Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 4, April -2017

Study and Implementation of Agile Practice using SCRUM in Software Development

(Based on incremental software development approach)

[1]Sapana Tapdiya, [2]Rashmi Yadav

[1,2]*Department of Computer Science, Acropolis Technical Campus, Indore, India*

Abstract—*The objective of the paper is to investigate how the perception of bottlenecks, unnecessary work, and rework to as changes when migrating from a plan-driven to an incremental software development approach with agile practices (flexible product backlog, face-to-face interaction, and frequent integration). The context in which the objective should be achieved is large-scale development with a market-driven focus. The selection of the context was based on the observation in related work that mostly small software development projects were investigated and that the investigation was focused on one agile model. It feels that Scrum is mainly about making things more visible and measurable. It makes requirements more visible, progress more measurable and impedances more visible as it is a management framework but does not define engineering practices. These relatively subtle changes are already leading to improvements in projects, and that in time, with organizations embracing Agile principles more completely, higher levels of effectiveness can be expected.*

Keywords — *Plan driven, agile software development, migration.*

I. INTRODUCTION

During the past fifty years, new software development approaches were introduced to fit the new cultures of the software industry and frequently changing customer needs. In the initial age, clients requirement were very much clear and stable. Initially users of software were very few and also software was not that much user friendly.

Numerous surveys on agile methods have indicated increased adoption rate of Agile methodologies in software companies. More and more companies have started to implement agile methodologies either in pure form or in combination with plan-driven methods.

Companies practicing agile methods have indicated various benefits such as reduced time to market, software development costs, planning efforts, increased customer satisfaction and so on. Software professionals have also reported higher project success rates of agile projects as compared to projects managed by plan driven approach. Just after three years of this survey, another survey indicated 70% higher project success rates of agile projects as well as higher quality, better stakeholder satisfaction and improved Return on Investment (ROI) as some of the success factors of agile projects. The results of 2012 survey indicated 90% improvement in ability to respond to changing priorities, 81% improvement in software quality and 85% increase in productivity [7]. The results of all these surveys indicate increased use of agile methods and various benefits offered by them as compared to traditional plan-driven approaches.

II. FOUNDATION

Failure of many software projects in 1960s and 70s led to evolution of software development processes. Many software projects either ran out of budget or were scraped due to delays or because of final product not meeting user expectations.

The main purpose of software development processes and models is to guide the software development process. They define the order of execution of various phases and transition criteria from one phase to another phase (Boehm, 1988) [4].

One of the assumptions in waterfall model is requirements don‘t change. However, this rarely happens. User requirements evolve during product development and hence they keep on changing.

Waterfall Model also suffers from Late Design Breakage‘. In waterfall model, software development progresses fine until the integration phase when design issues are uncovered and flaws are discovered (Royce, 1970). This results in unplanned rework and stress on budget and resources. Figure 01 shows comparison between waterfall model and iterative software development with respect to late design breakage phenomena.

Format	Evolving Management and Engineering Artifacts			
Activity	Inception	Elaboration	Construction	Transition
Product	Prototypes	Architecture	Usable Releases	Product Releases

@IJAERD-2017, All rights Reserved

Studies have investigated the advantages and disadvantages of plan-driven and agile processes. However, few studies present a comparison of the models in general, and the effect of moving from one model to the other. This section summarizes the results of existing empirical studies on both process models, presenting a list of advantages and disadvantages for each of them. The description of studies related to plan-driven development is not split into advantages and disadvantages as few advantages have been reported.

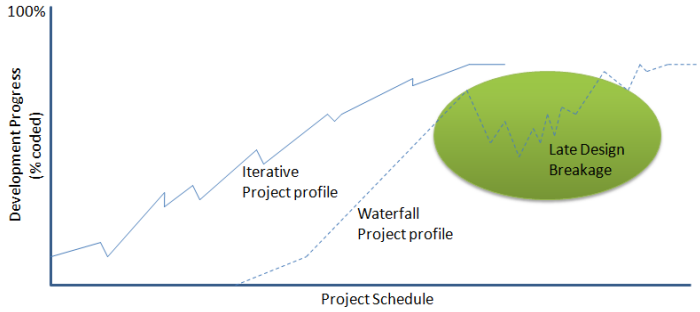

Fig 01: Comparison with respect to late design breakage phenomena.

