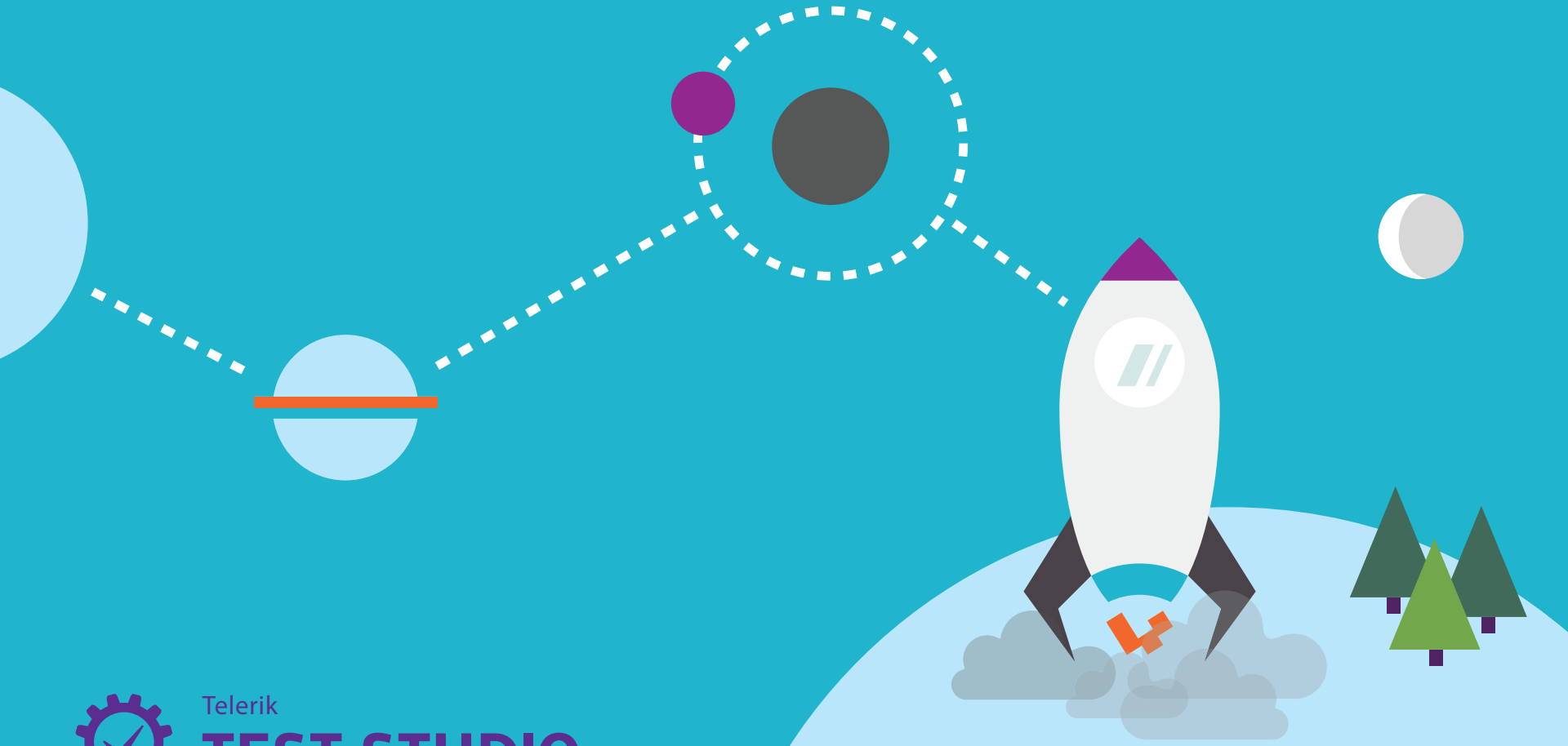
THE AGILE TESTER'S GUIDE TO THE GALAXY

Don't Panic !

6 steps to plan the testing of an agile project



Telerik
TEST STUDIO

A publication of telerik

Contents

Peter Varhol

Peter Varhol is an Evangelist for Telerik's Test Studio. He's been a software developer and software product manager, technology journalist, and university professor among the many roles in his past, and believes that his best talent is explaining concepts and practices to others. He's on Twitter at @pvarhol.

