

*What will happen to my role in the quality assurance process if my organization adopts Agile?*

Is your organization considering an Agile approach to software development? Are you hearing that Agile methods don't require quality assurance (QA)?

Agilists may argue that developers should do their own testing, so QA testing isn't necessary. They may also contend that Agile methods have done away with all the documentation QA requires. You may have heard that Agile doesn't define testable requirements up front. In addition, you may have been told that Agile is a very ad-hoc, unstructured development paradigm.

How should you, as a QA professional, respond? You may be asking yourself, "What will happen to my role in the QA process if my organization adopts Agile?" Before attempting to answer this question, let's take a look at Agile itself.

### What Is Agile QA?

As described in the Agile Manifesto, Agile is a software development approach that values:

- Individuals & interactions over processes & tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

This list describes a philosophy, not a set of practices. But what drives the philosophy?

According to a Gartner study, 74% of all IT projects failed, exceeded budget or were delivered late. While in 2002, the National Institute of Standards and Technology estimated the cost of defect recovery at \$59.5 billion.

The QA community developed numerous methodologies to address these problems, including CMM and CMMI, RUP and ISO 9001. But the problems still persist. While these methods make sense, they have been too inconsistently implemented to really improve the quality landscape.

IT organizations continue to look for ways to deliver higher quality. Agile is an overall description of a set of practices that attempt to deliver quality software on time and within budget.

## Does Agile Work?

This is difficult to answer with the limited studies that have been conducted. Agilists are sold, but there is little in the way of hard metrics to back them up. Quantitative evidence comparing projects of similar scope have yet to be published. Some of this evidence could include, reductions in defects per build or reduced customer complaints per release.

Shine Technologies, an Australian IT Consulting company, conducted a Web survey of Agile adopters and reported that following:

- 49% stated that costs were reduced or significantly reduced
- 93% stated that productivity was better or significantly better
- 88% stated that quality was better or significantly better
- 83% stated that business satisfaction was better or significantly better

These soft metrics should not be the basis for radical change, but they are interesting enough that professionals need to take a serious look at Agile and consider what we can learn from its methods.

## Agile and Requirements Management

Traditional approaches are aimed at locking down requirements at the beginning of the process. Agile tries to adapt to changes in requirements throughout the life of the project. Agile's philosophy is based on embracing change so that, the end product may not look like the original specification, as long as the deliverable still meet the customer's needs.

This is an area where traditional QA could stand to learn a lesson. Traditional QA methodologies do not embrace change. The requirements management process is often burdensome, so end users may not request a change, or may not get approval for an important change. This helps the project stay on time and within budget, but the requirements may no longer satisfy the customer. In the end, that is still a project failure.

Traditional processes were developed because you can't change requirements mid-project without incurring additional cost or delay; this seems an intractable problem. Does Agile truly resolve this?

Even Agilists can't build a product that isn't defined. They perform their actual development in successive short time periods with a very small subset of the overall requirements. The Agile methodologies specify how each successive subset is selected and developed, but these subsets are managed much like traditional requirements. When it's possible, Agilists break the requirements into functional subgroups to make the requirements management easier.

## Agile and Testing

Traditional QA methodologies assign the responsibility of unit testing to a developer. However, developers are not test experts and their unit tests often leave gaps for the QA test team to fill. In some cases, project pressures cause developers to reduce unit testing, increasing the number of defects. Poor and inconsistent unit testing often causes serious problems for the test team.

Agile methodologies employ new tools that make testing a part of the compile and build process. Software that will not pass existing unit tests can not be built into the final deliverable so unit testing is not short-changed. Agile testing is usually automated, which does not have a significant impact on the schedule. The developer must debug and correct any issues found before releasing the software. This structured unit test process makes the testing process more efficient.

Therefore, Agile developers must define unit tests before completing code. Complex systems will still require integrated tests, which can include system integration testing, string testing, environment testing, performance testing and user acceptance testing.

## Agile and the Development Process

Agile has a reputation for being chaotic. Is this true? Agile focuses on the need for a disciplined application of the practices and processes that are proven to add value. For example, Agilists replace some of the administrative overhead with tools to reduce the time spent on non-programming tasks. Agilists also communicate through tightly integrated teams rather than spending time writing documentation. In addition, Agilists include the end user, or a representative of the end-user community, as part of the project team, which keeps the project focused on the user's needs—even when those needs change.

What Agilists endeavor to do is evaluate processes and adapt them to work in the current project. Repeated mini-development cycles allow Agilists to perform process improvement during the life of the project. By reducing process overhead and ensuring processes are useful, Agile methods strive to achieve the same goals at a lower cost.

## Agile and Quality Assurance

An Agile approach may not be a good fit for all projects and all organizations. But where it is appropriate, how can QA increase its probability of success?

**Requirements Management:** An Agile project manager can keep the project goals in mind while developing the current iteration, but an enterprise must manage the entire project portfolio to ensure that IT investments remain in alignment with the business needs of the organization. QA has a key role in verifying that the information between projects is reported in a way that allows the portfolio to be managed well. The leaders of individual projects will want to reduce process overhead to optimize their projects. QA must ensure portfolio management does not become a casualty of single-project focus.

**Testing:** Agile unit testing may be more formal than in the past, but developers are not test experts. The QA test team can help the developers define and even automate the tests that become part of the development tool set. Affording QA skill to the Agile process can reduce the number of defects released to test. QA testing will always have the responsibility to test system interactions that unit tests cannot address, but will have a much higher accuracy based on rigorous unit test, which can reduce the test effort significantly. Furthermore, QA can also use root cause analysis to add tests to the unit test tool, improving subsequent builds.

**Development Process:** If an Agile methodology has to be implemented correctly, QA will always have a role in evaluating that methodology. A project manager can look at the needs of the current project, but the QA team looks at the needs of the organization, past, present and future. QA provides the tools to incorporate the lessons learned on past projects into the development management toolkit, so that future projects can be optimized and deliver even higher quality. It also provides the picture over time, freeing the project manager to focus on the project at hand.

## Additional Considerations

If an enterprise is considering adopting Agile for all its projects, QA can play a crucial role helping make this decision, and making sure Agile is implemented properly. QA can also:

- Help the decision makers focus on data rather than anecdotes.
- Ensure that all the needs of all the stakeholders are being considered.
- Verify that the new processes will allow Agile project teams to support the needs of the enterprise while focusing on individual project needs. QA lets the project managers maintain their project focus while QA focuses on the needs of the enterprise.

There is no magic formula that allows you to increase scope without increasing cost, schedule or both. QA can help the project balance the costs and benefits with the entire enterprise in mind.

## Conclusion

So, yes, there is a role for QA in an Agile environment. As we learn to merge these two disciplines, we can see where traditional quality methodologies have fallen short in the past, and how we can improve in the future. As quality professionals, we should welcome that opportunity.

*For more information, visit our Web site at [www.technisource.com](http://www.technisource.com).*