III. HYPOTHESIS FORMULATION

This section describes the problem, approach taken for data gathering for research. The findings discussed in the Review are used as the basis for developing the research methodology. It also covers the difference between Agile and Plan driven approaches and discusses the research problem. A conceptual framework is then presented, identifying the independent and dependent variables. This section also presents the design of the strategic questionnaire and discusses the methods of data collection with samples of selection, process & methods of data collection.

Boehm & Turner (2003) have identified home grounds in four areas namely Application, Management, Technical and Personnel for agile and traditional approaches where they work best.

Application: Agile methods work best in smaller projects of size up to 40 members, how-ever there have been cases where agile methods have been able to scale up. Traditional methods on the other hand work well in larger and complex projects. In terms of environment, agile methods are most appropriate for turbulent, high change environment where requirements keep on changing. Traditional methods are best suited for projects with stable requirements with change rate of only 1 percent per month. (Boehm & Turner, 2003) [4].

Management: Agile and plan-driven methods differ in the way they are managed. Agile methods depend upon close customer interaction. They stress upon regular customer involvement and feedback. Product delivery is considered more important than planning. Team‘s tacit knowledge is the base for planning rather than documentation. Traditional methods on the other hand are based upon formal contract between customer and developers. Extensive planning mechanism and documentation is used upfront in order to avoid any risk. (Boehm & Turner, 2003) [4].

Technical: In agile methods, development takes place in short iterative cycles delivering potential usable product at the end of each iteration. Simplicity is the basis of agile methods like XP where simple architecture and design is stressed upon. In techniques like TDD and ATDD, tests are written before code and tests drive development. Plan-driven methods depend upon upfront requirements gathering and analysis. Most of the requirements are explained in the beginning of the project and agreed upon. In traditional methods, testing occurs as the last phase of software development and hence they suffer from expensive late design breakage‘ which has already been explained in Figure 01. Personnel: Agile methods‘ success depends upon Collaborative, Representative, Authorized, Committed and Knowledgeable (CRACK) customers. On-site customer presence is considered very important. Plan-driven methods also benefit from the presence of CRACK customers but it is not the requirement. Agile methods have the dependency on highly experienced and skilled developers – 30 percent full-time level 2 and level 3 resources. Traditional methods can work with mixture of low skilled and high skilled resources. In terms of culture, agile approach-es work in a chaotic environment, whereas plan-driven approaches work in an environment with clearly defined roles and responsibilities. (Boehm & Turner, 2003) [4].

Mainly research problem divided into three parts of the simplicity, which will help us to focus on them one by one-

1) Analyze whether Agile (Scrum) is an effective methodology for software development above plan driven approach.
2) Measure its effectiveness with respect to meeting quality, cost and time targets.
3) Identify the factors which would have an impact on the effectiveness of the project with respect to scrum.

@IJAERD-2017, All rights Reserved

The different factors can be used as the starting point for an attempt to verify their applicability of the scrum methodology, as having an impact on the effectiveness of project development. In this research we assume that the effectiveness of a project can be defined as the overall perception of success of the project. Effectiveness can be measured on four dimensions which depict the overall perception of success of a particular project, namely:

1) Quality (i.e. delivering a working product)
2) Scope (meeting all requirements of the customer)
3) Timeliness (delivering on time)
4) Cost (within estimated cost and effort)

The conceptual framework for this research is defined below.