Steven Vore

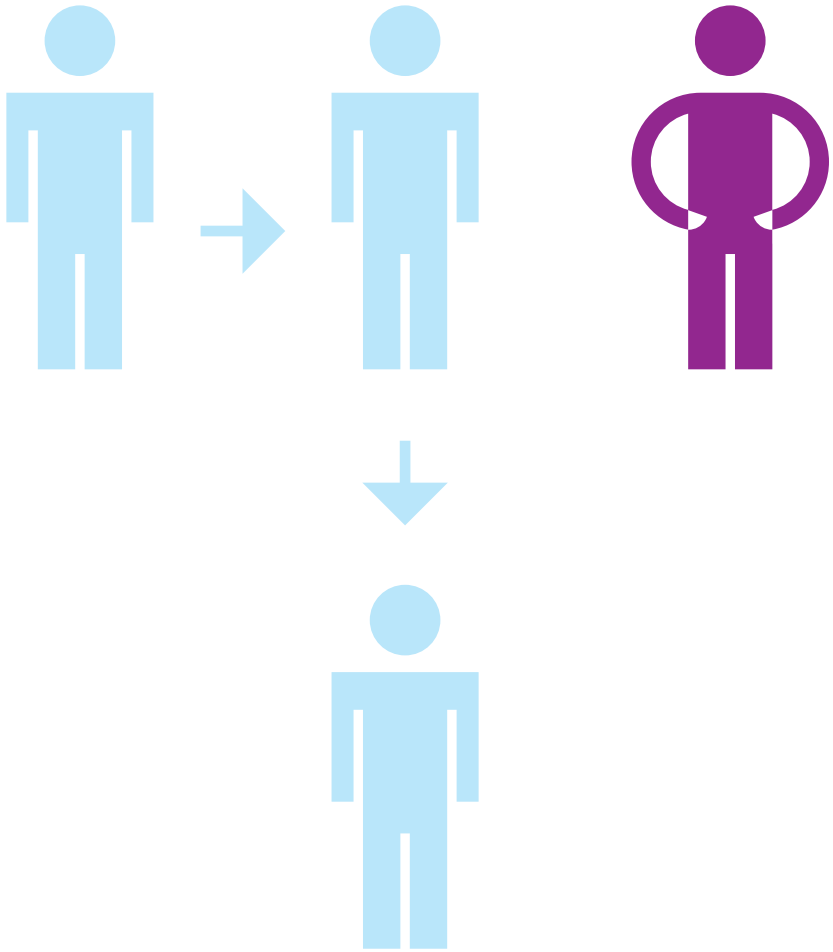
Steven Vore is an Evangelist for Telerik's Test Studio. He has worked in software support and testing for the better part of two decades, and enjoys exploring ways to make software easier to use. He is a fan of movies and music, and can often be found on Twitter as @StevenJV.

IS THERE A ROLE FOR TESTERS IN AGILE?

Application teams of all types and industries are increasingly adopting agile software techniques as the principal method of building applications for commercial or business use. Agile methodologies, such as Scrum, Extreme Programming, Feature-Driven Development, and Test-Driven Development offer the ability to iteratively develop applications with participation from the end user community.

In these methodologies, the role for testers is not always clear. Testers who seek to better understand how testing can impact agile development methodologies, and their roles in agile testing, must take the lead in defining just what those roles are. By understanding agile development processes and how testing can improve process quality, testers can define a role essential in building quality software.

In traditional software development methodologies, testers examine business and user requirements, determine the appropriate way of measuring quality and devise a plan to assure those requirements have been implemented through functional and regression testing during the latter stages of code development. The prospective software user is removed from the process. Instead, one or more business analysts interact with users to determine the business requirements, which are passed on to developers and testers.



AGILE AND DEVELOPERS

Agile methodologies put a premium on immediate and rapid code development, and less emphasis on upfront documentation. Instead, the development team collaborates frequently with the product owner and members of the user community to define and validate software as it's created. These users—the domain experts—define the functional requirements of software through user stories, high-level scenarios describing how the prospective software would be used as a part of a business process.

The team, including product owner, developers, and testers, take ownership of these user stories, organizing them, and using them as a basis for designing and building applications incrementally. Typically taking three to five user stories at a time, developers will reserve anywhere from a week to a month to implement that subset of stories, depending on the agile approach used and the complexity of the stories. Each of the features supporting those stories are coded and tested, and at the end of that time those features are operational. The product owner is an integral part of the team, working closely with developers to answer questions and resolve ambiguities as the stories are translated from business processes to software features.

