Converting traditional teams into x-functional teams based on industry standards @ Austrian Lotteries

Frank Bieser / Austrian Lotteries
Alexander Weichselberger / SEQIS Software Testing

Presentation @ AgileTestingDays.com / October 2013

SEQIS Software Testing GmbH, Neusiedler Straße 36, 2340 Mödling, Austria
Tel / Phone: +43 2236 / 320 320 ext 0 - Fax ext 350, http://www.SEQIS.com
Firmenbuchnummer / commercial registry number: 204918a, UID / VAT: ATU51140607
Facebook/SoftwareTestIsCool twitter@SWTestIsCool

About the Authors:

Frank Bieser started as a developer kid in 1983 and has been in touch with software engineering ever since. He holds Masters degrees in both Computer Science and Economics. During his professional carreer, Frank has been an entrepreneur, worked as management consultant, and succeeded in various IT management roles including C-level functions. His industrial background includes automotive, public transport, steel, internet search engines as well as gaming and gambling. Currently, Frank Bieser is Head of Applications Development at Austrian Lotteries and Casinos Austria, where he implements Agile Leadership.

Alexander "weixi" Weichselberger is a proud software tester and member of the SEQIS management board.

His IT background for SW test are Oracle/DB 2 administration in Windows/Unix/OS390 operating systems and IT architecture. After a number of smaller software testing projects, he focused on non-functional criteria testing like high-availability architectures and load and performance tests. After that he took over leading roles as test manager in exposed large-scale projects. Besides this his profound theoretical knowledge (ISTQB, CAT, PM, Scrum), his degree in European management and business management as well as his hands-on experience as member of the board helps in difficult management-oriented aspects.

"I'm sure that knowledge and experiences just together with respectful but also goal-orientated interaction and cooperation with colleagues and partners will fully enhance potentials. Analytical capabilities, flexibility and my engagement to realize necessary conditions are additional personal strengths."

Weixi is president of the Austrian Oracle Users Group (www.AOUG.at) and is in charge for experts meetings of the Austrian Testing Board (www.austriantestingboard.at).

Frank Bieser
Alexander Weichselberger

Let me tell you my favorite User Story:

As a development manager,
I want zero escalation meetings,
so that I can develop a vision
and help my teams grow.

Unfortunately, this story was more a fairy tale by the time I took over the role of Head of Application Development. Customer satisfaction was at an all-time low.

DELIVERY: LATE AND INCOMPLETE

QUALITY: BELOW EXPECTATIONS

COMMUNICATION: LOUSY

CUSTOMER: UNHAPPY

As software delivery was usually late, incomplete and did not meet the expected level of quality, we had quite frequent escalation meetings.

Communication inside the development organization and across towers was poor. It was channeled through a guild of people who were called IT coordinators and acted as SPOC, so all issues went across their desk. Cooperation existed only in those areas where it could not be avoided.

Since in our industry we have tighter restrictions than the finance sector, we must clearly differentiate between the people who write the code and those who test it. ATD is probably the only place in the world where this does not sound awkward... We understand that this comes with a price tag, and we are ready, willing, and able to fund this segregation of duties in order to comply with our regulations.

However, the existing organization was not fit for purpose: silo-ish groups with no general view, operating in command & control mode, no end-to-end responsibility for anything, and no collective ownership. Miserable collaboration was literally the result of our organizational design. The CIO, myself and a bunch of other people (not only decision makers) were absolutely positive that Agile software development would bring us back on track.

So we had to change the game.

To make things even worse, the Head of QA position (where the software testers belong) was vacant at that time. Since I know that software quality was crucial to win back our customers' trust, I had to quickly close this gap. Therefore I asked this gentleman to join me on the field as an interim manager in charge of the testers.

Frank, thanks for this great change over!

Before I dig into this project deeper pls let me briefly introduce myself: I'm not as Frank employee of Austrian Lotteries. I'm a software tester focusing on large scale projects especially in challenging situations. So I don't just tell people what to do - normally I stand in the mud with the team & work on the solution.

So – back in February 2012 Frank called me to be the temporary line manager in QA, responsible for helping with the Scrum introduction.

During the next about 40 minutes Frank and I will talk about our challenges and results during the introduction of Scrum.

Frank already did briefly explain the "why" we had to change and what we did introduce. I will also add some QA specific findings at the starting point. After that we will explain todays situation and focus on the process of transformation and the challenges we met. A brief outlook to what comes next @ Austrian Lotteries will be the last point of our presentation.

How are we going to make the presentation as you can see us both at the stage?

As Frank is the leader of SW development @ Lotteries leading all departments like PO, SM, Development, QA, RM and Application support – he will focus on the department wide aspects. I, as leading QA, will focus on the purely QA orientated points.

So lets briefly wrap up the QA situation back in Feb last year.