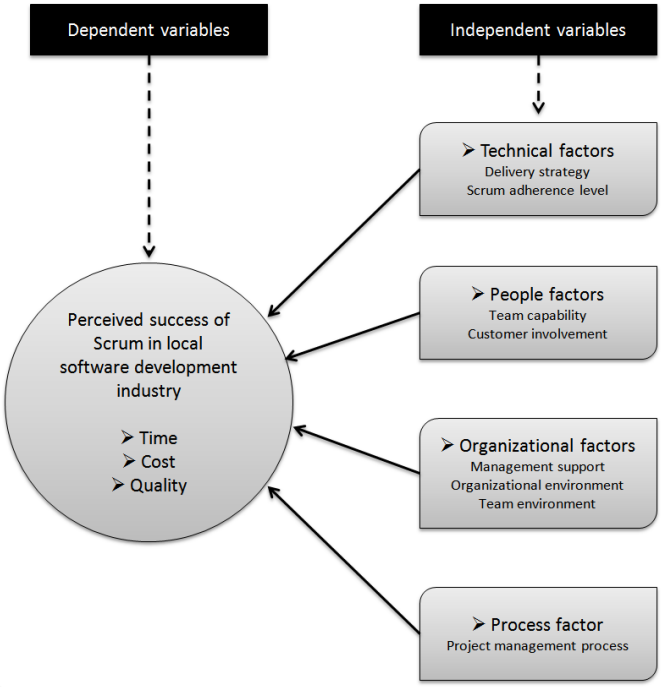

Fig 02: Conceptual framework of this study

Based on the initial research model, the following independent and dependant factors are considered in this research as shown in Fig. 02. The project factor is discarded as there are only a few companies which use Scrum and thus the project base is small and immature.

Based on the research and strategic question, we tried to answer the "what" and "how" questions by comparing critical factors derived from theories in relevant literatures to the empirical data gathered from our respondents. This thesis, therefore, is conducted using a qualitative method where most of the information we gathered were from the respondents' practical experiences. We interpreted the responses from our respondents to get an in-depth understanding in order to examine the critical factors that influence the outcome of a Scrum implementation.

After reviewing various literature on the concept of Scrum, success case studies on Scrum, and issues on Scrum; we have developed an analysis model based on theoretical observations and chosen to present our findings in this section. Our analysis model summarizes the problems of Scrum implementation and identifies critical factors that are essential to the success of Scrum implementation.

A list of these respondents is shown below in table 01. The respondents requested that their companies should not be mentioned, and therefore the names of the respondents are not disclosed.

@IJAERD-2017, All rights Reserved

Name	Background and Responsibility
Respondent 1	Respondent 1 is working as a Scrum Master in a telecommunication project in Sweden. She has an intensive background in software development and has experience in using both Waterfall and Scrum methodologies for various industries such as insurance, banking, and now telecommunication. Moreover, she also has strong business knowledge and technical skills with 10 years of experience.
Respondent 2	Respondent 2 is working as a Scrum Master in a printing service project for a customer in Sweden. He possesses extensive technical skills with 6 years of experience in web application and C# programming language.
Respondent 3	Respondent 3 is acting as a leader of the development team, which is equivalent to the role of a Scrum Master, in an insurance broker management project in Thailand. With 8 years of professional experience, he has a strong technical background specifically programming languages like Java.
Respondent 4	Respondent 4 is working as a Scrum Master and is developing web applications for a business consulting company in Sweden. He has worked in the field of software engineering for 5 years and has experience in both Agile and Scrum methodologies.
Respondent 5	Respondent 5 is currently working as a project manager and a Product Owner for an insurance broker management project in Thailand. He specializes in the insurance industry with 15 years of experience in software development.

Table 01, List of Respondents with details

B. Data Collection

Researchers should employ the method to collect and analyze data that minimizes errors and biases which affect the stability and consistency of the results. The data are collected from both primary and secondary data sources. Data collection details are shown below.

Population	Software professionals working in Indian & US based software development organizations that use Scrum.
Sampling Population	Software professionals who are working in small, medium or large scale software organization that use Scrum.

Table 02, Data Collection details

@IJAERD-2017, All rights Reserved

These organizations represent large organizations as well as small organizations, and also a reasonably broad range of years of operations. All of these organizations are also headquartered outside the nominated place, and this analysis only considered the India & USA based operations.

IV. ANALYSIS AND DISCUSSION

The information derived from critical literature review and finding would be discussed in detail and analyzed following the analysis model in order to come up with the critical factors that are important in a Scrum implementation. We created a qualitative analysis from the results of the completed questionnaires which was mainly used to answer the research question. This chapter also describes the analysis of interviews conducted with respondents in different industries located both in India & USA. We asked few questions that follow the same pattern as our research structure and our questionnaire. We were able to gather a lot of valuable information from the responses to these questions.

This section explains how weights were assigned to the responses to determine the project's adherence to the Scrum standard, as well as the internal consistency analysis.

Responses were assigned weights based on the standards of Scrum described in the section 2.4 (refer [19]). The expected answers for the questions were given a weight of 3 points if they agreed with the Scrum standard as discussed previously. Where the response deviated from the standard, a weight of 2, 1 or 0 was allocated depending on the magnitude of the deviation. The following assumptions were made during the coding as there were answers which could be interpreted in different ways.

