COMPARISON OF AGILE AND TRADITIONAL SOFTWARE DEVELOPMENT TECHNIQUES BASES ON PREVIOUS STUDY AND MARKET ANALYSIS TO FIND BEST METHODOLOGY FOR DEVELOPMENT

[1*]Javed Ali Khan, [2]Izaz Ur Rehman, [3]Salman Khan, [4]Shahpoor Khan, [5]Iftikhar Javed Khan , [6]Yawar Hayat Khan, [7]Muhammad Assam

[1,2,3,4,7]Department of Software Engineering, University of Science and Technology Bannu, Pakistan

[5,6]Department of Electrical Engineering. University of Science and Technology Bannu, Pakistan

*Corresponding E-mail: engr_javed501@yahoo.com

ABSTRACT*— This paper provides detailed assessment of agile and traditional software development methodologies. Parameters used for the assessment of agile and heavy software development methodologies are project size, customer, developers, risks, refactoring, and architecture, planning and control, requirements, documentation, emphasis and return on investment. Both methodologies are critically reviewed. To support the assessment survey is conducted in Pakistan software industries to find which methodology is best for different size of software development. Questionnaire is prepared to get feedback from the software industry practitioners. Our finding shows that agile methodologies show great result for small and medium projects as they are completed within time, resources and according to the needs and satisfaction of customers while heavy methodologies are best for large projects and such software which involved allot of risks.*

Keywords— Agile development, Traditional software development, Scrum, Requirement Engineering, Software Engineering, Best software Methodology[1].

INTRODUCTION

In the domain of software engineering, agile development is the leading epitome that is majorly choosen by software industry. Many research papers are published on agile development techniques in the last few years. The classical approach of constructing software techniques, accommodate the general engineering epitome of requirements, design, code and maintenance. These techniques are also known as waterfall based, acquired from traditional software development epitome .These are also called as plan–driven [2],documentation driven , heavyweight techniques and big design upfront [1]. Boehm and Phillip [3] shares their development experience that software requirements are volatile as it changes by 30% or more. Because of regular alterations in business world and technology, it become challenging for traditional software development techniques to collect requirements correctly [4]. Williams et al says that one more serious problem with heavy software development techniques are lack of response to software changes which often leads software project to failure [5]. From literature, it can be seen that software projects usually composed of many defects and failures [6;7;8]. Therefore, in order to develop quality software which fulfils the demands of the customers, it is therefore mandatory to use practices, development experience, case tools and methodologies in order to get the project successful [9,10]). In this modality, the market has got two techniques, known as agile and traditional [9,11,12,13,14].

Agile methodologies surround on adaptive techniques while heavy software methodologies surrounds on prognostic techniques [15]. In classical SDLC designs, groups work on elaborated projects having a entire table of contents and specific goals that should be achieved during upcoming months or whole life of the product. The prognostic techniques entirely depends upon the requirement elaboration and cautious planning before the start of thenew round. Any change required in requirements must under goes through the rigorous change control process and prioritization. The agile model utilizes accommodative method in which no brief plans are made and only most priority requirements are focused for development. Teams respond quickly and efficiently to accommodate the impulsive changes in the project. In agile development projects goes under frequent testing in order to minimize errors and their effects. The main constituent of agile techniques are communication with consumer and lesser documentation are the main features of agile development. The group communicates frequently and are within the similar geographic area.

In contrast to classical software development, agile development refrains upfront requirement gathering from stakeholders as complete and correct requirements can't be gathered at once [16]. Although agile development have immense success over traditional development but there are few shortcoming of agile development. One of among these problems is that agile dramatically reduce documentation process even some time agile claims that te code itself can act as document [17].

The basis for using agile techniques in software development is that software requirements are impulsive where they are governed by market forces [18; 20]. The Agile development techniques are evolved as a result of failure of the previous traditional techniques to control continuously changing requirements [24]. Williams et al explained that agile development is about the alteration and response [19], that is agile techniques are developed to be accepted rather to be avoided the greater extent of changes. The Agility is explained as the potential to understand and responding to business aspects to remain creative and energetic in the surrounding of an inconstant swiftly changing business world [55]. The agile methods to develop is regarding the agility of the development process, progress groups and their surrounding [21]. These methods involves shared planning's of the many stakeholders and a theme of constantly giving the consumers with projects characteristics in limited time [22].So these enormous and consistent features given is acquired through group work approach [23].

