

Effectiveness of Agile Test Planning Over Traditional Methods: A Software Development Perspective

H.U.W. De Silva and C. D. Manawadu

CAKE LABS Private Limited, Colombo, Sri Lanka

ABSTRACT

Technology innovations drive the global economy, and the software product engineering industry plays a primary role. The impact of software failure in today's business world is intolerable. Hence, quality plays a vital role. The industry's expectation for the software product engineering industry to mature and thrive with higher quality and development productivity is increasing daily. The test planning is the place to begin. In the agile context, the way test planning happens and its importance are left as a void in the interest of researchers. One of the main referral document which is being widely used in the industry is IEEE 829 standard test plan guidelines; but with the agile approach, most of the concepts discussed in this document cannot be adopted in the agile context due to practical shortcomings. What traditional features can be adopted by the agile approach in an optimized method will also be discussed in the paper. A cross comparison of case studies and pilot studies will be used to derive results. This paper will clear paths for agile practitioners to adopt test planning in an agile context and will encourage more research to be carried out in similar interests. The results of this study would benefit the current and future software product engineering practitioners, the industry as a whole and the students and academics of the tertiary education system.

1.0 INTRODUCTION

Nature of software development has been through some drastic changes in the recent past. The development methods have been shifted from waterfall to Agile; requirements gathering has been

more progressive and client oriented and testing has also been hit by this wave which has caused some significant changes in the Quality Assurance process [1]. Before analyzing the subjected title in detail, defining quality assurance would allow the reader to gain more insights on the discusses topic. One major misconception in the industry of information technology is that majority refers quality assurance as testing; but, testing is not Quality Assurance. Quality assurance in the industry of information technology is a vast area of knowledge which researchers find a lot of un-addressed voids. One way to identify testing is that; "a testing process would not only remove some of the risks, but also make the quality visible" [2] which means that testing is one way of assuring the quality of the product. Except testing, there can be other ways of reaching different levels of quality but as of now industry as accepted testing as the mostly adhered method of assuring quality in a software product.

Testing can also have various stages, levels or perspectives. For example, static, dynamic, structural-based, experience-based and specification based are few of them [3]. In the meantime there can be various different approaches which are used by software development teams to achieve aforementioned types of test completions; which will include, informal reviews, walk-through, inspections, error guessing, exploratory testing, equivalence partitioning, boundary value analysis, use cases, etc. interesting part of the analysis is that in an ideal plot, a team should be able to accommodate all these types of testing which will eventually deliver a bug free product to the customer [3]. In contrast to the theory, the actual business environment is much more competitive and short-termed for a

team to have all these test techniques in place and deliver the product to the customer. Hence, most of the teams have lately been looking for a mechanism to consider the best combination of these techniques in terms of achieving the higher level of quality. Finding the said combination of test techniques has left a challenging task for team leads and managers. In parallel to selecting the best testing combination, it will have to make sure that none of the components of the product is not left untested; where the real challenge lies in the team management to come up with a proper test plan. A Test Plan will define what and which techniques to be used during the testing phase of the product lifecycle which will reach a maximum level of test coverage, these combinations may differ from product to product and the severity of the software product [4].

Apart from the test plan which will create the test technique combination, team lead, managers will now have to take the software development mechanism into account. Agile has now become a widely used mechanism of software development which has left behind the waterfall practices completely. With the waterfall approach, teams only had to be focused on a single test effort and a single testing phase. However, with the implementation of agile practices, testing has become a recurring phase within the team for the same product. In the usual approach, teams made sure had a solid test design which was designed during the inception of the project which was not capable of adopting changes in the midterm. In contrast, Agile techniques require teams to be more responsive and progressive; which only demands very less documentation. Responding to change is one of the values in Agile mechanism, and pre-approved test plans will not complement this value; hence, teams will now have to have progressive test plans to support continuously changing requirements [4].

With the induction of agile practices, a question arises; what standards have teams followed previously for test planning? IEEE 829 has been the IEEE standard for test planning which has been used over the last decade. It is important to identify the components of IEEE 829 draft in comparison

with the agile manifesto to make sure that traditional test plan can be used with the current agile practices.

This paper is intended to propose a new template to be used in an agile context for test planning. This research set forth two-fold research questions;

1. How can agile test planning be used to increase team performance in an agile context?
2. What attributes of IEEE 829 can be adopted in agile test planning to improve the team velocity?

Further to the discussion, this paper will illustrate the traditional test plan comparison in detail in following sections.

