

AN INDUSTRIAL SURVEY ON THE TEST AUTOMATION CHALLENGES IN THE AGILE SCRUM TEAMS

Kiran Jammalamadaka¹

¹Research Scholar, Computer Science Department, K L University, Andhra Pradesh, India

Abstract

Agile methodology has been picked by many organizations for a decade, while there are many studies on agile development, relatively less focus is on the testing and test automation in agile, most of the organizations are facing a few challenges, and we gave an attempt to understand and consolidate the challenges with a focus on the test automation in agile scrum. We report out the results of an empirical study conducted across the world involving large, medium and small scale organizations' software professionals. We found that there are a few common challenges faced by most of the agile scrum followers across the agile scrum world. This survey will help the researchers to do a deep dive into the practical challenges faced by the agile scrum teams and assist them with the solutions.

Keywords: Industrial Study, Scrum, Agile, Agile Test Automation Challenges, Survey

1. INTRODUCTION

Software testing is an activity aimed at evaluating the program or system to determine whether it is meeting the required results or not. [1], Testing has become an essential activity to boost the confidence as it gives realistic feedback on the system [3], agile testing is a testing practice followed in agile software development [2]. Agile testing is a bit different from traditional testing where testing is a phase and here in Agile, testing goes in parallel with development. Agile testing demands new approaches and ideas to cope with the agile principles where focus is on the rapid delivery with quality. Many researchers and agile followers focused more on the development process and relatively less efforts made towards the testing and automation, being an agile test follower, we have faced many challenges in applying the agile principles and practices to the testing and automation, and would like to ensure our challenges are like the other test professionals in the industry who are also into agile.

We have taken survey approach to gather the inputs from the other software test professionals, in order to get the inputs, created a web based anonymous survey and floated through various mediums.

2. METHODOLOGY

A web based questionnaire [9] has been distributed among industrial software professionals using, e-mails, Facebook and other e communication means

Being test automation professional in agile scrum, we leveraged our experience and formed the questions as a direct statement which helps the responders to understand the point and respond clearly

2.1 Questions

Since Automation in agile is vast subject, we focused on a few issues and tried to get the opinion on these issues alone.

2.2 Responders' Answers

Responders were provided with only three options as many options will lead to confusion and deviate the focus which may result in incorrect responses, so we broadly categorized the responses into Agree, Disagree, neither agree nor Disagree(Sometimes), these limited answers help us to segregate the responses quickly.

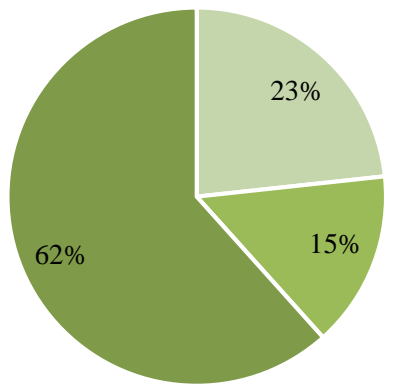
3. DEMOGRAPHICS OF THE SURVEY

We have created a set of 10 relevant question using a web based application called survey monkey. We designed questions with an intent to touch psychological and technical aspects software professionals, we had two parts in the survey. First part which consists two question asking about the size of the responder's organization and whether they are currently using Agile-Scrum. These questions let us know categorize further into small, medium and big scale organization related analytics, however these are out of scope for the study.

First two questions are as follows

1. My Company Size
2. We use Automation in agile->Scrum

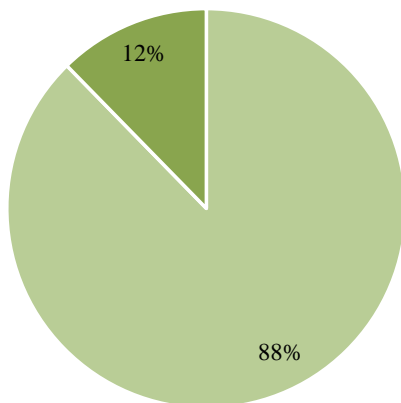
We considered organization's size by its no of employees rather than the revenues. Hence, organizations having less than 1000 employees are considered as small, and between 1000 to 5000 employees are considered as medium and greater than 5000 are big scale organizations. 72 Unique responders from across the world took the survey and 87% are currently agile practitioners, with an overall breakup as follows



- Small(<1000)
- Medium(>1000 and <=5000)
- Large >5000

Chart -1: Company Size

Second question enquires about the whether they are currently using automation in agile -scrum or not.



