedback loops. The quantity, but also the less detailed level of documentation available in advance, calls for different approaches. I think exploratory testing is an important one. This requires different skills of a tester.

To cope with the ever-growing complexity and ever-changing world around us, the requirements of a tester are getting higher (or at least: they should be). It requires a higher level for the skills that make a good tester: critical thinking, questioning, but also mastering testing skills, for example, applying testing techniques or doing (risk) analysis. In the past testers could get away with not having good testing skills. Because of the mainly confirmative way of testing, coverage of requirements was important. If the report was fine and all requirements were covered, testing was okay. However, finding the major bugs in a short time is a different story. In preparation for test execution, writing the test cases, the tester formerly had plenty of time and the opportunity to hide. In future, expectations will go up: with limited preparation he must show his testing skills applying testing techniques while he tests (exploratory). In projects testers collaborate increasingly closely with their teammates. The tester will quickly fail and not be taken seriously if he can't show his skills.

Pairwise learning

Back to the Agile Testing Days. I spent some time in the test lab doing some exploratory testing paired with a developer. It was an interesting session: he wanted me to show him how a tester does his job. And I learned how a developer looks at my way of testing by asking him questions and watch his reactions. He was nicely complementary on some points and some critical questions helped me generate new ideas. A nice learning session for both of us, and this can be done in projects as well!

Let's learn!

So is testing important in agile? Yes, it is! Things will change for testers, that much is certain. “Agile teams really do need testers – or at least people who have strong testing skills” [Hendrickson, 2008]. And there is still plenty to learn! Can we therefore agree that at the Agile Testing Days 2012 there will be more about testing? I love to discuss the issues of testing in agile projects. Which issues do we see? What will the future look like? Where can we improve? How can we learn from each other?

(*) TestNet is a Dutch network of, by and for testers: TestNet offers its members the opportunity to meet other testers and share knowledge and experience. TestNet organizes regular monthly evenings and has approx. ten special interest groups like agile testing, usability, privacy and model based testing. TestNet also organizes two major events every year and publishes a newsletter. Next year TestNet celebrates its 15th anniversary. The initiative for establishing TestNet was taken late 1996. The reason was the growing interest in structured testing. TestNet was formally established on May 15th, 1997 and currently has about 1,600 professional testers as members.

[Hendrickson, 2008]

Agile Testing, nine principles and six concrete practices for testing on Agile teams – Elisabeth Hendrickson – <http://testobsessed.com/wp-content/uploads/2011/04/AgileTestingOverview.pdf>

> About the author



Huib Schoots

has 15 years experience in IT and software testing. After studying Business Informatics he became a developer. Soon he discovered that development was not his cup of tea and software testing is fun. Huib has experience in various roles such as tester, test coordinator,

test manager, trainer, coach, but also in project management. He is currently Team Manager Testing at Rabobank International. He tries to share his passion for testing with others through coaching, training and giving presentations on different test subjects.

Huib sees himself as a context-driven tester. He is curious, passionate and has (unsuccessfully) attempted to read everything published on software testing ever written. He is a board member of TestNet, the association of testers in the Netherlands. He is a member of DEWT (Dutch Exploratory Workshop on Testing), student at the Miagi-Do School of Software Testing and maintains a blog on magnifiant.com