2.0 LITERATURE REVIEW

Studies indicate that testing accounts for 50% of the total development time. One reason for this is that the verification activities late in development projects tend to be loaded with defects that could have been prevented or at least removed earlier (when they are cheaper to find and remove) [5]. Identifying the issues at the earlier stages has been the best approach in minimising the overhead cost related to the quality aspect of the product. Whenever the bugs have been identified at the late stage of the development cycle the fixing cost will rise making the entire product less profitable or which will end up in a loss. As it has been pointed out by Lars-Ola Damm in the about extraction, the cost related to the quality will have an impact on the whole product. Over the years, professionals have come up with various different ideas and approaches in the making the software development projects more profitable by cutting down the overheads related to the same. Unit testing, Test based development, Test driven development are some of those methods which are in place to make sure that a maximum number of bugs have been identified at the early stages of the project [3]. In the perspective of the project management, SDLC has very had very less space and capabilities of accommodating these fancy methodologies which have been introduced in the recent past. Agile software development is one of the modern hybrid approaches and a substitute for SDLC which has the higher calibre in accommodating new development approaches.

According to Barsei, the quality process involves many activities that can be grouped into five main classes: planning and monitoring, verification of specifications, test case generation, test case execution and process improvements [6]. With all these activities in place, an unplanned QA process will end up in the chaotic situation without a doubt. To mitigate problems such as activity precedence, activity coverage and completion of test executions will demand a comprehensive test plan. A test plan will define the boundaries of the test effort and the time estimations and any risks which might be involved. Barsei further defines that planning activities has to start in early phases of development with the identification of the required qualities and the identification of an early analysis and test plan, and continue through the whole development by monitoring the quality process and by refining and adjusting it to take care of new problems and avoid that deviations from the original plan to project failure. A test plan can be defined in as a document which includes test plans and test strategies that summarise the quality strategies of the company or the group [6].

IEEE 829 has been the certified test plan format which was used by many teams in the past. This approach was very well in line with the expectations of the waterfall development model. In the waterfall approach, continuous development, progressive improvements were never a significant component. But with the rising trend of agility, most of the companies are now going through the paradigm shift from waterfall development to agile software development interesting conversation starts when not only the companies are keen on agile developments but most of the client demand for agile development methods due to man more good aspects of the agile approach. The agile manifesto defines Individuals and interactions over processes and tools, working software over comprehensive documentation, Customer collaboration over contract negotiation and Responding to change over following a plan are the four values of Agile methodology. The agile manifesto will encourage all the documentations, activities and processors to have adhered to the values and principles. Hence, the current test plan which is being used for test planning is questionable as for its validity against the agile practices. One other important factor is that the test plan document which was used in waterfall development will not support continuous improvements which is one of the key aspects of agile development.

In addition to the discussed agile values; 12 more principles will also define the agile community.

Number	Principle
AP1	Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
AP2	Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
AP3	Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
AP4	Business people and developers must work together daily throughout the project.
AP5	Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
AP6	The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
AP7	Working software is the primary measure of progress.
AP8	Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
AP9	Continuous attention to technical excellence and good design enhances agility.
AP10	Simplicity--the art of maximizing the amount of work not done--is essential.
AP11	The best architectures, requirements, and designs emerge from self-organizing teams.
AP12	At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behaviour accordingly.

Table 1: Agile principles

The documents which are being used in an agile context will have to adhere to these 12 values and four principals. James Bach's heuristic test strategy model is something which has grabbed the attention of many professionals. This paper discusses how to merge the heuristic model of James Bach's with the IEEE 829 test format to design a new test plan document which will adhere to the principles and values of the agile manifesto.

The heuristic model discusses 7 new elements which can be added to the traditional test plan. Structure, Function, Data, Interfaces, Platform, Operations and Time are the seven new elements which will bring in the essence of agility into the traditional software test plan. James Bach's heuristic model has three main components as Project environment, Quality criteria and Product elements. With all these implementations it is vital

that the mindset of all the employees are prepared to embrace the importance and qualities of agile practices [7]. Out of the three components of the aforementioned model, the product elements segment will be taken into the discussion as this section of the model will define the changing specifications of the client. This is illustrated in Figure 1.