- 1 month = 4 weeks
- 1 working day = 8 hours

The following table describes how the weights were assigned for each question based on the level of deviation. Decomposed factor is scrum adherence.

Question	Rule	Weight
Scrum roles in team	Identification of 3 standard roles Identification of 2 standard roles Identification of 1 standard role	3 2 1
Daily Scrum	Yes No	3 0
Daily Scrum duration	15 minutes, 5-10 minutes/10-15 minutes/20 minutes, 30 minutes 3 hours/60mins (Once/twice each week)/null	3 2 1 0
Team members for Scrum team	7 5/6/8/9 3/4/10/11 Less than 3 and more than 11	3 2 1 0
Deliverables at end of each Sprint	Yes No	3 0
Sprint duration	4 weeks 3 weeks 2 weeks Quarter/None	3 2 1 0
Sprint planning meeting	1 day 4 hours/2 days 2 hours/2.5 days	3 2 1
Retrospective meeting	2-4 hours 1-2 hours/4 hours to 1 day 15mins/30mins None	3 2 1 0

Table 03, How weights are defined based on level of deviation

@IJAERD-2017, All rights Reserved

Internal consistency of the dataset was tested using Cronbach's Alpha (= 0.881) and found to be highly consistent. Factor-wise internal consistency statistics are listed in Table 04 below.

Factor	Cronbach's Alpha (_)
Organizational factor	0.821
People factor	0.887
Process factor	0.824
Technical factor 0.702	0.702

Table 04, Factor-wise internal consistency statistics

According to Table 04, Cronbach's Alpha for all sections was above 0.70 and data in all sections were therefore judged to be internally consistent.

A. Analysis of Scrum Adherence Level

After performing the statistical analysis, the finalized underlying factors and the attributes are listed below.

Decomposed factors	Questions considered for analysis
Management support	1
Organizational environment	3
Customer involvement	6,7,9,10,11,12,13,14,15
Team capability	16
Project management process	17,18,20,22,23,24
Delivery strategy	25
Team environment	2,4,5,8,19

Table 05, Underlying factors and the attributes

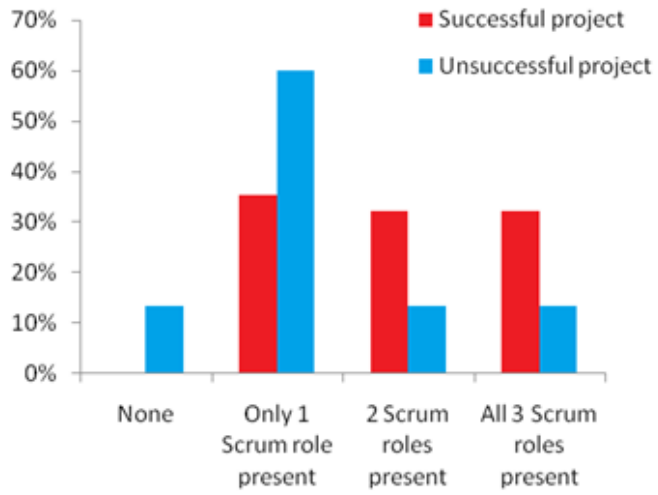

Fig 03: Scrum roles in team – successful projects versus unsuccessful projects.

While Scrum defines three clear roles the data showed that many projects had less than three roles. Only 27% of unsuccessful projects had at least two of the three formal Scrum roles, whereas 65% of successful projects had at least two of the three formal Scrum roles.

The below graphs show the percentages of successful and unsuccessful project that conducted a daily Scrum.