- YES
- NO

Chart -2: UseAutomation in agile->Scrum

Other than the above two questions, we provided three options to the responders as Agree, neither agree nor disagree (Sometimes) and Disagree these options will help the responders to quickly align to the category.

4. SURVEY QUESTIONS

Below are the questions and a brief introduction to the topic or the area.

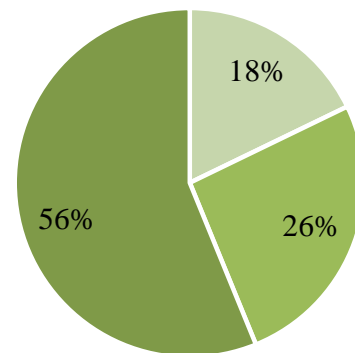
4.1 Within the Sprint, Test Team waits till Developer Delivers the Feature

This is common phenomenon in recently transitioned from waterfall to agile teams, though the practices were fully adapted in terms of having stories, points and acceptance

criteria etc., still teams are thinking in waterfall, i.e. phase wise, there has been a perception about testing, that is testing starts once development is completely done and in most of the teams, the person who is supposed to validate the story against the acceptance criteria won't be understanding the story till he/she hears that development is completes.

And the survey also proves the same, that 56% of responders strongly agreed that they wait till the development completes before starting the testing and 26% agreed that they do wait sometimes, however there is scope for addressing the issue of waterfall in sprints mindset.

Below Pi chart depicts the same.



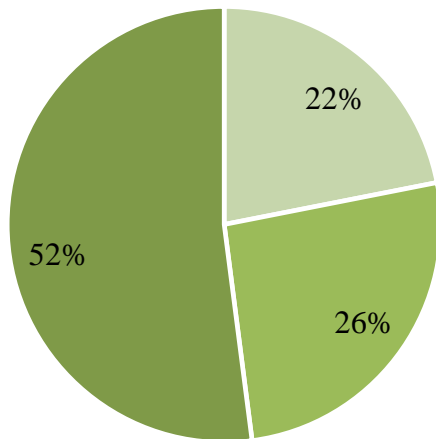
- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

Chart -3: Waterfall like mindset

4.2 Executing Automated Regression Tests on Every Overnight Build is a Major Challenge

As per the survey conducted by Sauce labs [4] on the continuous integration (CI), claims that CI users grown by 7% from 2015 to 2016, 77% of the responders using continuous integration in their organizations, however test teams are still struggling to port and making them run in the Continuous Integration environments like Jenkins or Electric cloud etc.

Most of the responders also expressed a similar view for the above question, 52% strongly agreed that they are struggling, and 26% agreed that sometimes they are also struggling, hence total of 78% responders agreed that it's an issue for them and 22% seems to be happy with their overnight regression test execution. The same has been represented in the below Pi chart.



- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

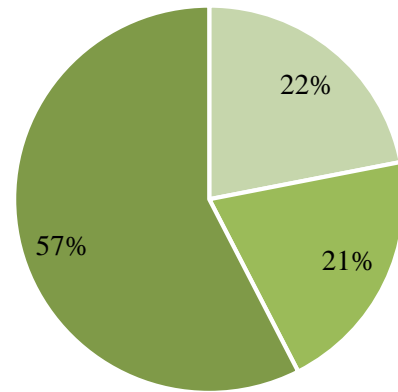
Chart -4: CI/CD maturity

Though it's a one-time job, maintenance is re occurring due to various reasons but not limited to, tests not being robust, virtual machine related issues, network issues etc., this opens a new area to address though may not eradicate all the issues, however can suggest some best practices and aids to the test team to minimize the maintenance.

4.3 Analysis/Debugging of Automated Regression Failure is Time Consuming and Tedious

Having Continuous Integration and Continuous Deployment in place, the frequency of running regression tests has become more frequent, from occasionally to daily and even hourly or even on every check in into the stream. As the number runs increase the number of times result analysis also increases.

