

Agile Record

The Magazine for Agile Developers and Agile Testers



Distributed Team Management

www.agilerecord.com

free digital version

made in Germany

ISSN 2191-1320

August 2013

issue 15

Integrating Test Activities in Agile Projects

by Leo van der Aalst

Introduction

People not working in agile sometimes have the impression that agile projects are chaotic, disorganized, and uncontrolled. Indeed, if discipline and structure are really lacking, an agile project will deliver poor quality software or will not deliver at all!

A proven approach to measure quality is product testing. However, both the *Agile Manifesto* [1] and Schwaber and Sutherland's *Definitive Guide to Scrum* [2] leave few clues on how to do that.

After addressing the lack of concrete steps on testing in agile and Scrum approaches, we describe our vision for agile testing, followed by suggestions for effectively integrating test activities into the Scrum approach.

The Agile Manifesto and Scrum Have a Lack of Footholds for Testing

The basis for all agile projects is the *Agile Manifesto*. The four values and twelve principles of the manifesto say nothing concrete about specific approaches to systems development or testing. An agile approach such as Scrum, for example, scarcely provides a foothold for quality assurance and testing. Moreover, the *Definitive Guide to Scrum* does not add much knowledge on quality and testing. In addition, real-life practice shows that the meanings of and the relations between lean, agile, and Scrum are not equally clear to everyone. It will be no surprise that the integration of a test and Scrum approach does not always run

Benefits of a Structured Approach to Testing in Agile Projects

- It gives insight into and advises of any risks that may arise in relation to the quality of the system under test.
- Defects can be found and tracked at an early stage, providing better insight into the root cause of the defect.
- Defects may be prevented in the first place.
- While test execution tasks remain on the critical path, delays should have fewer consequences.
- Test products, such as test cases, can be reused.
- The test activities are transparent and controllable.
- The automation of testing tasks (e.g. test specification and/or test execution) can be pulled into the project earlier.

smoothly. Actually, in their *Definitive Guide*, Sutherland and Schwaber state that “Scrum is ... extremely difficult to master.”

In the implementation of an agile approach, organizations find it difficult to gain a good overview of which parts of their current test approach can be reused and which cannot. The only thing the *Definitive Guide* says about testing is that there are no sub-teams dedicated to particular domains like testing. For this reason, it happens all too often that the organization decides not to adopt any existing testing practices at all. However, this means that the benefits of structured testing are not realized (see box below). In our opinion, to make sure that product quality can be measured, we must integrate a structured testing approach with Scrum events.

Vision for Testing in Agile

The four values and twelve principles of the *Agile Manifesto* have a neutral relationship to a test approach. Scrum, which is an approach based on the manifesto, scarcely provides any guidance for integrating testing within its model. In the next section, we will first provide a brief overview of our vision for testing in agile environments. It is possible to regard each statement in this vision as an add-on testing value for the *Agile Manifesto*. For a detailed understanding of this vision, we refer to the point-of-view paper entitled “Testing in Agile Software Development Environments with TMap NEXT” [3].

The four statements of the vision, each followed by a short elucidation:

1. Use the Agile Manifesto as your starting point.

The four values and the twelve principles of the *Agile Manifesto* form the starting point of each test activity in an agile approach.

2. The test process must be integrated into the agile approach.

a. The test activities must be integrated in the development process.

This means that testing is not a separate phase, rather a continuous activity of an agile team. This also entails a different implementation of test levels.

b. All team members must be prepared to perform test activities.

Although the team should contain a professional tester, this does not mean that all test activities can be carried out by this tester alone.

- c. Testing is the driving force behind the quality of the project.

Testing must not be seen as the last safety net before the software is implemented. The tester collaborates with all team members to provide continual information on the product's quality and its satisfaction of the business requirements.

- d. The use of automation is becoming increasingly important and indispensable in the realization of a successful agile project.

Model-based test design (automated generation of test cases) and model-based test execution (automated execution of these cases) are valuable practices here.

- e. Testing must be incorporated into the definition of Done.

The *Definitive Guide to Scrum* states that “each increment is additive to all prior increments and thoroughly tested, ensuring that all increments work together.” Thus, it is important to include test aspects in the definition of Done.

3. Find the right balance by making conscious choices.

For example, working software is more important than comprehensive documentation, which means that the lack of documented information must be compensated for with more communication.

4. Use the strength of the four TMap NEXT essentials.

- a. Be adaptive.
- b. Use techniques and tools suitable for agile environments.
- c. Apply a business-driven test management approach geared to the agile environment.
- d. Use the TMap NEXT lifecycle and TMap NEXT activities in an agile way.

The Test Method as a Basis for Integrating Activities

The test method we have integrated with Scrum events is TMap NEXT® [4]. This is a test method consisting of four essentials, seven phases and, as its basis, 53 activities.

An important essential of TMap is its adaptiveness. This enables TMap to be applied in a flexible way to any situation. In order to apply TMap well in a certain situation, it may be necessary to alter, remove, or supplement some activities and techniques. As it can sometimes be troublesome to keep a good view of the overall situation, we will now demonstrate how TMap can be integrated with Scrum to support a truly agile approach in which testing is fully integrated.

