BEST PRACTICE
Agile Panel Discussion

Date: 5/2/08
Expert Panel: Quality Opportunities and Challenges with Agile

Panelists:
Burke Cox, Stelligent Incorporated
Bill Rinko-Gay, Technisource
Lois Zells, Lois Zells & Associates, Inc.

Co-moderators:
Bill Rinko-Gay, Technisource
Nancy Kastl, Kaslen Group, Inc.

The following questions were answered by Bill Rinko-Gay

Q: How do you best approach being more Agile in QA when SOX Controls demand compliance to deliverables that development are 'non-value added.'

There are many ways to add value to a program. The Development team is a key driver in Agile methodologies, but there are many project stakeholders that have to be satisfied. Legal requirements have to be met, and even the Development team would agree that satisfying legal requirements adds value.

The real question here is how to actually meet the SOX compliance requirements adding the least burden to all the teams involved, not just Development. The two parts of the answer are:

- Review your SOX deliverables and ensure each is required to verify the controls are in place. Many organizations over-engineer their SOX controls out of a desire to be conservative. This is because they are either written by the legal team without a real understanding of IT, or the IT department without a real understanding of the law. Let a good partnership verify your SOX controls (and other compliance measures) aren't more burdensome than the law requires.
- Once this is complete, use process management techniques to verify the documentation is being developed in the most efficient manner possible. Look at key areas such as: verify the documentation produced at the correct phase to minimize thrash, have the correct owner to ensure the documentation contains all the required information, establish an

appropriate level of review to avoid delays. Also, verify the documentation is not bloated. Review the sections and materials to ensure everything is necessary and sufficient.

Having accomplished this, everyone involved in the documentation process should be satisfied that their work adds value. As a final note, organizations regularly find that an individual team has to take steps to satisfy stakeholders external to the team. Development is not unique in this. The Agile approach does not deny this need. It simply asks all stakeholders to verify what they are asking is really necessary and sufficient. Reviewing process for these two attributes is valuable whether your organization is "agile" or not.

Q: Wouldn't you all agree that to effectively implement an Agile Environment, you can not implement 'pieces' of Agile?

I can not speak for "you all" but I'll take my shot.
If you are speaking about a particular Agile methodology, then I would agree with this statement. The explicit methodologies require interactions with multiple groups.   Each group has to understand the methodology and its part in it.

If you are speaking about the philosophy as stated in the Agile Manifesto then every organization can implement a single process improvement to make it more agile. This could be considered a 'piece' of agile. However, as a philosophy, one 'piece' of agile will lead to another and the entire organization will become more agile.

As a specific example for the QAI audience, let's look at the methodology of writing out the steps that are part of a test case. All of the rest of the development methodology can be following an iterative or spiral development methodology, but the test team can decide to reduce churn by writing the test cases in partnership with developers. The approach is to sit with the developers as they write the code and verify that the test case steps match with the code as it is being developed. The tester is responsible for escalating code that does not seem to be conforming to requirements as soon as the tester sees it.

This is a 'piece' of agile because the overhead associated with the documentation of the code behavior for the testers, and the documentation of the test cases for review by developers, is significantly reduced. It also adds a low overhead step of verifying the code is being developed according to requirements before anything is delivered for testing. Yet no other 'piece' of

Agile is required to make this work or add value. All that is required is for the Developers and Testers to agree to work together in this way, and the entire organization benefits.

Q: What best practices have you seen when rolling out Agile successfully? (the steps we can follow before we identify lessons learned)

Burke actually spoke to the concept of best practices in rolling out Agile during the panel discussion. The consulting organizations regularly recommend picking a particular project and piloting Agile on that project before rolling it out enterprise wide. However, Burke also shared the experience of Salesforce.com which completely ignored this advice and went all Agile at one time. In my own opinion there are methods that work, but the entire Agile concept is not old enough to speak in terms of "best practices." There are, however, some issues to watch out for:

- Agile is as much a state of mind as a set of processes. You have to have an Agile organization if the Agile processes are to work.
- An Agile organization is one that is more interested in the success of the organization than the success of individuals. Put differently, individuals derive their success from the success of the organization. This won't make sense to you unless you've seen someone rise in income or status while overseeing a failing project.
- Value is determined by value to the entire organization, not to the individual department. This means that Test may have to do some things that aren't necessarily optimal for Test, but make it easier for other stakeholders to evaluate code quality. Development may need to take some steps that don't necessarily add value to the development process, but make it easier to document compliance with regulations. Testers must not have the attitude of, "It doesn't help me test, so I'm not interested in participating."
- You have to be willing to spend money on tools. Boeing has actually developed an agile process for developing its airplanes. To get there, it spent plenty of money on the tools necessary to model every aspect of an airplane before any pieces were built. Your tools infrastructure enables you to overcome the barriers to change. Don't try to force an Agile process into a tool set that won't support an Agile process. Buy new tools and the training to make them useful.
- Thoreau said, "Simplify, simplify, simplify." An Agilist would say, "Value, value, value." Focus on processes that do not add value proportionate to their cost. That's a problem – fix it.

Q: Can you elaborate a bit on what 'valuable' or 'value-added' documentation is common for a typical Agile project?

I can elaborate from the point of view of the test manager, as that is a role I've held. In a QA role I would work with each team to understand what it needs and move forward appropriately.

Without regulatory compliance, a test lead would need to have at least these things:

- A statement of the goal or the success criteria for the project. This can be called story cards, an elevator statement, or any level of requirements that can be agreed to. Everyone would have to be clear on what successful completion meant, including sufficient quality.
- A description of how the goal will be achieved. This is not necessarily full design documentation, but sufficient documentation of the design that everyone on the team knows the goal can be achieved.
- A description of how the success criteria will be measured. This is a test artifact that shows the approach or strategy that will demonstrate the goal has been met. It will likely include example reports or charts that can be used to share the results of the test measurement to the group.
- A manifest of the build so the test team knows what to test.
- The actual test reports themselves.

Each of these directly contributes to the Test Team's ability to accomplish its goal of documenting the quality of the software at any given point in its development. The first document sets the goal, the second sets the architecture of the software and the test, the third sets the method, and the last two document the test itself.

The list is necessary, in that I can demonstrate how the job can not be done without any of the items. It is sufficient because I can complete my task if I have these.

If there are any further questions feel free to contact me at williamrinko-gay@techisource.com.

References

Conference – Quality Engineered Software and Testing Conferences (QUEST) Chicago, 2008