As we had already 20 people in QA – why didn't we deliver the expected level of quality? Why would programmers look down to us testers? Besides some or even more uncertainties what comes next with Scrum, we in QA had following main issues:

1. New functionality was developed fast by programmers as they did create new functionality module based or just by changing the configuration. From a testing perspective even simple changes did result in time-consuming regression testing. Why? Well, we had a former target of "100% test automation" but historically implement as ice-cone and not as pyramid kind of TA.

2. Another basic problem was the missing of communication opportunities to learn from each other in QA: Mistakes in testing and program errors weren't communicated within QA, so everyone has to experience & solve issues by themselves over and over again.

3. As there was always overload in QA – you remember this fast approach of configuration-based programming – QA didn't fulfill basic documentation demands: "I

will do it later!" So we didn't really have an overview of our test cases, test execution and also no mapping with requirements.

To sum it up: Increasing technical (testing) debts from one sprint to the other, no real fundamental testing standards and a high pressure to go to market. And please don't forget that Scrum should be introduced and there was no QA leader. Quit a toxic mixture!

Before we now focus on the individual challenges we did face we briefly explain today's status. So, Frank, could you pls draw a picture of todays situation?

Where we are today

Today, we run 10 cross-functional Scrum teams with a total headcount of 85. The teams are clustered by platform such as Online Gaming, Lotteries, Casinos, or Sports Betting.

All teams are co-located in Vienna, so developers and testers are sitting next to each other. They have their Scrum Masters nearby and the Product Owners next door. The POs are IT staff, and yes, that works. We will talk about them in a minute.

We also practice distributed Scrum with two more Scrum teams who support our local platform teams from a remote location. We have established a Driving PO concept that ensures personal contact once a week.

Before all that could happen, I had to rebuild the organization.

A
PO
A
SM
A
DEV
A
TST
A

I rebuilt the organization according to the 4 (!) roles we needed: POs, SMs, Software Developers, and Testers – all assigned to their respective departments, every one of which is headed by a line manager with a robust expertise in this specific discipline.

As a Joker, we implemented a group called "Application Support" who act as a firewall. Their role is to protect the Scrum teams from production incidents as long as possible. In more than 80% of the cases, AS staff find workarounds, can correct data or otherwise troubleshoot without bothering the Scrum Teams. AS people participate in the Reviews, so they know what software changes are coming out of the long grass. They are extremely efficient and highly valued by the Scrum teams. If the incident turns out to be a problem, AS helps to further qualify the incident information, drill down towards to the root cause in order the utilize the Scrum teams in the most effective way.

At the beginning of our Agile transformation, I sent out a strong signal that we would no longer allow “programmers” to look down on “testers”. Both tribes of developers must work towards the same goals, collaborate intensively, and have interlinked and well-aligned responsibilities.

Another concept that turbo-charges your Agile Transformation is to deploy full-time, dedicated Scrum Master for each and every team. SMs tackle a lot of issues very early in the process and help the teams to get on the steep part of the learning curve.

However, management is having a hard time when Agile methods are being implemented. That is because they have to accept the concept of uncertainty. Fixed time, flexible scope doesn’t make the typical manager (project managers included) feel very comfortable.

At the same time, development teams are not really in favour of receiving fully blown requirements documents, which kill every spark of creativity.

The key to success is to establish a lively community of Product Owners. Together with the development teams, POs can literally change the game. In our case, we used to have “IT coordinators”, who were a mix of IT project managers, requirement engineers, communication heroes, and Subject Matter Experts – doing whatever was necessary to make the project successful.

In self-organized teams, two things are extremely important: 1. self and 2. organization. The last thing you want is a coordinator who tells people what to do. Good POs, however, make the difference, as long as they focus on the WHAT (and let the team figure out the HOW).

It is the PO’s task to develop a product vision, to groom the backlog and to demonstrate to the business stakeholders that together with their teams they are continuously delivering value for the organization. In our case, POs also translate between project milestones and sprint cycles, helping to align the different cadences.

A real challenge is to keep track of the big picture, while you are only working on small slices. We have also made good experiences with “Kick-Offs”. They may sound a bit folkloristic to you, but they give teams and individuals a great overview of the big picture, enriched with business perspectives and relevant background info. Our teams really appreciate those meetings before they dive into delivery of larger projects.

Wrong priorities, misinterpreted demands of the BU will result in a potential catastrophic failure, even if the Scrum team is doing good: Great landing - wrong airport...

And, POs are the only IT people who can waive tests. The team may suggest to skip the -say- performance test. The PO has to come to an agreement together with the customer and ultimately issue a GO/NO GO for the release. Quality is what the customer perceives.

The last escalation meeting related to poor performance of software delivery took place on June 11, 2012. Within six months we had taken this topic off the table.