Testing Must Be Fully Integrated

Experience has taught us that testing is not only an extremely important activity in a Scrum approach, but has also led us to

believe that testing must be fully integrated with this approach in order to be as agile as possible. Of the many Scrum variants with different concepts, we have chosen Schwaber and Sutherland as a basis, because it is used most frequently and lays the groundwork for elaborations taking account of new insights.

Please keep in mind that both Scrum and TMap are merely approaches supporting the project team in deliverance of a product at the right time to market, with a fit-for-purpose quality, at an affordable cost, and with an acceptable risk level. An approach can be adopted without any adjustment on a one-to-one basis, or can be taken as a guideline to be regularly realigned as needed. In the first case, chances are it will not work perfectly on all fronts, as all projects are different. In this way, the approach may become an irritably tight straitjacket.

The second case requires more initial effort to learn and implement, but it provides a tailored, flexible framework geared to an ever-changing world.

Our suggestions will have to be implemented in an adaptive manner to comply with your own specific situational and project requirements. However, we assume that with this article and the detailed explanation of our vision for testing in Scrum in the book *TMap NEXT in Scrum* [5], we can help you along the path of measuring product quality.

Mapping TMap NEXT Phases on Scrum

Based on this vision, we have mapped the TMap lifecycle phases on the Scrum model. The result of this mapping is shown in Figure 1.

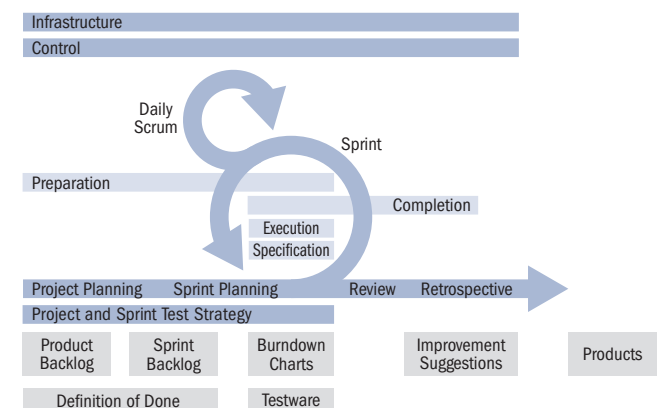
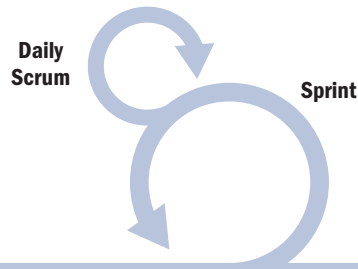


Figure 1: TMap lifecycle phases mapped on the Scrum model

The phases originally contain many activities. Some activities are relevant for the Scrum approach, others are not. We have determined which are relevant and matched them with Scrum events.

Test Activities in One Overview

In this article, we will not explain how the individual test activities within the phases are integrated with Scrum events, but rather show you the end result immediately. If you are interested in a detailed explanation please refer to the *TMap NEXT in Scrum* book [5]. Figure 2 demonstrates how test activities are integrated with Scrum events. These activities can be carried out



Sprint 0	Project Planning	Sprint Planning		Review	Retrospective
<ul style="list-style-type: none"> Formulation of DoD (and/or DoR, DoS) Make agreements about the inclusion of which test levels to include in the sprint (include in DoD) Determine high-level product risk and project test strategy (input for project planning) Make high-level estimation of test effort (input for project planning) Training for test knowledge and/or test skills (Begin to) set up the test infrastructure including tools Set up testware management (include in DoD) 	<ul style="list-style-type: none"> Determine product risk of each backlog item and record it on the (story) card (input for planning poker, assigning story points) Evaluate backlog items and communicate obscurities with product owner Set up test infrastructure as a technical product backlog item and add to product backlog (optional) 	<ul style="list-style-type: none"> Determine test strategy (test intensity and test design technique) for each backlog item Place rows from the test strategy table as tasks on the scrum board (input for estimation of time) 	<ul style="list-style-type: none"> Execute tasks (create and execute test cases) Communicate defects and register if required Build up regression test set (if included in DoD) Execute (automated) regression test Report progress in daily scrum (update scrum board and burndown charts) Automate test execution (if included in DoD) Preserve testware (if included in DoD) 	<ul style="list-style-type: none"> Perform product demo (acceptance by product owner) 	<ul style="list-style-type: none"> Evaluate process (retain the good points and improve the poorer ones)

Figure 2: Total overview of test activities in Scrum events

by all team members. Please note that it is possible to adapt the total overview to your needs. Various activities may not be relevant for your project, they may be executed at some other time, or you may need to add your own specific activities. The model is thus easily adapted to the situation within your own organization or project.

In More Detail

There is a lot more to say about testing in a Scrum environment. We cannot address all testing aspects in this article. Therefore, we have identified some aspects which are always subject to discussion. These aspects are:

- What are the roles of the product owner and the Scrum Master with respect to testing?
- Which team members perform test activities?
- What is the best way to deal with test strategy?

