By Erik Boelen
Test Consultant
QA Consult Services

Erik.boelen@qaconsult.eu
www.qaconsult.eu

'I am an Agile Tester!'

'I am an agile tester!' These are very trendy words nowadays! Most of the times they are accompanied by slogans like 'we don't use documentation', or 'We work very chaotic' and one of my favorites 'We scrum at least 2 times a day, every single day'. To 'scrum' is not a real verb, by the way (W*ictionary.org*). We perform scrums, would be a better sentence. That is, when ignoring the real meaning of Scrum in agile projects of course.
Don't get me wrong, I do believe in all the good intentions that are linked to these words. I've seen myself that these approaches have a very positive influence on final project results. I am a true believer, as they say. I just don't agree to the fact that using these words makes you an agile tester – let alone a tester in general. A good tester – agile or not - can use and apply all these principles, but using them does not make you a good tester by definition. We have to prevent testers and their customers from losing the actual meaning of agile, since testing in agile environments is not about terms and definitions.

A quick refresh on Agile

As a quick reminder to the reader on what exactly is agile in software delivery, I refer to a presentation given by Elisabeth Hendrickson at GoogleTalk on Agile Testing.

> ***Agile is providing a continuous stream of value to your customers. The two most important aspects of Agile are:***
> - ***Increase the rate of feedback***
> - ***Reduce the waste***
>
> ***A good way to describe agile is by saying what it is not. So, agile is not about:***
> - ***Compressing the schedule, nor***
> - ***Tossing out the documentation, and certainly not***
> - ***Coding up to the last minute.***
>
> ***It is all about maximum value for minimal cost.***

Elisabeth Hendrickson on Agile Testing

It all began with the Agile Manifesto for Software Development. If you read the principles of this manifesto closely, you will see that the previous statements are clearly derived from them.

> ***Individuals and interactions*** *over processes and tools*
> ***Working software*** *over comprehensive documentation*
> ***Customer collaboration*** *over contract negotiation*
> ***Responding to change*** *over following a plan*

Agile Manifesto for Software Development

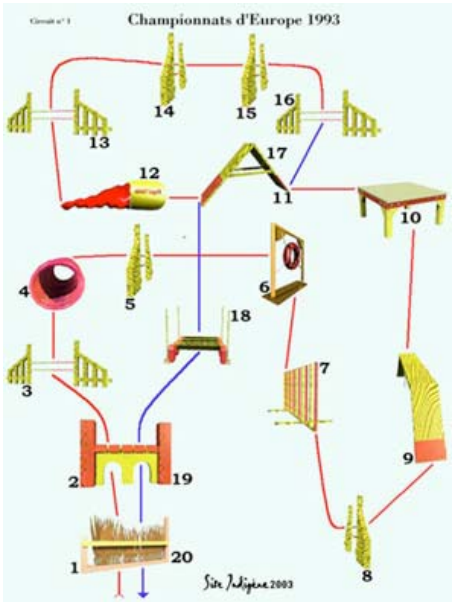
Agility comes closest to what I think is working in an Agile environment

A dog's life

The last few months, I've noticed that companies started an actual search for agile testers. So, there must be a definition of agile testers, a job description. That part in particular made me curious. When are you an agile tester? Is there something that you need to have in order to be called an agile tester? Which human aspects are important to fit in the process of agile testing?

Intrigued by these questions, I've done some research on the matter. And I found out that there is a sport named Agility that comes closest to what I call agile testing. Yes indeed, Agility as in the sport for dogs in which they run a full course of different obstacles in the most efficient manner possible. In order to do so, these dogs must be quick-moving and nimble (*wikipedia.org*). Especially the word 'nimble' in the definition showed the resemblance. *Encarta dictionary* says the word nimble means 'Fast, agile and light in movement' and 'Able to think quickly and cleverly'. Does this mean that agile testers can be compared to dogs?

Well, being a tester, it can be a dog's life sometimes.



Picture 1: Agility Course

Why a checklist?

Let's get back to the goal of this paper which is providing a checklist to assess the agile testers. I've had several reasons for setting up this checklist. The first reason was to see for myself if I could determine what an agile tester should be like as a person. Along came the second reason; since I believe that testers have that certain 'state of mind', I'm always looking for the person behind those skills. During a candidate interview, it's easy to talk about the technical skills and the previous experiences. But, what is the person I am talking to like? Is this person cut out for the job? And that's when I started using the checklist when talking to potential testers in agile environments. And finally the third reason is to provide some sort of self assessment to agile testers, providing an insight on the (personal) skills required and their actual score on this list of skills.

So, to me, this checklist has become a working instrument, a guideline towards becoming a better agile tester, not focusing on techniques and methodologies but on the human aspects.

The Checklist

Two disciplines are strictly required, being very tight version control and detailed planning

I love my customer

The customer – both internal and external to the company - plays a central part in agile projects since that role is responsible for ensuring the right thing gets built by:

- Providing the vision that drives the product.
- Steering the project through numerous specific decisions about the sequence of work based on an understanding of the overall business benefit relative to the cost of each piece of work. This is an ongoing process that accounts for changes to the business over time.
- Describing the requirements clearly and succinctly, at just the right time, and in just the right level of detail.