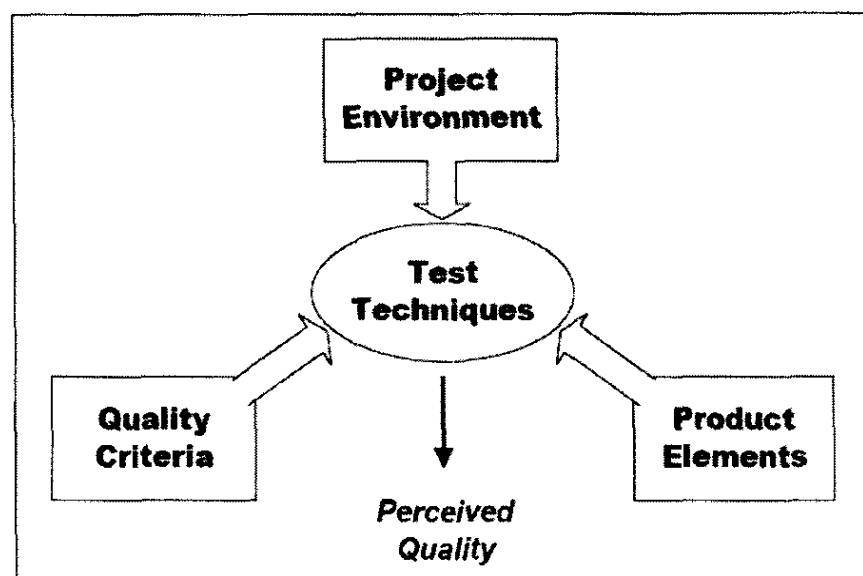


Figure 1: Product elements in the heuristics model

According to James Bach, ultimately a product is an experience or solution provided to a customer. The product has many dimensions. Each category, listed below will define a critical aspect of the software and testers who doesn't cover these components in a test plan will tend to miss many important bugs.

1. Structure – All the physical aspects of the product

Under this section test plan will what areas of tests has to be executed and the coverage of the physical components of the product. For a given product there can be external hardware devices which will provide input into the main data stream. In addition, the code structures of the product, non-executable files and any collateral files which are related to the product will be brought to the attention. Not all the companies in the IT industry will make Software, but there can be SMEs which may be involved in hardware integration, maintainable and etc [7]. A common test plan should have the flexibility of covering the all the aspects of the industry which are why the hardware component is also added to the test plan.

2. Function - the business aspect of the product – product capabilities

Functional capabilities of the product will be taken into the consideration in this section of the test plan. For example, calculations, time-related activities, data flows, transformations, start-ups and shutdowns can be considered. The function unit can also be divided into minor sections where the focus of the functionalities will be inspected at a micro level. Domann mentions in his paper that best agile teams will focus their tests at unit level or rather at the functionality level till they reach the level of regression or integration testing [5]. In the sense of testing at a functional level, it is important to be very clear on the test scope in the test plan so that none of the critical functionalities will be missed by the testers.

3. Data – all kinds of inputs which will be taken into the system

The types of data which can be inputted into the system will be analysed in this phase. The data formation, test data categorization and test data design techniques will be discussed under this phase of the test plan.

4. Interfaces – gateways which product can be accessed

System interfaces, APIs, SDKs and user interfaces will be tested under interfaces. What modes of communication do the product use and what type communication channels are being exposed to third parties will be tested. According to Mousumi, the continuous improvements and developments have to initiate with the client requirements; hence, interfaces are the main source of communication and the quality has to be at a superior level in these components.

5. Platform – product dependable

Any external hardware or software which are being in the system will be considered. What kind of interaction do they have and to what degree the system incorporates these third party tools will be analysed and tested. Not only the external components but the internal components will also be accountable under this section of the test plan.

6. Operations – how the product will be used in the business environment

The user roles which will be involved in the system and the various user privileges and extreme usage scenarios will fall into this category. It will make sure that the edge usage test cases and scenarios will cover the boundaries of the system while testing with different user roles and privileges.

7. Time – the relationship of the product with the time components

Any relevance of the system's interoperability with the time factor is subjected to test here. The timely processors, time zone operations and etc, are a couple of such examples.

Discussed heuristic model now can be merged with the existing IEEE 829 framework. Following is a list of components which are available in a traditional test plan document [7].

1	Test Plan Identifier
2	References
3	Introduction
4	Test Items
5	Software Risk Issues
6	Features to be Tested
7	Features not to be Tested
8	Approach
9	Item Pass/Fail Criteria
10	Suspension Criteria and Resumption Requirements
11	Test Deliverables
12	Remaining Test Tasks
13	Environmental Needs
14	Staffing and Training Needs
15	Responsibilities
16	Schedule
17	Planning Risks and Contingencies
18	Approvals

Table 2: IEEE 829 Test plan

the problem with the IEE format is it doesn't allow the test team to dig deep into the product functionalities and carry out a detailed design

approach; instead, it will encourage the team to have a common test strategy which will be applied across the product development lifecycle. In contrast to the waterfall model, agile will demand a changing test plan which has to update in and every iteration of the project. When continuous updates and improvements are needed, the most wanted components will have to available in the test plan as updating less important fields in the plan will be an overhead to the product team. [8].

3.0 RESEARCH METHODOLOGY

The research team has taken the approach of using multiple case studies to analyse the results of the proposed test plan document. Three case studies have been chosen from the information technology industry who have been following agile practices for more than six months. Theses teams have a proven record of good agile practices and each team consist a maximum of 8 members.