This analysis or debugging demands sometime in the daily schedule, in this context responders exhibited the tediousness in the survey, below Pi chart.57% of the responders are agreed that they are considering regression failure analysis is time consuming and tedious and only 16% of the responders are happy and not considering this activity as non-tedious and 22% also feel the same however together 79% responders feel this as a pain area, Hence there is a scope to come up with designs and frameworks to help the test engineers.



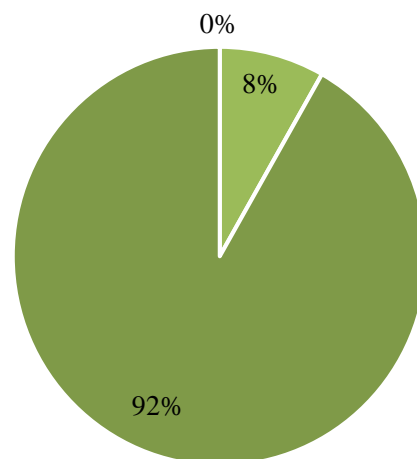
- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

Chart -5: Regression failure analysis

4.4 We Trust our Automation, however it can be improved Further

Most of software quality conscious teams are looking at higher degree of Automation as a solution for their rapidly growing test suits. However, a good amount of distrust [5] has been put into the automated tests and as such these automated tests could not be completely relied upon, the same has been expresses for the below question in the survey 92% of the responders trust their automation and they think that their automation can be improved further, and 8% feels that sometime they feel the same.

The below chart explains the same.



- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

Chart -6: Automation maturity

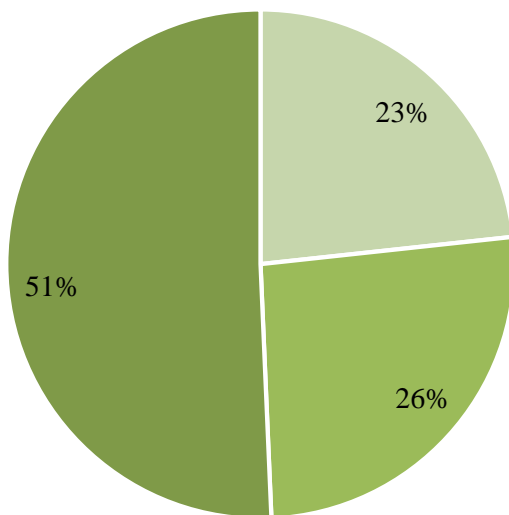
This challenge opens new areas to come up with to address the automation framework measurement and techniques to ensure their automation is matured enough.

4.5 Code refactoring always makes Automation Unstable and Create Maintenance Overhead

Code refactoring is intended to optimize the code without changing the functionality[10], however this scenario would happen in ideal world, it causes many regressions, the same has been expressed by responders.

51% of the responders agreed that code refactoring causes regression and increases the automation failures count and increases the analysis time.

This area needs some practices and frameworks to address this need and ensure a safe code refactoring with minimum or no regressions.



- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

Chart -7: Impact of the code refactoring

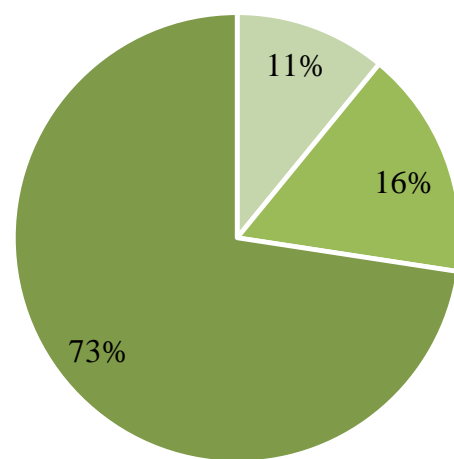
4.6 Test Script Developments along with Feature Development is a Challenge with Evolving/Unstable UI

Approach towards quality is major difference in the sequential and agile models. In agile every sprint on sprint no of test cases get added to the test suite, here the major challenge is to test the new software with in the sprint to ensure no regressions at the same time need to ensure no regressions with the delivered features in previous sprints, this increases more focus on the test automation to be in place and should start adding value to the product by reducing test efforts.

Evolving User Interface is challenge [7] teams are facing in this transformation, Product being incrementally developed, UI also will evolve continuously. This puts pressure on automation being developed or modified for changing UI. Often, test development must start before UI components are available.