The customer role is also responsible for assessing that the thing was built right by:

- Defining acceptance criteria / tests that drive product development
- Testing the delivered product.

(The Customer Role in Agile Projects - Workshop Position Paper by Jennitta Andrea)

As you can see, it's better to love your customer than to cherish hostile feelings. Therefore, it helps to feel connected to your test object and the environment that you work in. When you don't support the solution provided by that customer and feel confident about the requirements, motivation and determination can be hard to come up with. Agile projects stress the fact that everybody is a part of the chain that delivers the product. So, being uninspired can make you the weakest link, slowing down the entire project.

I work in a very structured way

Agile projects are generally called chaotic, hectic and ever changing. I will not deny that this is indeed the fact in most cases. But, my point is that within that chaos, it requires a lot of structure to keep you organized and up to speed with the progress of the project. In agile environments, two disciplines are strictly required, being very tight version control and detailed planning. Without those, the risk of turning into chaos is clearly present.

A former colleague of mine had the following opinion on this topic: *'for me agile testing is an extreme way of structured testing. It follows very closely well defined small development packages that need to be tested. Development packages need to be defined with a short delivery term (depending on the size of the project, but in a range of 1 to 2 weeks) within the development- & test department to have a qualitative product at the end of each iteration. Very often, the customer defines the development packages together with the supplier to have influence on the most urgent functionalities for their market. This can result in a phased payment to the supplier. Daily stand-up meetings are frequently used to have an accurate follow up. For me, it is the way to handle a very large project in small deliverables and have a faster return on investment. For me, it is not a way to handle constant uncontrolled changes from development. That would be uncontrolled testing...'* (Anneliese van Egghen - CTG).

Agile projects prefer providing solutions to customers, rather than writing documents on the solutions

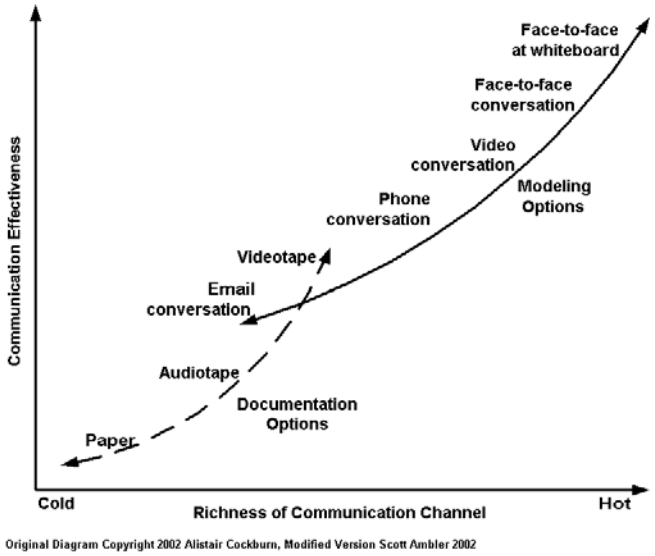
I adore unconventional test tools

A few years ago, I read an article from James Bach, 'Boost your testing super powers'. The most significant part of this article was about him using a transistor radio for testing purposes. This was to check if 'a program that appears to be hung is really stuck or just deep in thought'.
Of course it doesn't always have to be this extreme. Other tools can be used for assisting your testing activities, being CSV editors, screen cams, monitoring tools and mind mapping tools. These are all tools that are not meant for software testing, in first instance.
And the best part is that a lot of these tools can be found as freeware on the internet. The agile environment provides testers the opportunity to propose using these tools, if they prove to be valuable for the end result of the testing activities.
Besides from the already existing tools, a lot of programs are written on the spot, in order to improve the testing activities. Programming languages like Ruby and Python are working their way up in this branch of software development.

I communicate from an open position

Communication is crucial in testing, and particularly in agile testing. As already mentioned before, agile projects prefer providing solutions to customers, rather than writing documents on the solutions. Therefore, project members need to talk to each other in order to send and receive the information needed to do their job. Both verbal as written communication are important.

Alistair Cockburn has setup a graph on which he shows all forms of communication, by plotting the richness of the communication channel against the communication effectiveness in agile environments. Notice the uncommon possibilities for documentation options like audiotape and videotape.



Picture 2: Communication in agile projects (Alistair Cockburn)

So, where does the open position come into play? In an agile project, just like in any other, a distinction is made between the roles of analyst, developer and tester. When communicating with the other roles, you should be open to them and not just your own. The communication is an interaction between the different roles. Developers are involved in the tester's area of expertise; analysts are involved in development items and vice versa. No need to say that this requires a high level of honesty from all parties.