Once that set of stories is implemented, the domain expert next validates the software with respect to those stories, and the process begins again with the next set of stories. This cycle is performed rapidly, with almost continual application development, so coded and validated features can be deployed at almost any time.

Because the software is built early and continuously—with frequent validation by domain experts—applications can be deployed often, greatly accelerating feature availability to users. Later, software can be adjusted in response to user feedback and changing business needs before the whole cycle repeats.

AGILE AND TESTERS: IT ONLY MAKES SENSE!

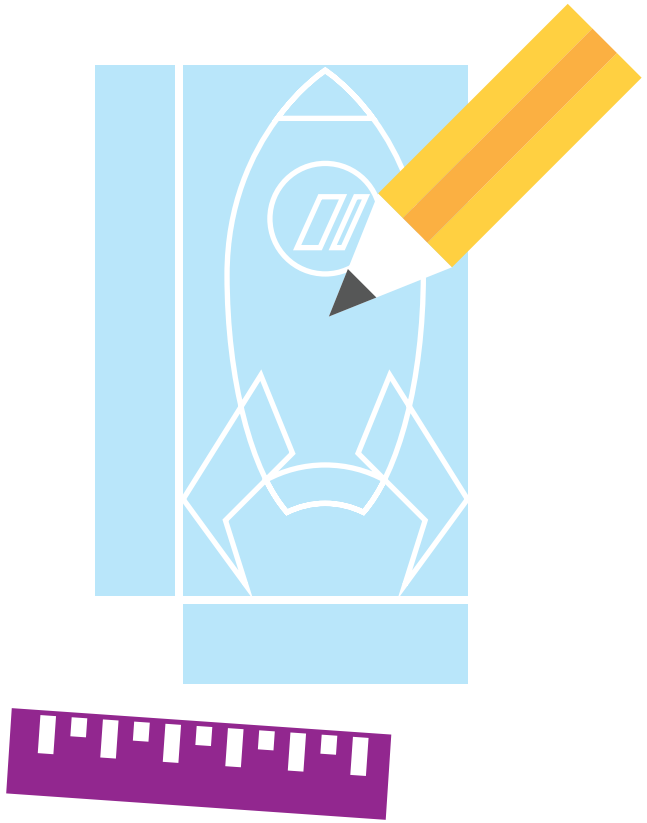
At first glance, an agile process seems to leave little for application testers to accomplish. The connection between end users and developers is fairly direct, rendering it almost unnecessary for testers to confirm the end product meets user specifications. And because domain experts try software multiple times throughout the development process, mid-course corrections are usually seamless and straightforward. Developers find and fix the obvious bugs, while domain experts advise on fitness for purpose.

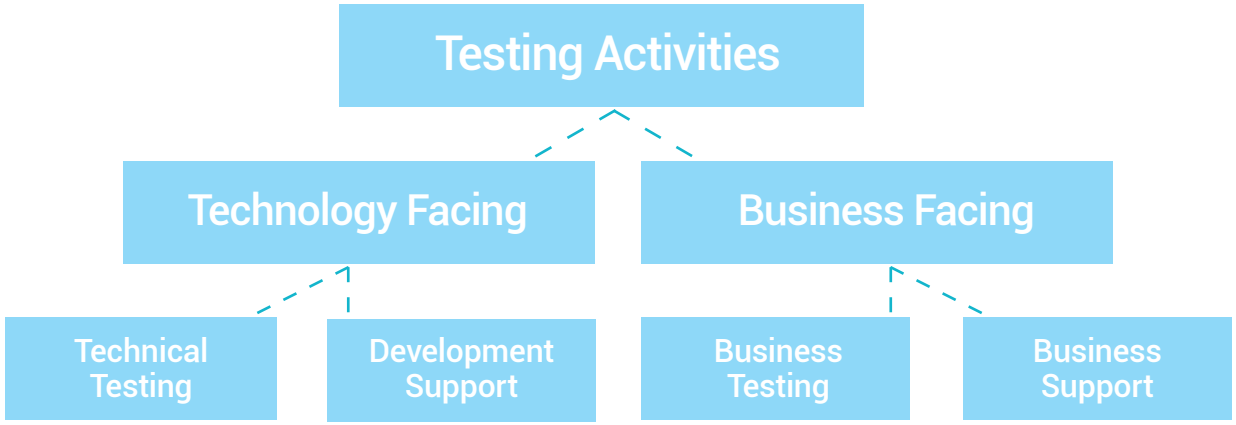
But agile doesn't invalidate the need for independent testing. Quality is still one of the most important application requirements, and testing is the primary way to verify quality and assure quality software production. This means testers have to substantially change their job approach to assume an essential role in agile development.