Main reason behinds selecting case studies are that the researcher will then be able to compare the results and build a constructive argument by exploring the behavioural references between the different case studies [9]. The goal is to replicate the results which were expected by introducing the new test plan. It is important to note that cases have been chosen very carefully to make sure that similar behavioural patterns can be expected via the selected teams. However, it is difficult to expect the ideal outcome from each case but the aggregated results can be used for final analysis and model interpretation.

In order to proceed, authors have selected three different case studies from a reputed Silicon Valley based company who run their engineering centre in Sri Lanka, Colombo. This company is a strong believer in agile practices and all the teams inside company practice agile development methods. As it was mentioned three teams who deliver software components which will contribute to a single product are subjected to this study. Each team had their own problems related to the quality of the delivered software components.

Case 1: The main product team of the company who are in the exercise of developing and advancing the existing product. The product focuses on small and medium scale restaurants which will enable them to automate the entire business process from customer relationship management to stock management and financial

management. The product is already having a substantial client base in the United States of America. The team consists of 10 developers and 3 QA engineers which are being supervised by a QA manager. In this context, the manager is responsible for creating the test plan with the help of the other QA resources in the team.

The team selects what features to be implemented in a sprint and within the sprint, the quality assurance process will also be covered. After completing the feature level testing the, the component will be going through a regression test phase which has a 90% of automated test cases while rest of the regression test cases are executed manually by a team resource. Upon completion of the regression, the integration test will take place which will ensure that the new component can be integrated with the existing product without any major barriers. However, when the integration is completed the software will be pushed into a different environment which will be tested in a production scenario which is called to be Pre-Production testing. During Pre-Production testing, the production like environment and real time user scenarios are tested. Each team will have their own test plan in relevance with the shared specification.

Case 2: The reports team is another significant team in the context which has taken into our analysis. The reports team comprises of 7 members with 5 developers and 2 QA engineers. The business operation of the team is that they will be working on report improvements which are plugin tool for the main product. Reports will be communicating with outside systems as well enabling financial analysis and ERP support for the merchants. Hence, this system will have external communication channels and open APIs which could be used by a third party teams. Reports team follows a similar process as the product team but has few minor differences. They will have their own format of test plan and the testing process will adhere to agile practices. Upon receiving a feature, reports team will develop up to 90% of the functionalities and a QA resource will do an unofficial smoke to make sure that the major data flows are working fine. Upon confirmation of the basic data flow validations, the QA release is deployed in a different environment for testing.

When feature testing is completed, a release will be handed over to the pre-production testing team and eventually it will be pushed into production after staging certification. Reports department is one of the most critical components in the system as it

incorporates all the daily transactions into one document which will be referred by many other third party stakeholders. Hence, the it is treated as a higher priority deployment whenever a report deployment is required.

Case 3: The other department which will be used in the case study analysis is into data processing and date formulating. The data processed by the component will be used across the product and will be visible to all the stakeholders including the end users. The release plan for this department is more frequent and with short sprints working pieces of software will be released. This team has got a 40% QA allocation and the test plans and tests strategies will be frequently changing in parallel to different sprints. The specifications are not clear cut for this team while the criticality of the end product is much higher. This will make is challenging to come up with a common strategy and a stable test plan but rather it will be often changing and will be subjected to continuous improvements.

The subjected case will create a common data platform for the entire spectrum of the product range, any malfunction in this component will cause a breakdown in the entire product which will eventually have an impact on the complete range of merchants. Therefore, the test plan is expected to be very comprehensive and detailed in terms of data range. The most challenging section in this department is that when the releases are very short planned, the identification of bugs and retesting will have to be done within a very short period.

When all the three cases are addressed in an accumulated approach, few common challenges can be identified. Even though all the teams have been adhering to the agile practices, the test planning and execution has not been updated accordingly. All these teams have been following the traditional approach of test planning and many significant issues have been found at the pre-production test level which is the affidavit test level before production. Identification of bugs at this stage is very costly and the release will be pulled back by the respective team. This will not only consume more time but the architectural changes will cause more uncertainty and customer dissatisfaction.

Furthermore, into the discussion, the mental effect which this rejection will cause on the team is important to be handled in a smart way so that the team will not be de-motivated. Continuous

rejections and bugs will create a bad image on the team as well and it will end up in lower team performance and higher employee turnover. It was later identified that main cause for this leakage of issues is not the incompetence of employees but the existence of unidentified functional areas during the planning phase.

4.0 PROPOSED SOLUTION

The proposed solution for the discussed matter was that new test plan template was introduced which can be used in an agile context. The new test plan will cover a wide spectrum of functional areas which were not brought into the light with the previous approach. Following is a template of the newly proposed test plan. This document comprises of both IEEE 829 features as well as features which have been extracted from the heuristic test model.