So we rebuilt the organization, created cross-functional teams, adjusted some methods - was that it? No. It was not even the beginning. It was a pre-requisite - just housekeeping.

One very important tool to support the transformation was founded training and possibilities to learn as an organization.

So right at the start Austrian Lotteries did invest in CAT training for the complete QA team. By this it was ensured the in the "new world" no language barriers as well as no gaps in process know-how will exist. With CAT testers will be really familiar with the ideas of Scrum, but did get it with the specific testers point-of-view. During the CAT training the testers could not just "play" in the practice session – they could discuss with each other how they want to setup their cooperation and could already challenge their ideas with the trainer. So I recommend not to pick one tester from each team → better send the all testers from a particular team to CAT training.

PROUD TO SAY…

I mentioned previously that in the starting phase we had quite some issues to learn from each other in daily business. So we decided that the QA line manager, in those days I, was in charge for the interpretation of all the standards out there: WLA, a security orientated standard but also IEEE and ISO – QA had to set at least the common minimum standards.

So how did we address all this? We defined that the QA line manager has to present standards to testers and discuss with them in a kind of experts talk.

We also introduced so called tester academies to discuss aside from daily business standards in testing, methodology stack and tools. Or even basic techniques like equivalence classes, boundary values, path coverage and so on. By this we supported continuous learning and development of people – and we gave a platform to exchange know-how.

In addition we decided that from time to time the QA line manager should take part in some standups or planning sessions. As QA line manager in Scrum you has to offer your people to work independently in the teams. So – this cooperation will let you be within reach to every tester. And you, at the other side, could get an idea of your testers performance and also issues.

So – to wrap up the HR duties of a QA line manager in a Scrum environment: You are in charge for setting basic standards and you are in charge for developing your people. But you won't be their contact for private issues → this must be the Scrum team. So, pls ensure that you take care about this too and you will support and establish this as well.

Trainings and co were established in other departments as well → but we had to face other challenges. Frank?

It was a bumpy ride.

The gaming & gambling industry has a vast number of standards to comply with. We must protect our organization from fraud & collusion and money-laundering from every direction – inside and outside. One prominent example is the standard issued by the WLA. It extends and refines ISO 27001, and –amongst other things- it forces us to ensure a robust segregation of duties. This is the exact opposite of poly-skilled engineers working in cross-functional teams with a "just enough" level of documentation.

So we had to find ways to remain compliant with WLA and other standards without torpedoing Agile software development. I would like to share two simple aspects with you:

1) artificial barriers on a system level: No write-access for developers on test systems. Those systems belong to the testers and the testers only.

2) 4-eyes principle on automated test cases: could be a review, could be done pair-wise, but needs a second person to double-check. Once it is automated, chances are that it runs for a while.

Agile methods do work in strictly regulated environments.

One of the duties of a line manager is to set standards. But he also needs to translate standards into guidelines and principles that work in our ecosystem.

COLLECTIVE OWNERSHIP

A good practice is to bake standards into the teams DOR and DOD on story, sprint, and release level. Our User Story template for example contains a section for NFRs to ensure that security and performance aspects are being taken care of. By this, we also expose the POs when we talk about quality.

This is particularly important when you work with external providers. Invite them to join your dailies (could be remotely), if necessary impose your quality standards on them, give them access to your tools.

But don't even think that anybody in the business wants to hear that it is "their" fault when a delivery goes south. It's a team effort to collaborate effectively with whomever they need to be successful – not a management task (see also: self organization).

Documentation is really helpful – not only for the sake of compliance.

However, you have to find the right dosage. One rule is, that we do not "lose" known bugs along the sprint – they're either fixed or documented. When a bug is detected during a sprint, the tester talks directly to the developer who can fix it. If there is no developer available, the bugs must be recorded for later fixing. How exactly the teams do this documentation is up to them; some use sticky notes, others JIRA – both are OK.

For certain things we demand a second person to review the tests: mainly when a developer supports the testers or when business critical features are being secured with automated tests. Again, it is up to the team how they document it. Practical examples are either to generate a "review" task on your Scrum board or to add a one-line comment in either the test automation tool or in JIRA. We also archive the session sheets and sometimes take photos of the task boards and keep them with the release documentation.

I recently had a conversation with a very skilled and experienced tester, who was wondering whether she could still use her ISTQB knowledge in a Scrum team.
To me, this question sounds a bit like a JAVA programmer asking whether he is still supposed to write JAVA code. The answer is clearly "yes": Good testing skills are still (and even more) required in an Agile environment - even if the process may be different.

So I continuously encourage our testers to speak up. A good place to do so is the team, and a good time (not the only time, of course), is sprint-planning 2. SP2s go terribly wrong, when developers talk JAVA to each other, while testers starve for another meeting where they can discuss their QA strategy in absence of the JAVA folks. The place to sort things out is exactly there, so they better utilize the meeting formats effectively. And typically, as the team becomes more mature, both disciplines –programmers and testers- benefit from cross-pollination as they work together.