The New Role of Testers in the Agile Process

For instance, testers are best equipped to connect business needs with actual results because of their unique position in the application development lifecycle. In particular, testers are in the best position to codify user stories, and determine how to objectively measure their completion. This is a business aspect to testing that is not typically the focus in traditional methodologies.

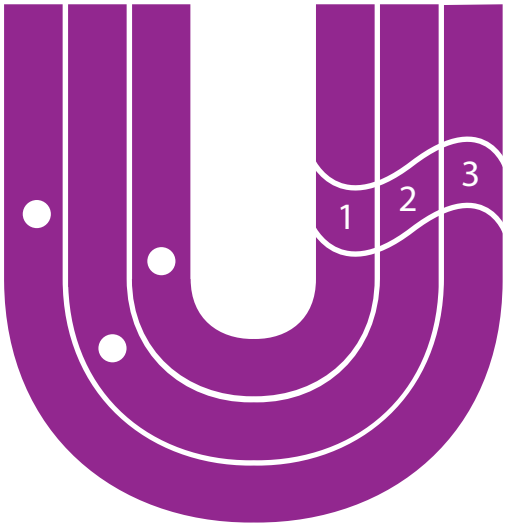
Testers often have, or can develop, a better understanding of user needs than developers can. Likewise, testers have more technical understanding of application design than users would need. This unique position allows them to take a holistic approach during user story definition, reconciling conflicts that would otherwise not be noticed until development is complete. Being part of these design discussions, testers can save the entire team time and frustration.





Many misunderstandizngs can be avoided by early tester interventions. If the domain expert is the principal author of the user story, likely he or she is making assumptions about the domain or the business process implicit in the story, but not likely apparent to the development team. This can lead to misunderstandings with developers who don't fully comprehend the assumptions or domain expert perspective. Testers are in the ideal position to work with the domain expert to highlight assumptions, make them more explicit, and clarify any story ambiguities before developers start coding.

Because testing is accelerated, it must be focused on user community needs and be designed to provide technical vision for quality. All test cases should tie back to the user stories, as well as to nonfunctional needs and organizational policies. By enabling tester and domain expert collaboration in defining software scope, testers are in the best position to determine how to test stories once they are implemented.



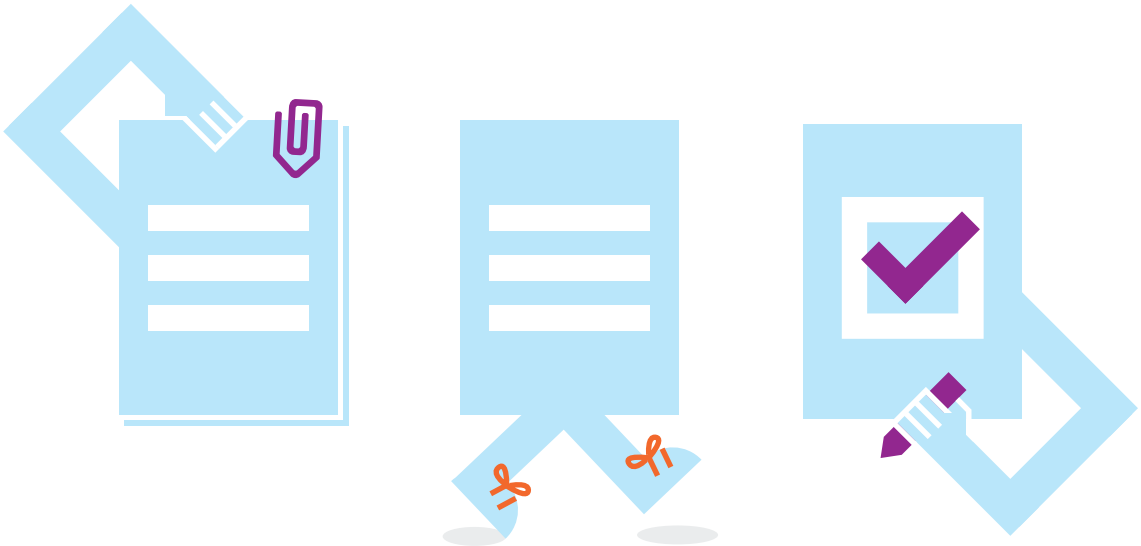
The Importance of Acceptance Testing

Agile methodologies tend to leave acceptance criteria up to the domain expert, but that person is unlikely to be an expert in evaluating software, —even against criteria he or she defined. Testers are especially qualified to that role, as it is the essence of the profession.