1	Feature/ Release overview
2	Testing Scope
2.1	<i>Functionality – in scope</i>
2.2	<i>Functionality – out of scope</i>
2.3	<i>Impacted components</i>
3	Test approach
3.1	<i>Special product elements</i>
3.2	<i>Testing techniques</i>
3.3	<i>Test levels</i>
3.4	<i>Defect management</i>
3.5	<i>Test tools</i>
3.6	<i>Test environment</i>
3.7	<i>Test data</i>
3.8	<i>Platforms</i>
3.9	<i>Continuous integration</i>
4	Entry and Exit criteria
5	Functional test automation
6	Performance testing
7	Test deliverables and responsibility
8	Test matrices
9	Project testing and Risk assessment
10	Assumptions
11	Test Scenarios

Table 3: Proposed test plan

With this approach it is forced that the team pay enough attention to the areas which normally that could be missed by traditional test planning. The heuristic model is incorporated in 3.1 section in the above test plan where the structure, functions, data, interfaces, operations, time and platform were taken into the consideration. The new sections which have been introduced in the aforementioned test plan are from the heuristic model. The traditional test plan always had the features of waterfall development architecture. But the heuristic model will bring in a new scope of agility to the test plan. All the sections which have been discussed in the new test plan will help the team to discover a wider spectrum of the productscope and potential errorness areas. The introduced test plan isn't an independent test plan which will deffer from team to team but rather will be an inherited test plan from a parent testplan which is being formulated at the organizational stratergy level. For example, entry exit criteria, test deliverables, test automation scope, and test matrices will be defined at an abstract company level. Rest of the sections in the plan will accomdate the projecct specific details. For instance, features will specify what features will be tested in the sprint while test-approach will define the selected test strategies, defect management and other related details.

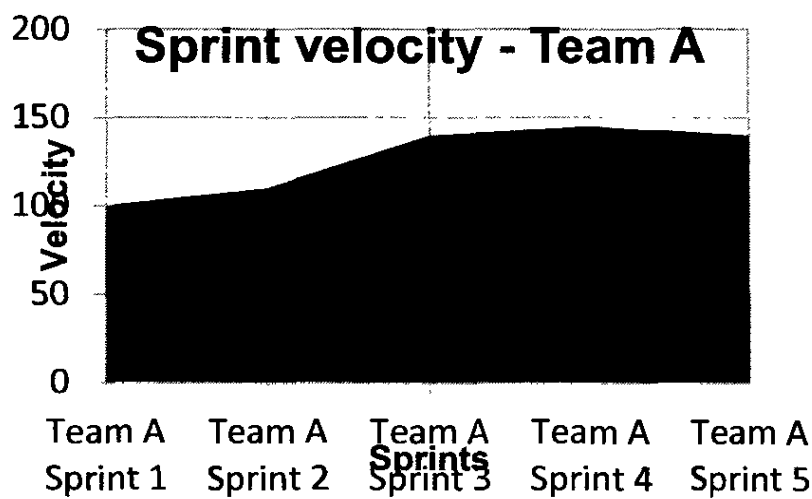
Out of the eleven listed items, there can be components which may not be applicable for every sprint. In a sprint where performance testing is not involved, that section can be left aside by the test designer/ planner. Having these components in the template doesn't mean it is mandatory to all these fields to be filled.

5.0 RESULT

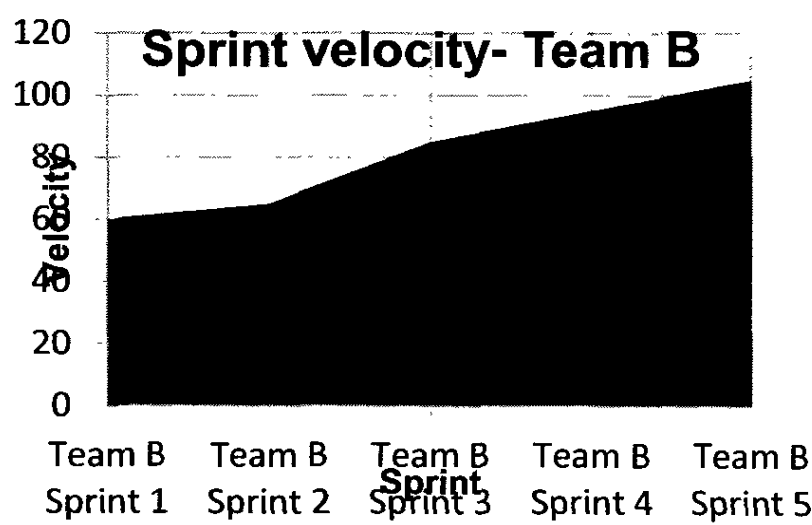
With the implementation of the new test plan, a significant uplift in the process can be observed. The results are more similar to all the use cases. During the initial planning session of the sprint, the test plan will be designed with the input of all the members of the team. This is no longer a task of the QA resources in the team but everyone else's contribution will also be noted in the same. Different dimensions of the product will be bough into light with the new approach and heuristic model will make sure that all corners of the product will be uncovered in the test plan.

