

Test Management and Challenges in Agile Scrum Methodology

Koti Reddy Bhavanam PMP, Project Manager
Innovation Delivery Centre, UnitedHealth Group Building
#14, Raheja Mind Space, Hyderabad. Hi tech City, Hyderabad, India
bkotireddy@gmail.com

Abstract

in the current economic conditions many organizations strive to continue the trend towards adopting agile processes to get the benefits like quicker return on investment, better software quality, and higher customer satisfaction. Since the beginning of the current world financial crisis many technology-driven companies have suffered the effects, being forced to lay off people or drastically diminish costs (Wauter, 2009). The survival of the company itself becomes dependant of the time-to-market, deliver on time to the customer and minimize costs. As a result, many organizations are now aspiring to adopt agile practices. In this process, agile testing became a big challenge and the reasons are many which include mindset of the stakeholders, level of the documentation at the beginning, process setup, continuous integration set up, metrics, accepting criteria, developers to tester ratio and the tracking of the test life cycle. This paper highlights the major challenges, root causes, solutions and best practices to be a successful agile tester which essentially leads to benefits of agile scrum testing. Also there is a focus on agile scrum metrics which are vital in measuring the agile testing performance.

Keywords: Agile Software Development, Scrum, Project Management Methodologies, Test Challenges, Test Root Causes, Pair Testing, Test Process, Test Metrics, pros, cons, Best Practices, Benefits

I. INTRODUCTION

The goal of the agile development is the ability of the team to understand, implement, tested and delivered in given short time frame. Team should be able to produce an increment of shippable code at the end of each cycle. This is going to be a significant challenge for the team which may be achieved through some agile training and guidance. In general everyone thinks that managing testing activities in agile methodology is a nightmare and most of the us thinks that agile may not be good for them as they cannot test the application as there is no documentation available and there is a scope for frequent requirement change. The test strategy is usually made with the best of estimation and forecast but implementing the same became highly impossible when there were delays in obtaining the test builds, failures in Build

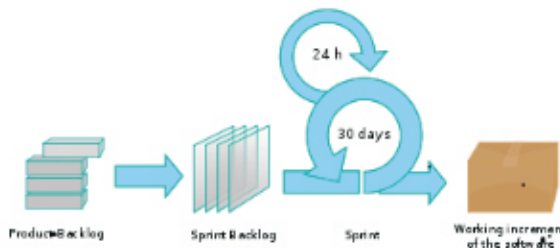
Verification Tests, Unexpected failures in configuration of test environment, changes in requirements, etc. Also many times, critical issues went unnoticed and later surfaced as risks. This caused conflicts within the team and sometimes delayed the release of the software. Apart from managing the activities to shipping of bug free software in time, management had the challenge to improve the productivity and performance of the team.

To avoid all the communication gaps and improve the test efficiency and effectiveness we need to have a better test strategy which can address all the issues discussed above. Also in traditional testing the process is heavy and need to wait a longer period and after spending a whole bunch of money the stake holders realizes that this is what not we are supposed to build and this is what we are not supposed to test. Hence agile

testing plays a vital role in understanding the requirements better and delivering the same with pre requisite quality alongside with the development.

II. SCRUM MODEL

The following diagram depicts the very well known scrum process. Scrum is an iterative, incremental framework for project management often seen in agile software development, a type of software engineering.



approach at the same intensity as development. Usually in agile scrum there is no difference among programmers and testers and everyone has to work as a team and have their own strategies for coding and testing.

The goal of the agile development is the ability of the team to produce an increment of shippable code in given short time frame with predefined quality.

III. AGILE TEST CHALLENGES

The two top most challenges in the agile testing is following test driven development and having continuous integration which includes build deployments and test automation. As for agile testing in particular, a great advantage is Test-Driven Development, wherein the tests are created before the code, and depending on the agile method used might be created by the developer in extreme programming or created by anyone of the team member in Scrum. Also we have the liberty of having the tests written by any of the team member depending upon their skill set, no biasing between any team member and everyone considered to be a developer rather than

programmers and testers. When it comes to continuous integration we need to have the all infrastructure ready which will be taking care of the build and deployment of the entire code after running the automated test suite. This will ensure that the build is clean and we are not introducing any new defects in to the system. If this advantage is lost, and then testing becomes a nightmare, as new increments are introduced many times per day, under short deadlines and with the expectation of a regression being run each time. Without automation, this can be almost impossible. In this process you should not expect some freeze documentation or spec and stable build like in water fall model, this is a continuous process. You may have to accept some changes as long as they are small or you need to push the changes into backlog and you should be able to test on a continuous build. The major challenges are in spite having small iterations unable to deliver the user stories as planned and the following are the mostly seen challenges by most of the agile followers.

- a. Mind set to get start with
- b. No detailed requirement documentation
- c. Lack of proper test process strategy
- d. Build readiness for testing
- e. Lack of continuous integration
- f. Poor quality
- g. Trade off on number of user stories to be taken when there is a bandwidth ratio issues between developer and testers
- h. Customer satisfaction

To address the above challenges one has to have clear vision of the deliverables and proper expectations set up front and track the same. Having clear exit criteria also help. Also one of the useful means is to have mid sprint health check to keep a tab on the sprint health to figure out the key indicators of risk if any. Also the sprint retrospective ceremony is one of the best tools to fix the potential issues identifies from the previous iterations and need to make sure that these issues are not repeated again.