There exists many diverse features between agile development and traditional development approaches,, for example Boehm [1] describes nine agile and heavyweight differences. He assume that the main aim of agile software development techniques is on fast values as well as the main aim of traditional software development techniques is on greater assurance. Dingsoyr *et al* [26] sum up as the discrimination between Agile development and classical development based on an untrusted world, also focusing on the value efficient people and their relation leading to software development. The research carried out by Nerur et al [25] showed a comparison of classical and agile development, they stated seven issues to discriminate classical and agile development methodologies. Their main perception of classical development :----system are completely specific, trustworthy, and are made by enormous planning, On the other hand agile progress ;----a supreme quality software is made by small groups using the principles of constant software design enhancement and continuous testing based on quick feedback and change.

Classical methods have been united in market, possessing constituted principles since 1950s [14;27], the negligence of specified groups accountable for stabilization and distribution of their experiences [28;29],that must be implicated to all projects in spite of size and difficulty in a smooth way, to make sure that the plans initially made for cost , time period , scope , quality are accomplished with lesser possible changes [27,29,31].

In 2001 a novel idea of project development has been developed known as Agile Manifesto, that focus to regulate the projects in an easy, smooth and practical way, enhancing the continuous involvement of consumer in the overall project development, with the focus to give projects efficiently [31]. Agile techniques are suitable to be applied on such projects which are challenging and that goes under constant changes. [14,31,32,33].

COMPARISON OF AGILE AND TRADITIONAL SOFTWARE METHODOLOGIES

Agile SDMs and Traditional SDMs hold many diverse characteristics among each other. Boehm, for example, has reported nine lightweight (agile SDMs) and heavyweight (traditional SDMs) discriminations. Where he states that since the agile methods are used on a small/middle scales and the TSDMs on relatively larger scale, the main objective of ASDMs is of immediate result whereas the primary objective of TSDMs is on efficiency and assurance of the system.

The traditional development is based on the pre-planned approach for software development and agile development uses advanced and more direct software development methods without holding any proper documentations or detailed prescriptions about the underlying system.

In TSDMs the team incorporates a predetermined plan for the diverse tasks and activities regarding the whole system development life cycle (SDLC) while in agile methods teams employ short iterative cycles, and rely on tacit knowledge about a specific task. In traditional software development the requirements elicitation phase occurs at the initial stages of software development and flows through the whole developmental processes which often makes the changes at the later stage of SDLC very difficult, expensive and time consuming, while in agile methods the requirements is a fluctuating entity at each stage of the development which proves it to be quite whippy to amendments. Detail comparisons are shown in Table 1.

Literature Review

In the current years many studies on the utilization of classical and agile methodologies are carried out in order to effectively understand the execution and every operation; it has analyzed them and merge them. Some examples are the research work done by Awad [34], Eder [4] , Fernandez and Fernandez [6] Cross [7], Spundak [9] and Mehar[35]. Hence for an efficient analysis of recent status quo of research on the subject of this paper, it is made effective to have an approach regarding the research objectives mentioned.

Awad [34] did comparison of agile and traditional software development techniques, but their scope was limited to dubai only also parameters used to compare both methodologies were limited.

Eder [4] has explained in his work the current project management approaches in companies , prioritizing them in means of employed project management approach (classical and agile) , to make sure the recognition approach utilized by a company. At the end of study it was concluded as an inventory of practices that assisted to recognize the practices in actual companies and their ranking on the approach used.

Furthermore, Fernandez and Fernandez [6] carried out a historical evaluation of methods and utilization of agile management in projects that have an extra classical approach. Furthermore it showed a background data of agile methodologies in order to boost up the acceptance by professionals.

In his study , Cruz [7] showed a review of the Scrum methodology and the PMBOK, however, he has not commented that one is superior over others , but revealed that both the approaches (classical and agile) can be utilized combinely to make project management useful. Hence the writer showed key concepts of both approaches, as described it can be merged forward together and made statements as in a case study.

In his study , Cruz [7] showed a review of the Scrum methodology and the PMBOK , however , he has not commented that one is superior over other , but revealed that both the approaches (classical and agile) can be utilized combinely to make project management useful. Hence the writer showed key concepts of both approaches, as described it can be merge forward together and made statements as in a case study.

Spundak [9] described a query series about the collection of classical and agile methodologies such as : “ Can you combine different approaches within a single project management methodology ? “ ; Is there a single methodology , which offers a better solution for all projects in a specific environment , for example , a company , or some kind of change is needed to create a best own methodology for project ? “ . At the end of research work, the writers have open up that both classical and agile methodologies have their own effectiveness and pitfalls.

Table 1. Differences between agile and traditional software development techniques.

