RESEARCH ARTICLE OPEN ACCESS

Introduction to Investigation And Utilizing Lean Test Metrics In Agile Software Testing Methodologies

Padmaraj Nidagundi
Faculty of Computer Science and Information Technology, Institute of Applied computer Systems, Riga Technical University, Riga, Latvia.

ABSTRACT
The growth of the software development industry approaches the new development methodologies to deliver the error free software to its end-user fulfilling the business values to product. The growth of tools and technology has brought the automation in the development and software testing process, it has also increased the demand of the fast testing and delivery of the software to end customers. Traditional software development methodologies to trending agile software development trend have brought new philosophy, dimensions, and processes having invested new tools to make process easy. The Agile development (Scrum, XP, FDD, BDD, ATDD, ASD, DSDM, Kanban, Crystal and Lean) process also faces the software testing model crises because of the fast development of life cycles, fast delivery to end users without having appropriate test metrics which make the software testing process slow as well as increase the risk. The analysis of the testing metrics in the software testing process and setting the right lean test metrics help to improve the software quality effectively in agile process.

Keywords: Agile, Software Testing, Software Test Metrics, Software Error.

I. INTRODUCTION

The rapid growth of software use in daily life with electronical devices has brought a new lifestyle flexibilities to all humans. Building software in a fast way and delivering in a fast way got a new dimension when Agile Manifesto established in 2001 and after six years it was re-introduced as agile methodology. The growing market demand of the fast delivery of the software build by build in short iterations deliver the potentially shippable software features. All software team keep focused on the goals for the planned short iteration, a short and tight learning feedback process helps discover and optimize the process & solutions.

Agile methodology has ten core principles, and many specific agile development methods. All focus on the teamwork, collaboration and process agility with limitations throughout the life-cycle of the software development.

The meaning of agile or agility is quick. In software it means a quick focus on short iterations. Software development and testing can be challenging in programming for the [1, 2] complex or porotypes of software development. Agile development has different approaches and all approaches have a unique value proposition. The agile software testing plays a key role in the verification of code and validation of the software component.

The agile software test process is complex due to short time, the software needs to be very well tested in a very short time covering all test cases, modules, unit tests, integration tests, performance and including end user beta tests. Agile brings one

Significant gain in software development, test & delivery speed, but on the other hand it brings the time constraints that affect the software test process.

My core contribution is to show that having well identified agile lean test metrics it is necessary to identify and to apply to agile different development and testing methodologies.

Specifically:

- Scrum, where any one of the team member can act and test the software for the errors in the dedicated sprint time.
- In extreme programming where a developer writes a unit test and later does an acceptance testing.
- Behavior design development, where user stories tested with business requirements.
- In the test-driven development, at first write a failing test and then make the test pass and finally refract the code and follow the first stage.

Iterative and incremental features which drive the development, in which each feature of the software is developed and tested.

II. THE PROBLEM WE TACKLE

In the software development process, software testing brings a key benefit to verify and validate the full software product before giving it to end customers. In the agile testing process there is a chance of (a) repeated testing part of the software that has already been tested (b) without having the clear path in terms of utilizing the human resources and budget, (c) increasing complexity of the software with each release criteria makes the testing

process more intricate (e) and the testing project may be collapsed (f) the lack of the road map increases the risk in the testing process.

Traditional development is the waterfall software development methodology in which the software testing process comes to the focus on the very late stage and we have time a for test plan and build the strategy. In agile where requirements are changing more rapidly according to stakeholder wishes, in a short time testing needs to be carried out in the right way to deliver a good quality software.

In software testing life cycle, software needs to be tested in a different way such as in the case of unit testing, functional testing, performance testing, UI/UX testing..etc. The goal of the testing process is to improve the software quality in each iteration. In this process it's very important to build the list of key metrics list which helps to drive the test process in the right way with a high throughput.

My approach: In agile software testing process it is very important to test metrics needed to be identified as early, clearly and to be divided by test process, in agile where short iteration plays a key role where this approach can bring the significant value to improve the test process.