In the context of agility, one of the main measuring yards for team performance is the velocity of the team, with the introduction of the new test plan,

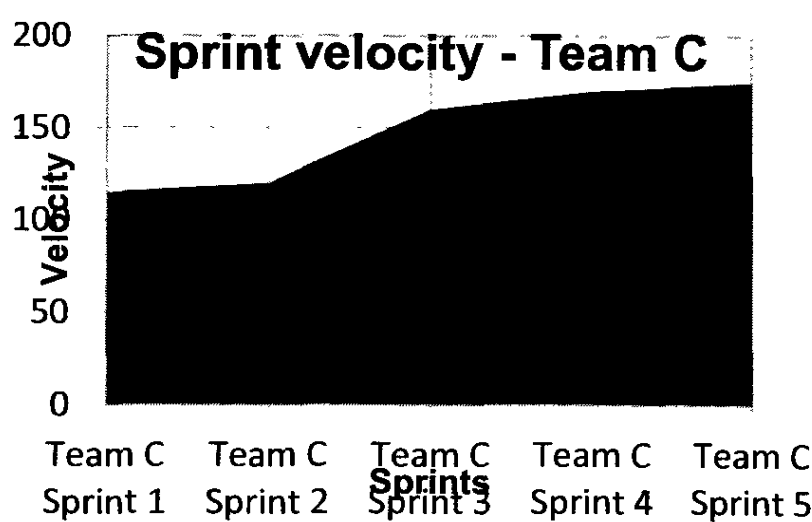
researchers managed to closely observe the changes of the velocity of each team. It was interesting to see the improvement of the velocity. Each team had an improvement of its velocity. The important part in this discussion is to identify how the test plan helped in improving the team velocity. When the issues were found at the early stages of the project, it reduces rework which will complement the team effort to take more story points to the sprint and complete them as well. This has improved the velocity of team which enables faster delivery and team member satisfaction.



Graph 1: Velocity Team A



Graph 2: Velocity Team B



Graph 3: Velocity Team C

One of the main favourable outcomes which we determined was that this helps the team to identify the conspiracies even before formal test cycle starts. This model will uncover design errors, data mismatches and even performance and UI issues of the system. With the earlier approach, it was just the usual test plan with entry and exit criteria. But the new template will define test techniques, test design methodologies, test areas, test data and Above graphs will signify the growth of the velocity of the each team over five sprints. The velocities have been increased and they have got a consistent growth as well.

Individuals and interactions
Working software
Customer collaboration
Responding to change

Table 4: Agile values

The new test plan will bring the entire team together in test designing, test execution and even in quality management. The isolated test teams will no longer be appreciated in the team and the human factor will always be appreciated. The test plan is with very limited data but with the most information. Less documentation is encouraged with the new test plan. The company will have a parent test plan through which rest of the test plans will be inherited to other departments of the company. With the induction of the heuristic model, the team will have to pay more attention to the functional requirements which encourage the team to get the support and the input of the client. Since that the test plan will not be based on wage assumptions, yet it will be more realistic and client involving. Finally, the maintenance of the plan is as not complicated as the traditional approach. Traditional approach with the waterfall model is difficult to be changed when the project has moved to the final stage of the development. But since the entire team is involved in the process test plan and test design process, changes are easy to be accommodated as the entire team is aware of the existing plan and any circumstances which may come along after the change.

6.0 CONCLUSION

The software teams which practice agile methodologies and upcoming teams can be benefited with the proposed approach. With this study, it will open voids for researches to study about the involvement of project environment and quality criteria to the proposed test plan while keeping it maintainable and simple. It is interesting to see a team have been performing well efficiently with the proposed test plan.

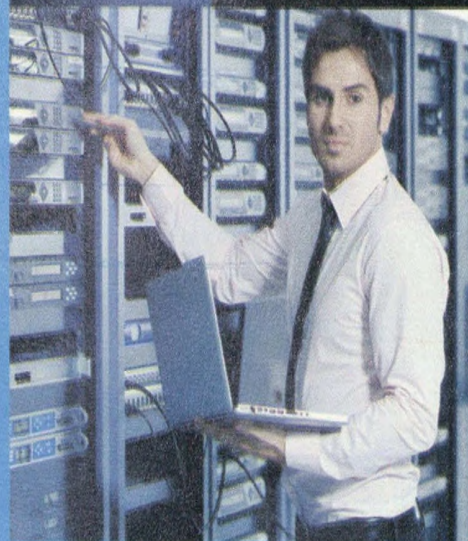
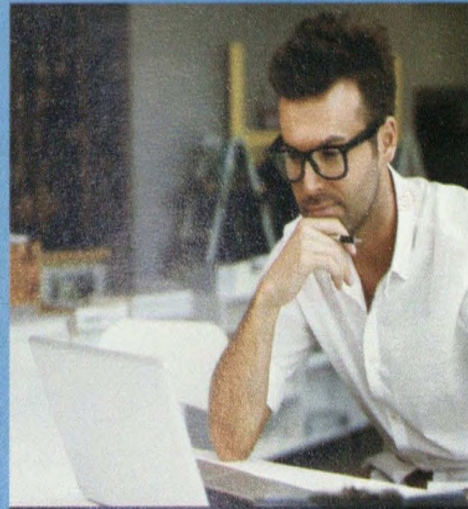
7.0 REFERENCES