@IJAERD-2017, All rights Reserved

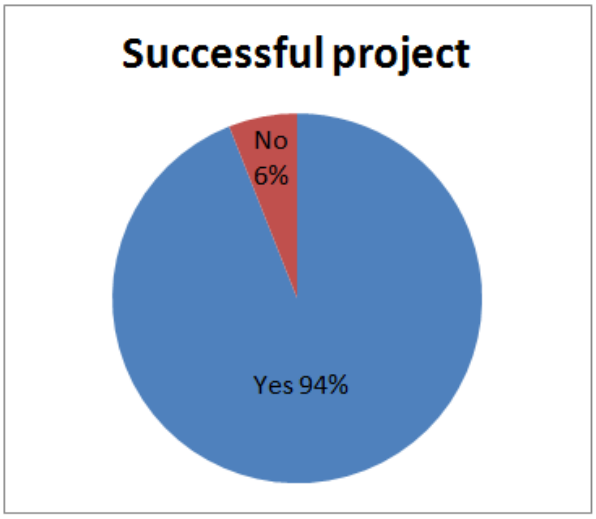

Fig 04: Daily Scrum - Successful project

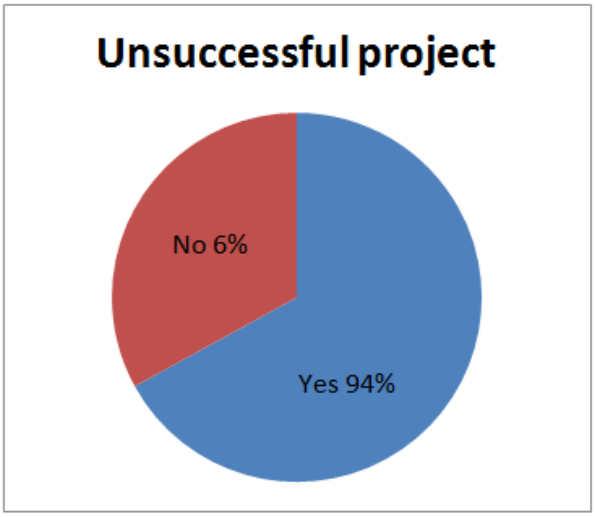

Fig 05: Daily Scrum - Unsuccessful project

Only two thirds of the unsuccessful projects had a daily Scrum, whereas almost all (94%) successful projects had a daily Scrum. The Scrum standard requires a daily Scrum of 15 minutes. The below graph compares the distribution of daily Scrum durations between the successful and unsuccessful projects.

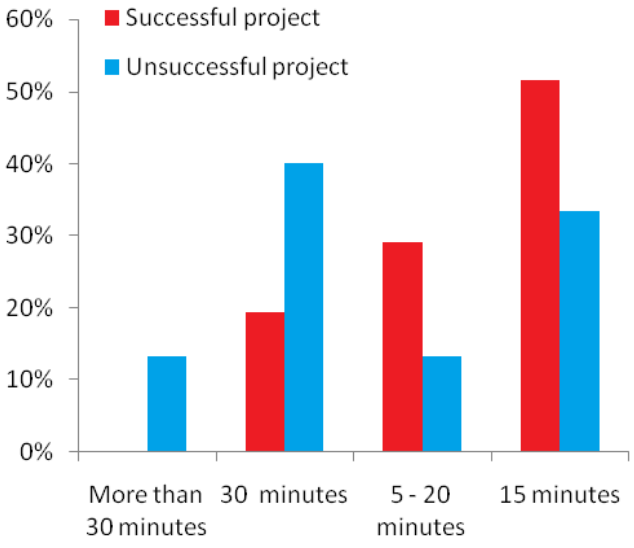

Fig 06: Daily Scrum duration – successful projects versus unsuccessful projects.

@IJAERD-2017, All rights Reserved

Where the Scrum standard recommends a daily Scrum of 15 minutes, 52% of successful projects reported having a daily Scrum of exactly 15 minutes, where only 33% of unsuccessful projects adhered to this standard. Further, 80% of successful projects had a daily Scrum that lasted between 5 and 20 minutes. On the other hand, 53% of unsuccessful projects had a daily Scrum that was 30 minutes or longer.

B. Measuring project effectiveness

A measure of effectiveness was introduced and this section analyses statistics based on this composite effectiveness score as well as its components. The below graphs show the agreement of the respondents with their projects' adherence to initial time estimates.