Responders also expressed in sprint automation is a major challenge for them, 73% of the responders agreed that in sprint automation is a challenge due to evolving UI, along with feature development. And only 11% are comfortable with the parallel test script development along with the feature.

To cater these kinds of needs in agile, automation teams should come with a novel approach which can make the tests ready the moment UI is ready and best practices to achieve in sprint automation.



- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

Chart -8: Script development without User Interface

4.7 Unit Tests, Integration Tests and UI Tests are Essential, however we did not consider the Test Redundancy within these

Unit, Integration and UI are tests are the most common tests any team has, and the has been explained in Test Automation Pyramid by Mike Chon [8], in his book "Succeeding with Agile", and he also suggested that to focus more on the Middle layer which is a service layer, below Figure 1. Depicts that.

User Interface (UI) tests stay top and Unit tests will be at the bottom of the pyramid and Service tests will be in between user Interface and unit tests.

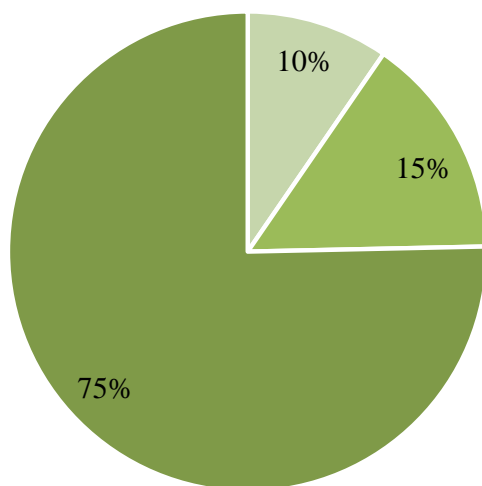


Fig 1: Test Automation Pyramid.

Since this pyramid is a very popular and most of the teams adopted this approach and what the missing piece from this pyramid is duplication of the tests, some of the functions or same lines of code had been executed by the user Interface tests, Service sets and Unit tests, it may differ from the data point of view most of the times, the same code get executed and we see a redundant testing.

One reason could be UI tests get developed by test engineers and other could be Test developers or by the developer himself, irrespective of that there is a need to reduce the redundant testing within these three types of tests, keeping this in mind floated a survey question, whether the teams are considering the duplication of the testing, this indirectly hints about the coordination between the test engineers and developers

The below Pie chart explains about the test redundancy, as per the results, 75% of the responders are not considering the test duplication, which could be a major area to work on and suggest some designs or solutions to save the test duplication, and only 10% of the responders are considering test duplication, in their development



- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

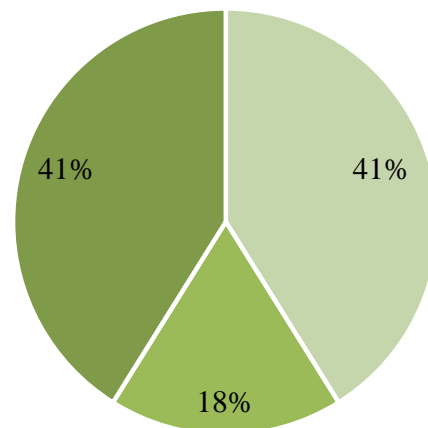
Chart -9: Test redundancy

4.8 Success of Test Team is measured by no. of Defects found in House

The no of defects found in house was used to be a performance metric once, where testing focus on the detection. However, in agile the focus is on the defect prevention and which cannot be measured directly. This survey question is intended to understand whether agile teams still consider no of defects logged before release to calculate as metric to monitor the metrics like defect density or defect removal rate [6] etc.

Half of the responders were agreed that their management still considers to determine the test teams' success by no of defects detected prior to release. To be precise 41% of the respondents agreed that their team consider no of defects logged in house as a performance metric, and another 41% did not agree to this statement.

The below Pie chart explains the same.



- Disagree
- Neither Agree nor Disagree (Some times)
- Agree

Chart -10: no of Defects as metric

Half of the responders, disagreed that defect count is no more a metric to measure teams' performance or success, which should be the case with agile teams. However, it is very difficult to count how many defects have been prevented by the teams. Hence, there is a need to re-look at such traditional metrics in agile.