- [1] Asnawi, A. L., Gravell, A. M., & Wills, G. B. (2012). Emergence of Agile Methods: Perceptions from Software Practitioners in Malaysia. IEEE, 30 - 39.
- [2] "Introducing Test Automation and Test-Driven Development: An Experience Report", Electronic Notes in Theoretical Computer Science, vol. 1162005315, 2005.
- [3] Pressman, R. (2014). Software Engineering: A Practitioner's Approach. New York: McGraw-Hill.
- [4] Ahalt, S., Band, L., Christopherson, L., Idaszak, R., & Lenhardt, C. (2014). Water Science Software Institute: Agile and Open Source Scientific Software Development. Computing in Science & Engineering - IEEE, pp. 18 - 26.
- [5] "An Introduction to Software Testing", Electronic Notes in Theoretical Computer Science, 2006.
- [6] "Moving from traditional to agile software development methodologies also on large, distributed projects.", International Conference on Strategic Innovative Marketing, 2014.
- [7] "Empirical Analysis of Quality Models in Practice in Small IT Companies in SEE Region", WCES, 2014.
- [8] "An Agile Method for Multiagent Software Engineering", The 1st International Workshop on Developing and Applying Agent Frameworks (DAAF), 2012.
- [9] Rodríguez, P., Partanen, J., Kuvaja, P., & Oivo, M. (2014). Combining Lean Thinking and Agile Methods for Software Development A Case Study of a Finnish Provider of Wireless Embedded Systems. 47th Hawaii International Conference on System Science. Hawaii: IEEE.

AREN'T YOU A
CSSL?
MEMBER YET ?