- *I am a VIP in my project team, just like the others*

As mentioned in the intro of this paper, the Agile Manifesto holds as one of their principles 'Individuals and interactions over processes and tools'. Elisabeth Hendrickson also talks about it in her presentation: 'I felt appreciated as a tester! Wow! That was new to me!'
In agile environments, the contribution of each person to the end result is highly appreciated. The agile team consists of people, all with their individual capacities, combined together through interaction. A good agile project is actually getting the best out of every individual person, in order to serve the greater cause.
Therefore, the reward and recognition is to be done in an equal, yet individual way independent on the role played in the project. If you keep the item on communicating from an open position in mind, it could also become difficult to just evaluate testers on their testing skills since they are also involved in the analysis and development of the project.

- *I don't spend time 'covering my ass'*

This is where the documentation comes to play. The agile manifesto holds the statement 'Working software over comprehensive documentation'. By asking the question 'What are your main reasons for setting up documentation', in about 90% of the cases I get answers like 'Well, I do this to pass on information, to get agreements on the content; or maybe to build a database of permanent documentation'

We shouldn't spend time covering ourselves in times of need

Unfortunately, the main- yet unspoken - reason for setting up all that documentation is covering ourselves in times of need. Setting up documents and their review processes are useful, when being used correctly. I talked about reducing the waste in the beginning of this paper. Overhead in documentation is part of this waste. The test plan is a good example. In most cases the test plan holds preconditions and predetermined scope definitions. By rigidly defining these two items, the flexibility is immediately taken out of the test activities. They will always hold the possibility of referring to when things go wrong and you don't feel like taking the blame. Solutions to problems are provided by the whole team in agile environments. No scarce, useful time is spent on finding the guilty.
The collaboration between the different team members overrules the points of control built in between the different roles. Entry and exit criteria are another good example in this context. Without denying their purpose, they should not be abused in terms of 'It wasn't me' or 'I told you so'!

Of course, not all documentation should be thrown overboard. An example of documentation that is not considered 'waste' is problem reporting. A good documentation of issues and problems is crucial to the follow up in an agile environment.

- *I know my boundaries*

Testing in an agile environment stands a risk of becoming chaotic, unstructured and uncontrolled. It's keeping that borderline in sight that is a difficult management item in agile environments. The tester, as well as the other roles, enjoys freedom combined with a lot of responsibilities. Amongst others, the correct setting of the boundaries is definitely a risk in agile environments. And by boundaries I mean boundaries on a content level as well as on a timing level.
Agile environments are well known for their releases of small parts of the solution, also known as increments. The content of these increments is agreed upon before the start of each release. As a tester it is crucial to understand and respect these increment requirements. Based on these requirements the test plan and the test preparation need to be set up. Crossing these boundaries would cause overhead in analyzing and deferring possible issues found. Determining what is out of scope is as important as what is in scope of the testing activities.
Together with the content boundaries, a planning needs to be set up for every increment on the testing activities. Time boxing is a widely used technique in planning activities on agile projects. During test execution it is important to keep the time schedule and indicating possible risks if the test execution is not finished in time. A decision can be made by the project team to extend the testing period, or go along with the risks. In order to make this decision possible, the tester always needs to work according to schedule.

And finally, the last item

The normal way to conclude this paper would be by emphasizing on the differences between the agile testers and the more traditional tester. But, the careful reader has probably already noticed that there aren't that many. Of course, I cannot deny that testing in an agile environment does require some personal skills for a tester that are different or more refined that testing in a conventional environment. Examples are communication skills, openness to others and flexibility. These are also characteristics that wouldn't hurt traditional testers in the execution of their jobs.

So, I inserted an additional point to the checklist on being a tester.

- *I am a tester*

This is definitely my favorite item of the list. Above all, for being an agile tester, you need to be a convinced tester to start with. Testing is a mind set; testing techniques are in place to refine or brush up the testing talent and create more structure in the testing. By defining agile testing as 'testing in an agile environment' it becomes clear that an agile tester is a tester that adapts to certain ways of working. Also, at some point in time, specific techniques of testing in agile environments like exploratory testing come naturally to a tester. If I look back at my career, I was already practicing agile testing techniques before I ever heard of them. This brings me back to the intro of this paper. It's not by using the definitions you automatically become a tester, let alone an agile tester.

Agile testing is testing in an agile environment

Sources

- Article James Bach – boosting your testing superpowers. (http://www.stickyminds.com/sitewide.asp?Function=edetail&ObjectType=COL&ObjectId=3033&tth=DYN&tt=siteemail&iDyn=2)
- Anko Tijman – Agile Testing (Testen als teamsport)
- Wictionary.org
- Wikipedia.com
- LinkedIn.com
- Google Video – Googletalk – Elisabeth Hendrickson ((QualityTree.com) - http://video.google.com/videoplay?docid=-3054974855576235846
- The Customer Role in Agile Projects - Workshop Position Paper by Jennitta Andrea
- The Agile Manifesto for software development - http://agilemanifesto.org/

More information about QA Consult Services is available on the web at www.qaconsult.eu