Like many other organizations, we have an enterprise-wide system of appraisal talks, which is known as MBO. However, I wanted to clearly strengthen the newly created entity called "team", so I needed to find a way how I could tweak our performance management system without starting a battle with HR.

We actually inserted team goals into the regular process, inviting every team to define their own goals (which should not be contradictive to overall IT goals though). But we also kept individual goals, because we needed them for the line managers to bring in their development actions for the personal growth of their employees. Both types of objectives were considered equally important.

So overall, we took a very sporty approach to our transformation.

In addition to giving the right standards and trainings we had to address the given testing approach. As explained @ the start especially regression testing resulted in an enormous effort prior to Scrum. But our view is that at the end of the sprint we deliver software, which can be installed and used – that means, sufficient testing goes hand in hand with a commitment.

One mechanism to tackle it is obviously to ensure a consistently high level of test automation. However, nobody says that this is the job of the testers only, just because it's called TEST automation. I strongly recommend leaving this first word "TEST" away and than you easily can add individual mappings like "developers automation" or " testers automation". On the lower end of the testing pyramid, programmers and testers can proof that they understood the concept of collective quality ownership. As long as this is not part of your culture, it may be helpful to state in the team's definition of done.

Ok – so we targeted the right TA, certainly based on the TA pyramid. We worked on introduction of CUT for programmers and used an API based framework as well as GUI automation for the tester.

For sure we did take care about basic standards like test cases should run independently from each other and, if appropriate, they will build their own data as well. In addition we focus on the main streets of the application especially when you go up the TA pyramid & avoid automating every aspect of the application.

And we always had an eye on ROI as well:
1. Take care about the coverage of developers automation and testers automation – ensure, that the overlapping is given but not too much

2. Ensure benefit from TA: Test automation doesn't end in itself – it has to have a benefit. So if it doesn't pay to execute particular TCs because they don't show any relevant results: Don't execute them, especially if they are at a higher level in the TA pyramid! We considered also deleting them: Why invest time to maintain them over and over again? So, if completely useless: Delete them!

So, how to realize test automation with the given initial debts? Our starting point here were masses of old code with more or less assured functionality.

So how did we approach here? We introduced following rules to address given technical TA debts:

a. If the old code isn't assured but is in production > take the risk and keep it outside the regression as it runs anyway
b. If you have to work on old code by current development → add TA of this old code to the DoD
c. When you develop new code make sure that you will do your TA duties

By this the TA coverage will increase naturally and we improved the situation without tremendous efforts on the starting point for doing things that probably will never pay off.

As mentioned before we addressed C.Q.O: So to boost tester automation to further reduce debts we spread task to developers as well. To respect given guidelines we set on 4 eyes principles for developing TA: A tester had to accept TC developed by programmers. And we introduce the strict rule, that that a programmer is not allowed develop testers automation that's assuring even parts of his own developed code.

Why this? Well on one hand side you will be blind for problems that you developed by yourself. Just think of all the exciting advantages of pairing that will proof this rule. On the other hand especially in the gaming and gambling industry its necessary that you avoid every chance that someone could influence the result of a particular draw. Simple think of the given impact of a developer could introduce a backdoor to online poker or so. This has to be avoided.

Certainly we had other debts to address too. TA is just an example. And we had to work on other transformations as well: Introduce new standard tools and performance engineering, as well as cooperation with operations on OS patches or donating people to work on the individual communities of practices.

So Frank – what are the plans for the future?

Where we want to go

A business-facing initiative is that we are currently working on the implementation of Agile Release Planning. We invite project managers as well as business planners to attend a series of workshops together with the POs. The aim is to generate a common understanding on how to use Story Maps and techniques such as Impact Mapping in order to get the maximum out of Agile software development.

Looking at IT, we initiated a workgroup to bring us all closer to a DEVOPS setup. This is particularly tricky because we also must comply with WLA standards on the edge of DEV and OPS, which means that segregation of duties is also required in this area. As a starting point, it may be interesting to consider job rotation as a concept to increase the level of automation along the entire deployment queue.

Last but not least the people perspective: Our leadership team will continue to put a strong focus on supporting personal growth for every team member and fostering a culture based on Agile values.

Well, in QA we will continue to reduce technical debts. And we work on developing even smarter ways how to create TC and their execution.

For further improvement of knowledge in QA Testing Dojos and a test architect might be introduce. We have to work on that…

But overall – ensure fun in testing and continuous development of the personal skills is in our prime scope.

And also have an eye open what come next in this great industry IT.

Pls evaluate my presentation and use for this the AgileTD Mobile.
I would appreciate your feedback. Thank you very much!