IV. TEST CHALLENGE'S ROOT CAUSES

The root causes are many and from the experience and various surveys few of the causes are shown below.

- a. Just wanted to call themselves as following agile
- b. Not having clear requirements at the beginning
- c. Late feedback from the stakeholders and customers
- d. Lack of forecast in technical complexities
- e. Unrealistic estimations
- f. Lack of understanding in system integration
- g. Lack of identification on impact areas
- h. Stake holders not being involved in early stage of the project
- i. Not understanding the definition of DONE
- j. Not setting the right expectations upfront
- k. Not escalating the issues on time
- l. Not being proactive in communication

The best way deal with these root causes is having a proper review with the requirements documents and have a mutually agreed up on scope and track the changes to the sprint backlog during the sprinting. Set the ground rules for measuring velocity with an agreed up on bench marking velocity. The definition of DONE along with the exit criteria is most crucial part to have the sprint closure smooth. Also the burn down plays a vital role in completing the task estimation remaining effort. If the task effort is not properly burn, will leads to chaos, so one has to be very careful on updating the remaining effort and the trick part is to update the more time to required while updating the tasks, it is not just reducing the effort spend from the original planned estimate.

V. TEST PROCESS

Agile usually means fast development in short cycles with frequent releases. A lot of times this is done under time constraints. In testers perspective this has the potential of being very bad.

The developers usually have to finish their tasks in a short time which leads to bad code with a lot of shortcuts. The testers have to test this bad code also under time constraint which leads to many undiscovered bugs. Usually the team has time to fix only the high priority bugs which leads to a buggy releases with many unknown and known bugs. To overcome the above said issues you need to have a proper plan for incremental code testing. To success in agile you have to define your own strategy to follow as a process. The key is to know whether agile fits into your business or not but most of the cases the agile process is feasible. Also testing personal usually start the work along with developers, while developers are coding the testers should be completing the test plans and ready for the manual testing based on the test plan prepared. Also ad-hoc testing is encouraged and the flexibility is given to have bug fix window before raising the issues.

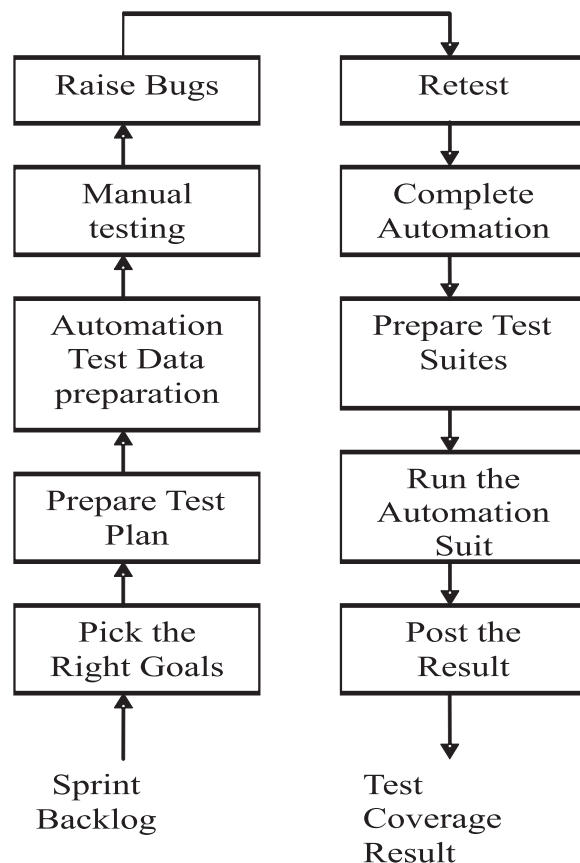


FIG. 2 TEST PROCES

In this process you will be getting some high level document from where you will be designing the test plan, which may tend to change during or after the sprints. Here is an opportunity to estimate the testing effort and using the poker game or any other estimation techniques. While estimating for the testing, the key areas are depending on the test strategy of the client. They are like Manual effort, automation effort, internationalization support, Performance effort etc. During this phase team should have discussions among the team members and with the Business Analyst and understand the feature and based on the technology implications and the feature complexity, the team derive the estimation figures. Once the estimations are accepted by the client the team has to prioritize the requirements and start implementing. Thus one should understand the priorities of the backlog and should be able to deliver a piece of working and shippable code at the end of the cycle.