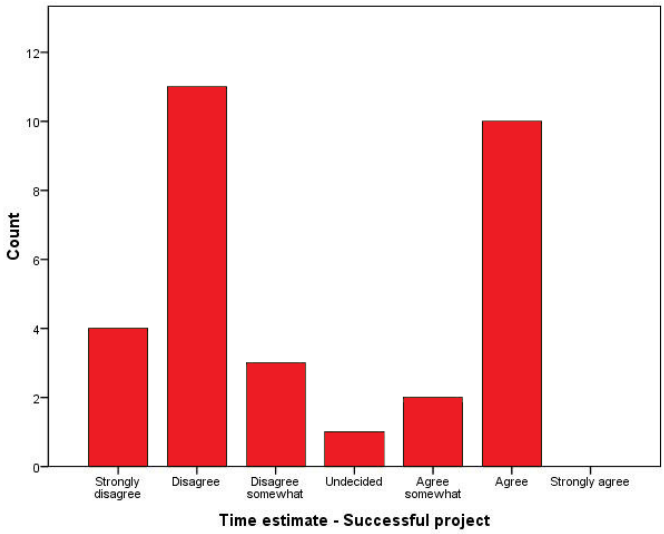

Fig 06: Time estimates – Successful projects

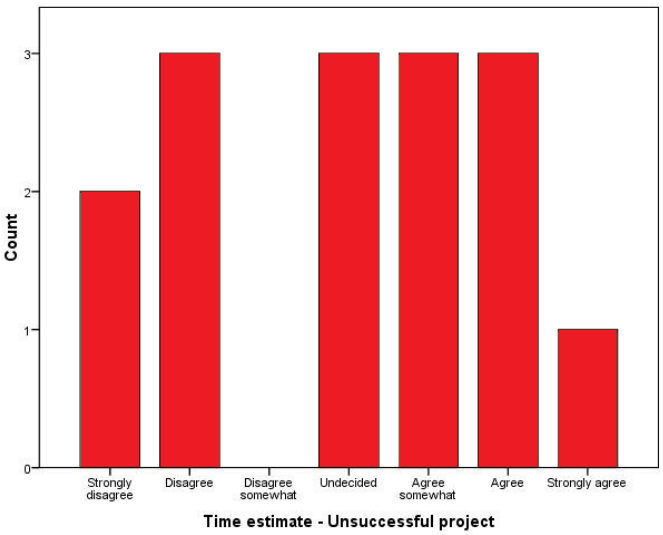

Fig 07: Time estimates – Unsuccessful projects

The distribution of adherence to time estimates among the successful and unsuccessful projects did not appear to be normal. A bi-modal distribution was seen in the successful projects, and a roughly random distribution in the unsuccessful projects.

A possible explanation for this anomaly is that by its nature, Scrum leads to implementation of the highest priority requirements within a given fixed time frame. A possible further explanation for the mode at "Disagree" for adherence to Time estimates is that, some projects may have added extra Sprints in order to implement all requirements.

C. Testing the project effectiveness score

In order to test the derived effectiveness score, a paired T-test was performed, comparing the effectiveness scores of the successful projects against that of the unsuccessful projects. The individual components (i.e.: Time, Cost and Quality) that make up this measure are also compared similarly. The table below summarizes this test.

@IJAERD-2017, All rights Reserved

Pairs	Mean	Std. Deviation	Group Mean
Time – SP	3.06	1.79	-.933
Time - UnSP	4	1.96	
Cost – SP	3.87	1.51	0.6
Cost – UnSP	3.27	1.58	
Quality – SP	4.73	1.39	1.2
Quality – UnSP	3.53	1.73	
Effectiveness – SP	13.53	2.56	2.73
Effectiveness – UnSP	10.80	2.51	

Group Std. Deviation	t-test value	Degrees of freedom	Sig. (2-tailed)
2.43	-1.49	14	0.16
1.92	1.21	14	0.25
2.34	1.99	14	0.07
3.88	2.73	14	0.02

Table 06: Summary of the test

Based on this test, the derived mean effectiveness measure is seen to be significantly different between the successful projects (M=13.53, N=31) and the unsuccessful projects (M=10.8, N=15), paired $t(14) = 2.73$, $p < 0.05$. This verifies that the effectiveness value is actually able to differentiate between successful and unsuccessful projects.

No such significant differences were found in the three components that made up this effectiveness measure. Because of this, statistical analysis in the rest of this thesis will focus on using this composite effectiveness measure, although tests with the individual components will also be reported where applicable.

V. TESTING OF HYPOTHESIS

This chapter describers the problem findings of the hypothesis testing based on correlations and large/small scrum. Each of the hypothesis is checked in turn with the bivariate correlation. Ho: There is no relationship between a correct delivery strategy and effectiveness of Scrum, where Ho:$\beta=0$ and Ha:β is not equals zero.