What are the roles of the product owner and the Scrum Master with respect to testing?

The main focus of these two roles must be the criteria they set both for the level of quality needed to reach a fit-for-purpose business solution and for the level of risk acceptable for the organization. Further, the product owner is involved in accept-

ing the product, which requires participation in acceptance of testing activities.

Which team members perform test activities?

In agile teams, all team members simply have the title of developer. That said, many different capabilities are needed within the team, requiring a variety of members.

Next to various experts, it is advisable to have a professional tester in the agile team who can guarantee expertise on quality and testing. This tester should possess experience performing risk analysis, reviewing artifacts, and formulating and executing test cases using relevant test design techniques. But this does not mean that all test activities must be carried out by this tester. All team members should be prepared to perform test activities. To be able to perform the activities necessary in each sprint, all team members must take part in the specification and execution of test cases, for example. In such a case, the experienced tester can act as a coach to his team members. Of course, this works reciprocally: the tester may be asked to support other team members in their tasks, requiring a broadening of the tester's own knowledge and skills. Further requirements of a tester in an agile environment include communication skills, flexibility, knowledge of the domain, creativity but also practicality, solution and customer orientation. A tester should be a team player who supports the product owner in

the formulation of product backlog items and acceptance criteria, supports the business analyst, can evaluate unit tests, and is prepared to pair with a developer on both design and programming. However, the most important characteristics are the abilities to be proactive, open-minded, and with their team responsible for the delivery of a good result.

What is the best way to deal with test strategy?

In the Scrum model, we distinguish two events in which we could define testing strategy (see Figure 1): project planning and sprint planning.

Project Test Strategy, Defined in Project Planning

The planning schedule for the total test process is formulated at the beginning of a project. However, Scrum is not primarily concerned with the actual planning of tasks in terms of fixed points in time. These are placed on a Scrum board and become active when the time is right. At this point, the focus lies more on defining a *global test strategy*. We refer to this as the project test strategy – in a traditional development environment, this would be a component of the master test plan. It is good to be aware that the formulation of the project test strategy takes place during the planning Scrum event and is incorporated into the project planning (product backlog) schedule.

The main objective of creating the project test strategy is to establish what testing activities are suited to cover the risks identified for the product created by the team. This includes deciding the test depth required for various parts of the product and the level of regression testing to be performed.

Sprint Test Strategy, Defined in Sprint Planning

The sprint test strategy is defined during the sprint planning event and included in the sprint backlog. For example, during the sprint planning stage, the development team estimates (with the aid of planning poker) the amount of time required for each task listed on the sprint backlog. The amount of time required can be influenced by factors such as the required thoroughness of the test, which is related to product risk. Therefore, we recommend including the risk classification of a backlog item – particularly in the case of user stories – in addition to the priority specified by the product owner, *before* the planning poker is initiated. To make this practical, just add the risk level to the product card on the Scrum board. During the planning poker session, the risk level is one of the factors which determines the number of story points.

Conclusion

While the suggestions in this article will probably require adaptation to comply with your own specific situation and project requirements, we assume that this article and an explanation of our vision for testing in Scrum can help you along the path to increased efficiency.

Sources

- [1] Beck, K. et al, *Agile Manifesto*, Ohio, 2001. <http://agilemanifesto.org/>
- [2] Schwaber, K., J. Sutherland, *The Definitive Guide to Scrum: The Rules of the Game*. 2011. http://www.Scrum.org/Portals/0/Documents/Scrum%20Guides/Scrum_Guide.pdf
- [3] Davis, C., L. van der Aalst, “*Testing in Agile Software Development Environments with TMap NEXT*”. Vianen, 2010. <http://www.tmap.net/tmap/downloads/testing-Agile-software-development-environments-tmap-next>
- [4] Van der Aalst, L. et al. *TMap Next for result-driven testing*. UTN Publishers: 's-Hertogenbosch, 2010. ISBN 90-72194-80-2 (available on www.ict-books.com)
- [5] Van der Aalst, L., Cecile Davis, *TMap NEXT in Scrum – Effective testing in Agile projects*. Sogeti Nederland B.V.: Vianen, 2010. ISBN 978-90-75414-64-6 (book), 978-90-75414-65-3 (ePub) (available on www.ict-books.com)

> about the author

Leo van der Aalst



Leo van der Aalst has more than 25 years of testing experience. He is experienced in international consultancy projects and has developed agile testing services.

Leo is a lector on software quality and testing at Fontys University of Applied Sciences in Eindhoven, The Netherlands. He is also a co-author of the *TMap NEXT® in Scrum* guide and the *TMap NEXT® for result-driven testing* and *TMap NEXT® Business Driven Test Management* books. A member of the Dutch Country Board of ISO/NC 381007 “Software and System Engineering”, he works both as a member of the R&D unit of Sogeti Netherlands and as development lead of the ISTQB Agile add-on syllabus.

Finally, Leo is a much sought-after teacher of international test training courses, a regular speaker at national and international conferences, and the author of several articles on testing in agile software development environments.