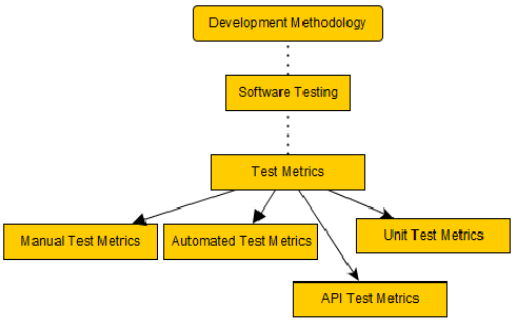

Fig 1: possible way of identifying test metrics in the test process.

III. IDENTIFYING THE RIGHT TEST METRICS

Key metrics work as indicators to measure the performance of the software testing process and assist in the software development process to improve the software quality continuously. In the testing process the right metrics answer all questions to test software for the next stage. In agile it bridges the gap between project goal and actual result.

Step 1: Identifying the result/goal of the project. We must set the goal of products, for example either we test the software product in manual testing or use automation tools.

Step 2: Identifying the impact of the next level. If we use manual testing, what tools, skills, technology and how much human resource we need.

Step 3: From step 2 we have already got basic metrics. They grow and at the same time it is at this stage that we can identify how to divide the project components (break a large system into a small one) in terms of architectural level.

Step 4: Once you identified the software components, now classify the type of testing to create a baseline you want to perform on (functional, regression, smoke, API, security, other)

Step 5: In functional, regression, smoke, API, security, other set the how many test cases you develop, execute and results out of them in agile environment.

Step 6: Establish the review process for these metrics to improve them time to time. The identification of the right test metrics create the loop feedback system that give the continuous input to the test team members to how they are progressing in each stage.

Let's focus on the identifying the most common defect-related terms in agile.

In agile we need to consider all the time how many errors and defects are found in the short cycle. As I have highlighted and showed in figure 1, the test metrics can be further divided into a more sub category on a deep level according to what kind of a test needed to be performed on the software application.

The deep level test metrics may be manual, automation, performance, unit, integration, and usability…etc kind of a test or more than this. As we know the agile development has different methodologies in itself and the goal is the same to first identify the right set of software metrics and the to keep measuring the results.

Table I: identifying the common defect-related terms.

Terminology	Definition	Explanation
Error	Find an error and run time or after completing the product	Software error may be run time or when product built
Defects	Variation at intervals expected and actual result	It's a mismatch of the user requirement
Software Test Metrics	Measure the quality of the project	It is a scale for the measurement.

A. Test metrics formula generation

In software development and testing industry there are different test metrics formula. They are used to evaluate the software quality process, but the goal is very identical - to keep the measurement of the software quality in order to get a deep insight of meaning full data, so use such data to improve the software testing process in each stage and give the right impact.

B. Importance of classifying the right test metrics in agile

The benefit of classifying the right test metrics should be more than what I have identified below here.

- To cover the agile manifesto 10 principles.
- To bring out the deep meaningful data from each testing activity in agile.
- To do the exact type of testing in short iteration.
- To identify' types of defects efficiently in each short iteration
- To have confidence in testing efforts

IV. INVESTIGATION AND UTILIZING LEAN TEST METRICS

The term "lean" got more in the focus when in 1990 Toyota production system used it for a manufacturing plant, also known as a "lean manufacturing management philosophy". The term "lean" means elimination of waste.

Nowadays "lean" is a trending term in software development industry, lean is not applied only in management but also in software development. Most of the time the term "lean" was linked with six sigma principles, which is a combination of reduction of variation willing to solve the process and business problems. Testing is a process, and identifying the right test metrics is a key goal, my focus is more towards on how we can utilize the lean in principle and practice it to get the right test metrics in short iterations of the agile.

The main focused goal of lean is to improve the workflow and eliminate unusual activities. Lean manufacturing aims to eliminate seven different types of waste.

Now we can consider these seven terminologies, compare them and find out how we can adopt them in agile process.

