Agile Testing Documentation
- How much is enough?

By Test Advisory Services Team

For questions and clarifications, please contact:

Mindtree Contact	
Contact Name	Navaneetha S Kowdle
Designation	Director – Test Advisory Services
Phone (M)	+91-9886203041
Email	Navaneetha.kowdle@mindtree.com
Address	Bangalore, India

Introduction

The Agile way of working for software development has become the norm and naturally, software testing has also evolved over time to firmly embed itself in the process. While some intrinsic activities pertaining to software testing have remained largely unchanged with suitable adaptations to align towards Agile (such as test case design & execution, defect logging and triaging), an important aspect that is clearly impacted (due to short release cycles and therefore a perceived paucity of time) is documentation. This document gives a perspective on documentation considerations for testing for Agile projects.

Waterfall vs Agile – A comparison

The waterfall world of software development was easier as each activity pertaining to software testing was individually bucketed (viz Test Requirement gathering, Test Design phase, Test Execution phase, Defect Logging etc.), with each activity having the luxury of weeks (if not months) to be completed.

Consequently, each phase came with robust documentation as a part of the deliverable to be used by the subsequent phase. A project typically had a detailed test strategy document, comprehensive test plan, detailed test cases, test execution reports, defect triage reports etc. (Figure 1)
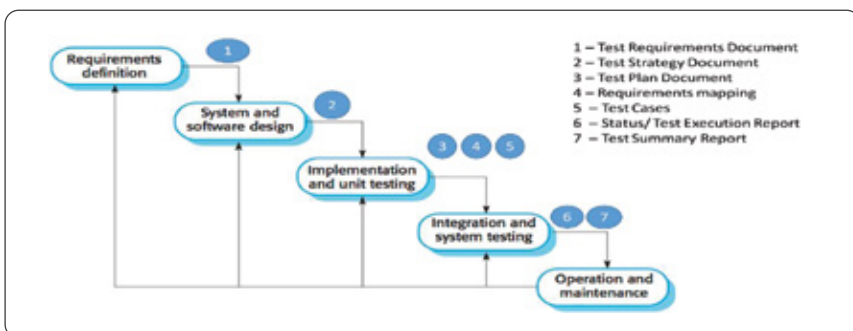

Figure 1: Test Documentation for Waterfall

As the Waterfall model typically spans over a longer period between releases, it makes good sense to capture all details through relevant documentation. However, with Agile, a serious consideration of the cost benefit analysis of the need for documentation vis-à-vis the productivity gain/loss from it, considering the quick release cycles must be made. If we were to map the above documentation as-is into the Agile framework, the documents and the timelines would look something like the below (Figure 2)
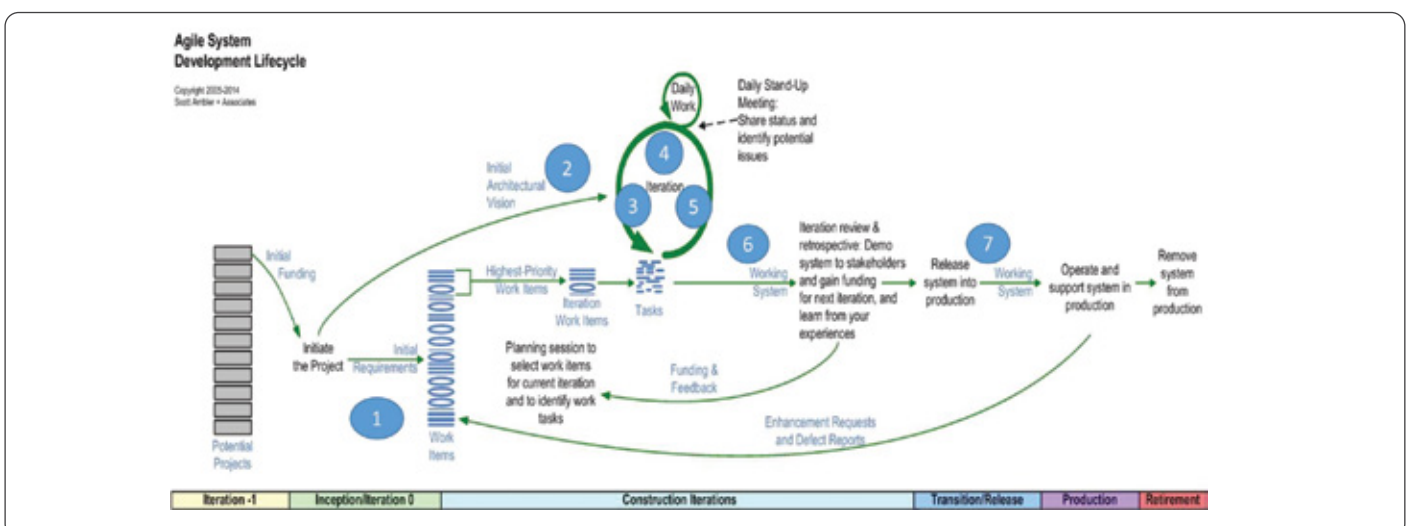