JOIN THE
PRESTIGIOUS ICT
PROFESSIONAL BODY
WITH A HISTORY OF 4 DECADES

APPLY ONLINE
WWW.CSSL.LK



— SINCE 1976 —

Computer Society of Sri Lanka

275/75 Prof. Stanley Wijesundara Mawatha,
Colombo 07, Sri Lanka

Telephone : +94 11 4713290, +94 114713336,

Fax : +94112508009, Email : info@cssl.lk, Web : www.cssl.lk

ayubo.lk

Your one stop travel partner

proudly 100% Sri Lankan

HOTELS

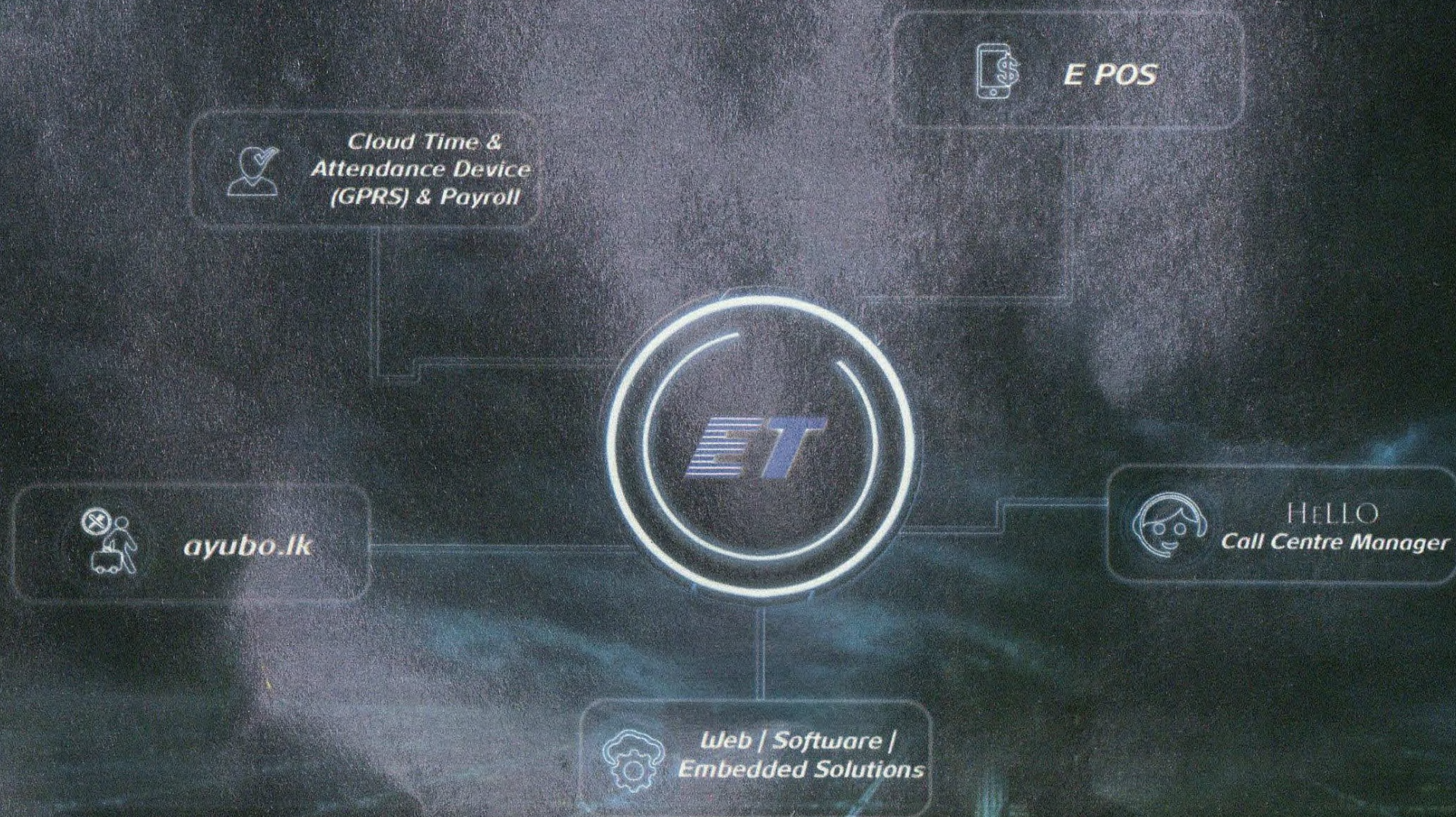
ACTIVITIES

TRANSPORT




Ayubo Leisure (Pvt) Ltd.
No. 182, Galle Road,
Colombo 04.


Tel - 0112 553 899
0772 834 561
0773 099 290



 **ESOFT Technologies®**
— doing it right ...

#182, Galle Road, Colombo 04

 +94 112553899

@hello@et.lk |  www.et.lk



DISCOVER

YOUR FUTURE @

SLIIT



SLIIT

COMPUTING | BUSINESS | ENGINEERING

COMPUTING

BUSINESS

ENGINEERING

ARCHITECTURE

QUANTITY SURVEYING



www.sliit.lk



info@sliit.lk



011 7544801



What would you do with
the power to do more?

"We would set up our own
design studio."

Dell's versatile range of notebooks help you do more,
so that you have time for the things that truly matter.



1717 FOLLOW US
www.mobitel.lk



INTERNET SPEED THAT MATCHES THE SPEED OF LIFE

Mobitel Internet

Bringing you an incredibly fast Internet experience

4G_{LTE}

Sri Lanka
Telecom
Mobitel

We Care. Always.



Sponsors & Partners

Education Partner



Software Partner



Mobile Communication Partner



Technology Partner



National Partner



Exclusive Print & Digital Media Partner



Electronic Media Partner



Endorsed by



Official Airline Partner



Education Media Partner



Hospitality Partner



Annual CSSL Strategic Partners



Supported by



Computer Society of Sri Lanka

275/75 Prof. Stanley Wijesundara Mawatha,
Colombo 07, Sri Lanka

Telephone : +94 11 4713290, +94 114713336,

Fax : +94112508009, Email : info@cssl.lk, Web : www.cssl.lk



2279-3895

National Digitization Project

National Science Foundation

Institute : National Science Foundation

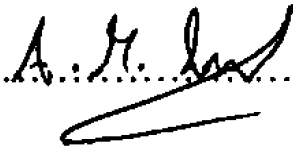
1. Place of Scanning : Sanje (Private) Ltd, Hokandara

2. Date Scanned :02/06/2017.....

3. Name of Digitizing Company : Sanje (Private) Ltd, No 435/16, Kottawa Rd,
Hokandara North, Arangala, Hokandara

4. Scanning Officer

Name :Angelo Melvin Luwis.....

Signature :.....

Certification of Scanning

I hereby certify that the scanning of this document was carried out under my supervision, according to the norms and standards of digital scanning accurately, also keeping with the originality of the original document to be accepted in a court of law.

Certifying Officer

Designation :Information Officer.....

Name :Renuka Sugathadasa.....

Signature :.....

Date :02/06/2017.....

“This document/publication was digitized under National Digitization Project of the National Science Foundation, Sri Lanka”