

5 ideas for result-oriented agile testing



Yogesh Pathak, Founder

yogesh@attagis.com

“Conceptual integrity is central to product quality.”

— Frederick P. Brooks Jr., The Mythical Man-Month:
Essays on Software Engineering

Attagis Solutions Pvt. Ltd., based in Pune, India, provides software quality engineering and solution engineering services to help you excel in your product journey.

We help software builders to integrate quality in their development process. Our teams work as integrated members with client teams working under your processes, your vision, and your culture and ethos.

Agile: A moving target

- 01 A multi-pronged approach to quality**
- 02 Whole team contribution**
- 03 Continuous Review**
- 04 Automation with a purpose**
- 05 Diversified QA skillset**

With the agile methodology being adopted in a majority of software development environments, testing and quality engineering processes have undergone a significant change in the last 10 years. Interestingly, while the impact of agile on software development velocity and product competitiveness has been well-documented, that on product quality has not received enough attention.

At Attagis Solutions, while working with mid- to large-sized enterprises we have observed the unique evolution of quality engineering under the agile model. From this experience, we have extracted five best practices to deliver better quality and result orientation under the agile development model.

1. A multi-pronged approach to quality

From the get-go, the approach to quality needs to be multi-pronged. A variety of techniques including UI test automation, functional/backend test automation, customer-specific testing, regression and sanity testing are available today.

The dimensions that need to be looked after through all this, are product complexity and customer environment complexity. Testing done in isolation without simulating complex or demanding customer environments can result in performance issues, response time delays, or crashes after deployment. Studying this complexity needs collective brainstorming and new, emergent types of testing.

A map of features and their impact on product components allows the team to recognize the areas of the product that will get impacted due to a feature change. This map should guide regression testing strategy in the dynamic environment of agile methodology.

2. Whole team contribution

In the agile environment, developers, test engineers, technical leadership, and product management can all contribute in distinct ways to establish a multi-faceted testing methodology. Developers can maximize unit testing coverage in their code and write code with testing in mind. Engineering leadership can drive intelligent and efficient coverage of the product under manual and automated testing, smoke and regression testing, and new types of testing necessary due to evolving product components. Product management can guide with the rapidly evolving product definition, use cases and expected behavior. They can also guide on customer-specific testing. Finally, the QA team can contribute to virtually all areas of product quality outside actual development. They can develop and own shared understanding of product behavior, own the testing strategies, plans, and schedule, manage the evolving scope of testing, and evaluate the tools to be used for testing, including automation.

3. Continuous Review

Product quality reviews based on pre- and post-deployment incidents and user feedback keep the team focused on what more needs to be done, beyond the ongoing motions of quality-checking. In cross-functional team meetings, QA should express their views on product quality evolution in the recent past as well as over longer periods like a year or more. Regularly held reviews about how to make the product more usable, more reliable, and defect-free are necessary. In other words, whatever has been promised to the customer, needs to be met with a high degree of reliability and quality. It should not be uncommon to dedicate large chunks of agile development solely towards reliability and quality.

4. Automation with a purpose

Test automation has the obvious benefit of taking care of repetitive testing and improving reliability of the testing process (which is otherwise subject to manual errors during testing). Thus, automation can free up QA engineers' time to do more interesting intellectual work of the kind mentioned in this article. Over the long term, automation also means a more reliable product that has been subjected to exacting and rigorous tests on a daily basis. Test automation should not be an afterthought but be woven into the overall engineering and development strategy.

Beyond these general principles, one must keep in mind the rapidly evolving features in an agile environment. The automation tools and cadence of automation should be chosen such that they are able to keep up with (and not lag) the velocity of feature addition.

5. Diversified QA skillset

Given the wide spectrum of testing skills required, a moderate- or large-sized organization can easily take the path of having QA engineers with specialized skills needed for a specific set of tasks.

However, we prefer another approach, which is to let each QA engineer develop all the manual and test automation skills needed for testing a majority of the product.

This results in a QA engineer with better analytical skills, a richer understanding of the product behavior, and having a higher degree of comfort with a diversity of testing strategies.

To discuss your software QA and solution engineering requirements and where we can create value, contact us at yogesh@attagis.com or at +91 98505 52051 (India business hours).