Figure 2: Test Documentation for Agile

Obviously, with a 2-3 week sprint, the question begs to be asked on whether so much of test documentation is needed. In fact, some Agile evangelists are of the opinion that documentation is a complete waste of time and not needed. On the other end, traditionalists believe that developing software without all the necessary documentation for posterity, regardless of the development methodology, is a bad practice. The reality lies somewhere in the middle. Documentation cannot and should not be eliminated from the Agile process. At the same time, we should not be dedicating time to document each and every activity in an already crunched release cycle. So, what is the correct approach for test documentation in Agile? The answer can be arrived at when the following questions are addressed:

Who will read the documents that are developed?

Documents should not be developed just because they ought to be there. The users of the documents should first be identified and only if there is a sufficient need for it, should a document be developed.

How much time is needed and who should contribute towards the documentation effort?

In the Waterfall approach, each document has a single owner. In the Agile world, it would be both time and effort-consuming to do it this way. Typically, a test artefact should be a live document that has contributions from multiple people. While the template/standard should be consistent, the onus of keeping it updated should lie with multiple contributors based on their own responsibility towards the QA activities (both at Sprint and Release level). From a time standpoint, the effort towards documentation should be factored into the velocity and should ideally be a small component (~5%).

How long will the document be in use?

This is a key question that needs to be answered before embarking on a particular document in Agile. For example, the effort towards the Test Strategy document should be upfront and the lifetime should be until the end of the project spanning across multiple sprints/releases. Of course, it will undergo minor updates during the course of the project. On the other hand, artefacts such as Test Summary report etc. are needed only for the day/run. Consequently, the question on how much effort should go into these reports rises. Should this be standardized at a daily level or can/should this be automated with no manual effort? Test Plans are typically at a sprint level based on the features and should be a continuous work-in-progress document across sprints.

What should be the size of the document?

Bigger is definitely not better always. The Waterfall approach needed comprehensive documentation Because the timelines were longer, with multiple requirements being addressed together, and there was a good chance of multiple people using/requiring the same document due to factors such as team size, attrition etc. Documentation in Agile must be concise and should have only the informtaion needed by the user and no more. Both Waterfall & Agile can have a Test Plan, but the contents will/should be vastly different. The Test Plan for a Waterfall project will be voluminous because it is meant to contain information on hundreds of test cases. For an Agile project, the focus will be on just

that sprint and therefore, as mentioned earlier, ~5% of the sprint effort should be spent on documentation. This means that precise and concise information will be updated as and when required.

Conclusion

Each project is unique and so are the documentation requirements. Based on the type of project and maturity of the team, some specific types of test artefacts should be created. Agile does not translate to no documentation. It only means that selective and concise documents that aid the team for the present and for posterity ought to be created and retained. However, the following points should help the team in creating the right set of documents. To summarize, the test documents should:

a) Address only the consumers

b) Be precise and concise

c) Be a live document and allow inputs from multiple members

d) Lastly, it be created only if there is a definite need, based on requirements of the project

About the Author

Navaneetha Kowdle

Senior Director, LTIMindtree

Navaneetha Kowdle is the head of Advisory Services - Quality Engineering at LTIMindtree. He is an experienced IT Professional with 23 years of extensive global experience in the areas of QA Consulting, QA Program Management, Account Management and Client Relationship Management.

- -