A. Correlation of Delivery Strategy and effectiveness in terms of Quality, Cost and Time

A two-tailed Pearson correlation test was performed, and a highly significant correlation was found between delivery strategy and effectiveness of Scrum, $r(46) = 0.527$, $p < 0.001$. Therefore, we reject H0 and accept Ha under the 99% confidence level. In other words, there is a relationship between good delivery strategy and the effectiveness of Scrum.

The following table displays the breakdown of effectiveness in terms of quality, cost and time.

@IJAERD-2017, All rights Reserved

	Pearson Correlation	Significance Value	Significance
Quality	0.608	< 0.001	Significant
Cost	0.451	0.002	Significant
Time	-0.046	0.764	Not Significant

Table 07: Correlation of DS and effectiveness in terms of Quality, Cost and Time

According to the Table given above, Cost and Quality are significantly correlated with delivery strategy whereas Time is not significantly correlated. A possible explanation for the lack of statistical significance in the correlation between delivery strategy and adherence to Time estimates.

B. Correlation of Team Environment and effectiveness in terms of Quality, Cost and Time

A two-tailed Pearson correlation test was performed, and there was no significant correlation between team environment and effectiveness of Scrum, $p > 0.05$. Therefore, we accept H(0) and reject H(a) which means that there is no relationship between Agile-friendly team environment support and the effectiveness of Scrum in project management.

The following table displays the breakdown of effectiveness in terms of quality, cost and time.

	Pearson Correlation	Significance Value	Significance
Quality	0.103	0.495	Not Significant
Cost	0.126	0.404	Not Significant
Time	0.610	< 0.001	Significant

Table 08: Correlation of TE and effectiveness in terms of Quality, Cost and Time

C. Correlation of Scrum adherence level and effectiveness in terms of Quality, Cost and Time

A two-tailed Pearson correlation test was performed, and no significant correlation was found between Scrum adherence level and effectiveness of Scrum, $p > 0.05$. Therefore, we accept H(0) and reject H(a) which means there is no relationship between Scrum adherence level and the effectiveness of Scrum in project management.

The following table displays the breakdown of effectiveness in terms of quality, cost and time.

	Pearson Correlation	Significance Value	Significance
Quality	0.070	0.645	Not Significant
Cost	-0.079	0.601	Not Significant
Time	-0.288	0.053	Not Significant

Table 09: Correlation of SAL and effectiveness in terms of Quality, Cost and Time

According to Table given above, none of Time, Cost and Quality are significantly correlated with Scrum adherence level. While this was a rather surprising finding, a possible explanation for the lack of statistical significance in the correlation between Scrum adherence level and adherence to Time estimates.

D. Correlation of customer involvement and effectiveness in terms of Quality, Cost and Time

A two-tailed Pearson correlation test was performed, and a highly significant correlation was found between customer involvement and effectiveness of Scrum, $r(46) = 0.527$, $p < 0.001$. Therefore, we reject H(0) and accept H(a) under the 99% confidence level. In other words, there is a relationship between strong customer involvement and the effectiveness of Scrum in project management.

@IJAERD-2017, All rights Reserved

The following table displays the breakdown of effectiveness in terms of quality, cost and time.

	Pearson Correlation	**Significance Value**	**Significance**
Quality	0.568	< 0.001	Significant
Cost	0.435	0.003	Significant
Time	-0.092	0.544	Not Significant

Table 10: Correlation of CI and effectiveness in terms of Quality, Cost and Time

According to Table given above, Cost and Quality are significantly correlated with customer involvement whereas Time is not significantly correlated. A possible explanation for the lack of statistical significance in the correlation between customer involvement and adherence to Time estimates.

E. Summary of test result

Team environment and Scrum adherence level were determined to not have an apparent relationship with the effectiveness of a project. While this is rather surprising, this may be the result of the relatively small sample size.

The following table summarizes the results of the correlations.

Factor	**Correlation result**
Strong management support	**Significant**
Agile-friendly organizational environment	**Significant**
Agile-friendly team environment	Not Significant
Team with high capability	**Significant**
Strong customer involvement	**Significant**
Agile project management process	**Significant**
Good delivery strategy	**Significant**
Scrum adherence level	Not Significant

Table 11: Result of the correlations

VI. CONCLUSION & FUTURE WORK

This research was conducted to fill the knowledge gap to some extent, for the benefit of the software industry. This research study set out to design a survey to explore the factors that have an impact on the effectiveness of a project using quantitative methods. The data was collected from six companies of various sizes and provided enough empirical information for statistical analysis to arrive at a number of conclusions.