In traditional testing, this is referred to as "acceptance testing," a practice that ensures every business requirement has been satisfied through a corresponding software feature. To facilitate acceptance testing, testers need to be intimately involved in translating user stories into application features. Doing so enables them to understand what the user community intended, and how it was translated into the application.

In agile development, acceptance testing needs to be performed at the user story level. Making crucial transitions from story, to implemented feature, and back to story requires participation in story refinement, story break down into features and test case development for those features.

This level of upfront involvement makes getting from user story to acceptance test significantly easier. Even so, tracing user stories implementation has the potential to be much faster, simpler and require far less overhead. Because developers aren't writing extensive specifications, and test plans aren't required to reflect non-existent specifications, testers can proceed right to building test cases that reflect the particulars of one or more user stories. An acceptance test case should enable the product owner to verify and validate that the user story has been correctly implemented.



The Crucial Role of Test Automation

Virtually all these activities can be improved and accelerated with automation. Tracking a story from inception to acceptance test can be done manually, but that means testers have to spend time examining documentation to perform the activity. A requirements and test management package will enable testers to record user stories and resulting features, while tracking acceptance test results back to both features and user stories.

Likewise, functional, regression, and acceptance testing can be done manually, but at the cost of additional time and resources. However, that's not being agile. Once tests are validated, most can and should be executed automatically, even as part of the build process. Automation accelerates testing, user acceptance of the application and ensures repeatability of an important part of the agile development process.

THE 6 KEY STEPS TO PREPARE & EXECUTE THE TESTING OF AN AGILE PROJECT

For testers to take a leadership role in assuring the quality of agile software development and delivery, they have to work with—and at—the project pace. This means working with both users and developers in defining the project and ensuring its quality and fitness for the user business needs.

Here are 6 key steps involved in preparing and executing the testing of an agile project:

1. Get to know the domain expert and user community. Fundamentally understand the business goals of the application.
2. Break down user stories into prioritized testing needs and track those needs until completion. Use automated systems to capture user stories, distill requirements and trace requirements through to implementation, and back to user stories.
3. Translate testing needs into test cases as early as possible. Work with users and business analysts to ensure the test cases reflect real business needs. Work with developers to devise technology-facing tests, such as integration and unit tests. Use automation tools such as Telerik's TeamPulse and Test Studio to enhance communication between developers, testers and users.
4. Automate test cases and test execution using automated testing tool so tests can be rerun automatically, ideally as part of the build process.
5. Track test case execution to ensure the fitness of the application. Work with both project and testing tools such as TeamPulse and Test Studio. Be ready to report on test execution at any time, so decisions can be made on the deployment side.
6. Trace requirements and user stories from inception to delivery to ensure business needs have been adequately addressed.

vyou maintain the performance gains you achieve.

These are the ways testers can demonstrate value and make essential contributions to software development and delivery in an agile process. An equal focus on both the business and the technology, with continual support for the agile team and development project, enables testers to make a complete contribution to project success. The automation of these activities can ensure timeliness and repeatability of testing activities, not only in the current iteration, but future iterations as well.

HOW TELERIK'S TEAMPULSE AND TEST STUDIO CAN HELP YOU

TeamPulse provides a management approach that, when used in conjunction with Test Studio, lets agile teams and in particular agile testers have visibility into user stories, turn those user stories into acceptance tests, run tests, report defects, and determine whether or not user stories and other requirements were successfully implemented. In addition, thanks to the integration business, users always have full transparency into the software development process. Finally, TeamPulse facilitates communication between team members and with the user community on the state of agile iterations and the features that are working and ready for use.

Test Studio provides an important component of automation in the agile equation. It automates functional performance and load tests, enables scheduling at the completion of builds, collects analytics enabling testers to quickly judge iteration and feature quality and identifies defects against user stories and requirements.
Agile testers can use both solutions together to seamlessly collect project and test data that can increase responsiveness to the speed and flexibility of agile techniques, ultimately delivering better software more quickly.

TRY OUT TeamPulse

TRY OUT TEST STUDIO