4.9 Summary

Here is the snapshot of the overall survey leaving the demographic questions with actual numbers, as 72 unique responders took the survey across the world with the actual number are mentioned below for number for reference [11].

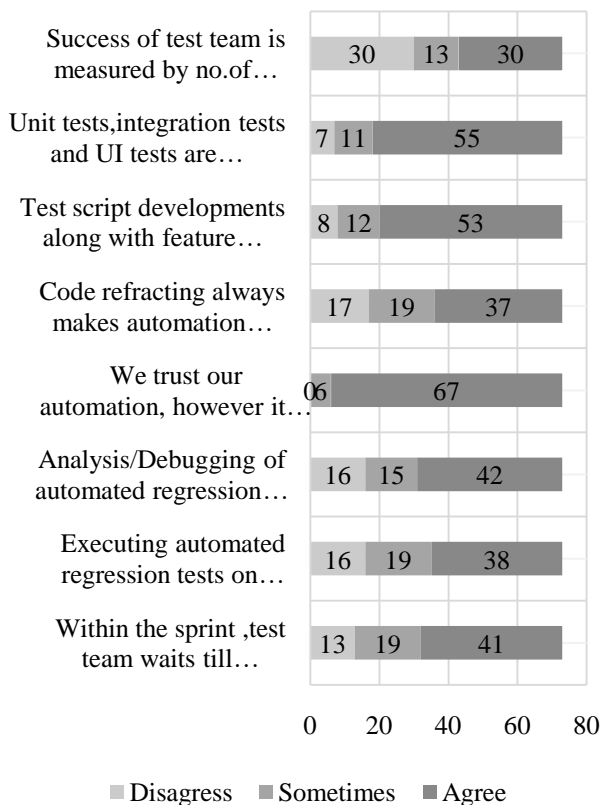


Chart -11: Summary of the responses

5. CONCLUSION

In this paper, we made an attempt to seek opinions on the contemporary industrial aspects on the test automation in agile software development, the uniqueness of our survey is we prepared direct questions to the responders which is easily understandable by the responders and helps us having a good insight. These findings help us in understanding challenges faced by the agile scrum teams during the adoption and transformation to agile methodologies, areas of improvements. And, assert our assumptions and beliefs.

Our ongoing research work attempts to address these challenges by proposing frameworks, design approaches and other best practices. We are also working on a model to quantify and improve the maturity of the existing automation framework of the teams.

Our survey results say that team's maturity on the Agile transformation is at intermediate state and needs improvement with test automation in Agile scrum.

ACKNOWLEDGEMENT

I would like to thank Devicharan Garigipati for his valuable insights and support.

REFERENCES

[1]. Hetzel, William C., "The Complete Guide to Software Testing", 2nd edition. Publication info: Wellesley, Mass.: QED Information Sciences, 1988. ISBN: 0894352423

[2]. https://en.wikipedia.org/wiki/Agile_testing

[3]. Bardolino, Antonia, "Software Testing Research: Achievements, Challenges, Dreams," Future of Software Engineering, 2007. FOSE '07, pp.85-103, May 2007, IEEE.

[4]. <https://saucelabs.com/resources/white.../sauce-labs-state-of-testing-report-2016.pdf>

[5]. <http://luxagile.blogspot.sg/2007/05/five-obstacles-on-way-to-successful.html>

[6]. <https://www.equinox.co.nz/blog/software-testing-metrics-defect-removal-efficiency>

[7]. Adam Geras, Of Schools and Tools, An Agile Tester 's Resource Kit. ACM, 2009

[8]. Cohn, M., & Safari Books Online (Firm). (2010). succeeding with agile: Software development using Scrum. Upper Saddle River, N.J.: Addison-Wesley.

[9]. <https://www.surveymonkey.com/>

[10]. https://en.wikipedia.org/wiki/Code_refactoring

[11]. <https://www.surveymonkey.com/r/3N3VXF5>(Currently closed)

BIOGRAPHIE



Kiran Jammalamadaka received his MCA from Acharya Nagarjuna University, India. He is currently associated with GE India as a Staff Quality Assurance Engineer. And also perusing his PhD from K L University. His interests include Software engineering,

Agile->Scrum and automated debugging.