1) Overproduction – in manufacturing and producing the item before it is needed to the customer.
   Overproduction in agile testing - in short iteration is not focusing on what we need to test and spending more time on unnecessary things and over producing such as deep documentation. In reality we need light weight short documents for short iterations. Load testing took place for 1000 users, but according to the requirement in that iteration it is needed only for 10 users..etc.
   Possible test metrics – test strategy for iteration.
2) Waiting – in the production line goods are not moving from one place to another for waiting.
   Waiting in agile testing - agile focused on better collaboration, but still we can notice a tester is waiting for a proper test environment. A developer is waiting for a designer UI for the software application.
   Possible test metrics – response metrics, volume measurements metrics, wait time metrics.
3) Transport – It's a product cost incursion, in reality which add no value to the product.
   Transport in agile testing: spending more time on test meeting, without any agenda or early testing without of scope product features or start testing application when application features are half built.
   Possible test metrics – clearly define a test component for iteration.
4) Over processing – using the complex or not a right process.
   Over processing in agile testing: one must choose the right model for testing: manual testing or automation or mix of both. Which module of application we automate and which not is also important to specify.
   Possible test metrics – test coverage for the iteration.
5) Inventory – work in progress or finished products not well processed directly lead towards the unnecessary inventory.
   Unnecessary inventory in testing –at the development stage testers are doing nothing, just waiting for the product ready to test.
   Possible test metrics – test resource planning for the iteration when a project is big and complex.
6) Overproduction – excess motion or overproduction.
   Overproduction in testing – not utilizing the proper resource or not planned well, so a test team is over testing on the early stage.
   Possible test metrics – proper test planning for the resources, boundary values for every iteration.
7) Defects – errors in a product.
   Defects in testing – we can identify them as defects found in the production by the end customer.
   Possible test metrics – software defects found in each layer of the testing.

While we are identifying the lean seven muda noticed that, if we use these seven muda in a correct way in iteration, we lead towards the new test metrics in agile manifesto ten principles. The investigation of the metrics helps to improve the software quality as well as to increase the software defect traceability in in agile each iterations.

V. SOFTWARE TEST METRICS CLASSIFICATION

In agile we clearly classify the lean test metrics in three category products, processes, and resources. The aim of the classification of the muda (waste) is to identify different possible test metrics in each process, how much accurate we will identify

them that much clear path we get in the agile to utilize them.

Table II: muda (waist) classification and characteristic.

Muda classification	Characteristic
Products	Agile development activities, each iteration deliverables and documentations.
Processes	Agile product delivery to each stage such as test, pre-production and production.
Resources	Agile testing resource planning for hardware, software and people.

VI. CONCLUSION

The study explains that it is a new approach towards adoption of the lean principles in the agile software testing and identifying the more appropriate test metrics to improve the software quality. It's a never ending search for the lean agile different test metrics those drive more value for the agile process itself, further next level research needs to be done.

REFERENCES

[1] Hanssen, G.K.; Yamashita, A.F.; Conradi, R.; Moonen, L. "Maintenance and agile development: Challenges, opportunities and future directions", Software Maintenance, 2009. ICSM 2009. IEEE International Conference on, On page(s): 487 – 490.

[2] Choliz, J.; Vilas, J.; Moreira, J. "Independent Security Testing on Agile Software Development: A Case Study in a Software Company" Pages: 522 - 531, DOI: 10.1109/ARES.2015.79.

[3] Hayata, T.; Jianchao Han; Beheshti, M. Facilitating Agile Software Development with Lean Architecture in the DCI Paradigm, Information Technology: New Generations (ITNG), 2012 Ninth International Conference on Year: 2012 Pages: 343 - 348, DOI: 10.1109/ITNG.2012.1571989.

[4] Karvonen, T.; Rodriguez, P.; Kuvaja, P.; Mikkonen, K.; Oivo, M. Adapting the Lean Enterprise Self-Assessment Tool for the Software Development Domain, Year: 2012. IEEE Conference Publications, Pages: 266 – 273

[5] Korosec, R. ; Instrum. & Test Syst., AVL List GmbH, Graz, Austria ; Pfarrhofer, R. Supporting the Transition to an Agile Test Matrix. Software Testing, Verification and Validation (ICST), 2015 IEEE 8th International Conference, 13-17 April 2015.

[6] E Kupiainen, MV Mäntylä, J Itkonen. Performance Alignment Work: How software developers experience the continuous adaptation of team performance in Lean and Agile environments. Information and Software Technology, 2015 - Elsevier. Volume 64, August 2015, Pages 132–147

[7] H Edison, X Wang, P Abrahamsson. Lean startup: why large software companies should care. XP '15 workshops Scientific Workshop Proceedings of the XP2015. Article No. 2.

[8] P Chandu, T Sasikala. Implementation of regression testing of test case prioritization. Volume 8, Supplementary 8, April 2015.