VI. BEST PRACTICES

From the testing perspective, agile methodology requires that as each component is added, other previous components are regression tested. This is a burden on the testing organization. The final test should be a complete end-to-end system test; some agile teams skip this step, just testing the "latest" code, resulting in severe error in production. To overcome this issue you should have your own strategy of running the acceptance automation on each small release and running the regression suites on frequent intervals. The following are the few best practice one can adopt to see get the benefits of agile scrum

- a. Test Driven Development
- b. Working parallel to development
- c. Implementing continuous integration and automation framework
- d. Following agile release management
- e. Measuring agile metrics
- f. Setting the sprint exit and entry criteria
- g. Setting the bench mark for the velocity measure

- h. Setting the correct and realistic expectations to the stakeholders

During the implementation testers can start the test plan and writing the unit tests, it depends on what kind of tools you are using. Testers can start writing the test on day one which may be failing initially and then pass once the code is implemented which is a true test driven development approach. You can also do the ground work required for the automation before actual piece of code given for the testing and also you may try to finish the remaining testing activities. During the sprint there may be changes from the Business Analyst, we may accept the changes as long as you can accommodate as part of the sprint. If the changes have more impact and complex in nature you need to estimate for the same and add in the backlog as a priority item. Once the graphical user interface is getting ready quality folks can start automating step by step, even quality folks can direct the team to get the code to be delivered in the order of automation and logical priority. At the end of sprint release make sure that all the required automation is covered and passing the cases.

VII. SCRUM TESTING METRICS

At the end of the day metrics speaks about what we delivered to the end users. There are various ways of measuring the same and the following are the few of the agile metrics which really drives the agile model

- a. Velocity
- b. Burn down
- c. Escaped defect rate
- d. Fix quality
- e. Bug find rate
- f. Bug fix rate
- g. Quality stability
- h. Automation coverage percentage
- i. Percentage of additional scope delivered

These metrics mentioned above plays a vital role in getting any agile project successful. Metrics are the true reflection of the projects in progress. Velocity is measured usually planned versus delivered. Percentage of additional scope is essentially what is delivered which was not planned. Few of the other metrics mentioned above are self explanatory and these metrics would vary from organization to organization.

VIII. PROS & CONS

The largest disadvantage we have with Scrum from a quality assurance point of view is that by the process you can code up to the last day and as long as whatever tests you have for the code pass and they are your passing criteria then you are done, this is where you need to define your own process to make sure that whatever is done means it is completed by all means implemented, tested and accepted. In agile usually most of the time the benefits are more than the drawbacks. The benefits varies in from discipline to deliver and few of them are mentioned below

- a. Time to market
- b. Predefined quality
- c. Reduced cycle time for automation suites
- d. Customer delight
- e. Flexible work life

As mentioned above these are the core benefits which are going to be the primary value of the agile model. Few of the drawbacks are mentioned below.

- a. Need highly disciplined team
- b. More ownership
- c. Responsible
- d. Team buy-in
- e. Good judgment and decision skills
- f. Read accept the changes more frequently

As mentioned above when we talk about drawbacks in detail it is all about being responsible and taking ownership while enjoying

the flexibility and freedom of the scrum process. So until and unless we build the trust within team nothing is going work in agile scrum model. It is all about trust and building relation ship and work will happen automatically.

IX. CONCLUSIONS

To conclude, at the end customer will expect a working piece of code with good quality and good automation coverage. In achieving the quality and automation coverage everyone can have their own strategy based on the environment, project, technology used, tools used etc. As a tester you may be the driving force in achieving the goals as you will be using the Test driven development. That means you have to have a test before the code is implemented and you should drive the team to make those tests to pass which essentially leads to a great quality. Also having continuous integration mechanism in place will be helping in getting a clean code. The communication is the only key success factor for an agile project to be able to be a successful one. Setting the right exit criteria and defining the test strategy for a given a sprint would be useful.

ACKNOWLEDGMENT

I sincerely thank Suresh Karthik, transitions manager, UnitedHealth Group, Hyderabad, India, and Dinkar Singh Rathore, Senior Project Manager UnitedHealth Group, Hyderabad, India, for sparing their valuable time and doing in depth review and providing me the valuable feedback.

REFERENCES

- [1] Wauter, R. (2009, 01 22). Sad Day For Microsoft: 5,000 Laid Off, Earnings And Revenues Down.
- [2] [http://en.wikipedia.org/wiki/Scrum_\(development\)](http://en.wikipedia.org/wiki/Scrum_(development))
- [3] <http://www.agilealliance.org>
- [4] <http://www.scrumalliance.org>
- [5] <http://www.scrum.org/>
- [6] Schwaber, Ken (1 February 2004). Agile Project Management with Scrum

- [7] Schwaber, Ken (2007). The Enterprise and Scrum
- [8] Schwaber, Ken; Beedle, Mike (18 February 2002). Agile Software Development with Scrum

AUTHOR'S PROFILE



Koti Reddy Bhavanam PMP, Project Manager, UnitedHealth Group, Hyderabad, Research Scholar, JNTU, Hyderabad – 500072, Andhra Pradesh, India, bkotireddy@gmail.com; +91 967 686 3535. I have about 10 years of experience in IT industry and currently working as a project manager in Unitedhealth Group, Hyderabad. Before joining I have worked for companies like Oracle and Symphony services. I am a PMP certified professional from PMI, PA, USA. I am also six sigma black belt certified by ISI, Hyderabad. I have been handling projects in ERP and Health care domains. I have been handling teams of sizes from 10 to 125. I have a strong background in project management, quality and release management areas of IT industry