Understandably, this conclusion was derived from testing correlations, and so no judgments can be made about whether changes in these identified factors lead to changes in project effectiveness. Practitioners are however encouraged to consider the six factors identified above as potentially having an impact on the final effectiveness of their projects, and are encouraged to facilitate a positive level for each of these factors.

The main finding of this research is the relationship with the factors identified and the effectiveness of Scrum in software development. The organizations that want to adopt Scrum and the project managers who wish to try Scrum as a software development methodology should consider the above significant factors, namely, strong management support, Agile-friendly organizational and team environments, a capable team, strong customer involvement, Agile project management process and having an effective delivery strategy to make the project a success.

This thesis also shows that the level of Scrum adherence does not have a significant correlation with effectiveness of projects. Even in the interviews conducted, some professionals from at least two organizations commented that they do not use Scrum as it is but they use a mixture (hybrid) of Agile concepts. The author feels that these hybrid approaches explained previously contribute to this surprising finding.

It feels that Scrum is mainly about making things more visible and measurable. It makes requirements more visible, progress more measurable and impedances more visible as it is a management framework but does not define engineering

practices. These relatively subtle changes are already leading to improvements in projects, and that in time, with organizations embracing Agile principles more completely, higher levels of effectiveness can be expected.

Even though we found many key success factors and a lot of benefits from using the Scrum methodology in projects; there are some disadvantages of using Scrum. Further research can be carried out to find a set of failure factors that illustrate Scrum's limitations. In addition, we found that a sub-factor from a factor could also influence other sub-factors in another factor, e.g., Communication can directly affect Customer Support. Further research would be carried out to try to show the relationships between the three factors Organizational, People, and Technical, and seven sub-factors Management Support, Customer Commitment, Work Place, Tools and Technology Support, Communication, Learning and Training, and Plan-driven Project.

VII. REFERENCES

[1] Highsmith, J. (2000). Adaptive software development. New York: Dorset House.
[2] Highsmith, J. (2002). Agile software development ecosystems. Boston: Addison-Wesley.
[3] Ahern, D., Clouse, A., & Turner, R. (2001). CMMI distilled. Boston: Addison-Wesley.
[4] Boehm, B. (2002, January). Get ready for agile methods, with care. Computer, 64-69.
[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[6] Copeland, L. (2001, October). Developers approach extreme programming with caution.Computerworld, 7.
[7] Claes Wohlin, (July 2010) "Incremental software development approach with agile practices".
[8] Karlström D, Runeson P (2005) Combining agile methods with stage-gate project management. IEEE Softw 22(3):43–49
[9] Koch AS (2005) Agile software development: evaluating the methods for your organization. Artech House, Boston
[10] Laplante PA, Neill CJ (2004) Opinion: the demise of the waterfall model is imminent. ACM Queue 1(10):10–15
[11] Larman C (2003) Agile and iterative development: a manager's guide. Pearson Education, Boston
[12] Bahli B, Abou-Zeid ES (2005) The role of knowledge creation in adopting xp programming model: an empirical study. In: ITI 3rd international conference on information and communications technology: enabling technologies for the new knowledge society.
[13] Baskerville R, Ramesh B, Levine L, Pries-Heje J, Slaughter S (2003) Is internet-speed software development different? IEEE Softw 20(6):70–77
[14] Beck K (1999) Embracing change with extreme programming. IEEE Comput 32(10):70–77
[15] Benediktsson O, Dalcher D, Thorbergsson H (2006) Comparison of software development life cycles: a multiproject experiment. IEE Proc Softw 153(3):323–332
[16] Ceschi M, Sillitti A, Succi G, Panfilis SD (2005) Project management in plan-based and agile companies. IEEE Softw 22(3):21–27
[17] Cohen D, Larson G, Ware B (2001) Improving software investments through requirements validation. In: Proceedings of the 26th annual NASA Goddard software engineering workshop (SEW 2001). IEEE Computer Society, Washington, p 106
[18] Cohen D, Lindvall M, Costa P (2004) An introduction to agile methods. In: Advances in computers. Elsevier, Amsterdam.

@IJAERD-2017, All rights Reserved