Activities	Agile Methodologies	Traditional Methodologies
Base Concept	Delivers rapid, flexible and high quality adaptive software by a small group that use the principle of continuous enhancement in design and testing based on personal feed-back and fix	System is fully specified, predicted and is developed through rigorous and detailed pre-planning.
Project boundaries	Small	Huge and detailed
Management style	Supervisory & joint effort	Preplanned approach & control
Emphasis	Intact-People	Specification-Oriented
Change	Open/flexible	Low flexibility
Requirements	Emergent, Rapid Change, Unknown Discovered during the project	Knowable Early, Largely stable, clearly defined and documented
Interactions	Process-driven/informal	Pre-organized/formal
Restarting Charges	Mild	High
Iterations	Multiple (small and big ones)	Formally Specified
Developing Teams	Self-organizing teams	Pre-Specified Team
Quality Focus & Control	Fully control on specifications, design and solutions, permanent testing	Ambiguous planning and tedious control, irregular testing.
Systems’ Knowledge	Implicit/tacit	Explicit/system-related
Developing Model	Iterative & Incremental	Sequential Life-Cycle model
Payback on Investments	Preliminary stages	Completion stages
Developing Groups	Small/Logical/Creative	Huge
Project Boundaries	Small/medium	Large Projects
Primary objectives	Immediate result	Assurance/efficiency/safety
Architecture/Structure	Current requirements	Current & Predictable
Aims	Short Time/Independency	Proper & detailed product
Clients Impressions	On-site/stand-by/attentive	Low intervention, passive
Pre-Planning	None/minimal	Detailed & Elaborative
Success factors	Quick value, adaptive	Encompassing the specifications
Life-Cycles	Iterative ; incremental/evolutionary delivery model	Liner; life-cycle model(water fall, spiral and some variation)
Developers	Active, Vast field knowledge and cooperative	Plan-oriented, having domain knowledge
Risk-tolerance	Higher risk-tolerance	Minor/not-at-all risk-tolerance
Extra required developers potentials	Domain knowledge, ready	Not so important

Where fore the approach choices should be made cautiously considering the design features and characteristics of the systemic environment. Hence, it is possible to aggregate both approaches to a project and within a single methodology, in mind when it is best to use each approach. At last, it was realized that it is crucial that the methodology is used for the project and vice versa.

Mehar [35] conducted a detailed comparison of agile and traditional approaches based on literature also it presented detail implementation of both methodologies. Mehar presentation lacks in conducting survey for both methodologies.

The frame of the research above is a minute sample demonstrating the great curiosity of the scientific community to study the bonding between classical and agile methodologies in order to show the significance of this research.

QUESTIONNAIRE

A questionnaire is developed in order to know which software methodology (either agile or heavy methodology) is followed by software experts to the developed software’s in government and private sector organization in Pakistan for different size of the projects. A detail survey has been conducted in order to know importance of agile and traditional methodology. Below is the summarization of the survey conducted in different software houses of Pakistan.

Questionnaire format

Questionnaires were designed in way so that it make easy for the experts to give answers. For this purpose questionnaires were arranged in such a way so that I covers all the aspect of the software organizations, organizational teams and experts.

Questionnaire Range

Different types of organization were considered for questionnaires ranging from small (having not more than 10 software experts in the organization), to large software organization containing more than 100 employees. Average time taken to complete the questionnaires were around 10 minutes.

Questionnaire Result

Questionnaires results obtained are discussed and simulated in below section.

Organization information and participant's information regarding questionnaires

While taking the survey around 70% of the participants were having the background of information technology, others were mainly from the academia, telecom and engineering departments. 20 organizations were chosen for the survey out of which 70% organizations ranges from 12 to 100 full time employees and remaining organizations were larger above 250 regular employees. Around 85% of the Participants claims that they have enough knowledge about agile software methodologies (which we rate average knowledge) while 90% participates claims that they have excellent knowledge about heavy software methodologies. Similarly, participants having less knowledge about both methodologies are given low weightage in survey.

Mostly, organizations claimed that their organizations are frontrunner in market, as new technology arrives in the market they tries to adopt it in order to get better results. Remaining of the organization place themselves as frontrunner followers as they are happy to adopt new technology after frontrunner adaptation. Almost 85% of the organization uses agile software development methodology as compared to traditional software methodologies techniques. While in case of larger software development organizations happy to use traditional software development techniques in order to having grip on the product.

Methodology Adopted by the Organization.

Results obtained from the expert shows that scrum and XP are the most successful and popular agile software methodologies. It is also observed that developing small and medium scale softwares both XP and Scrum methodologies are used. While development of large scale software traditional software methodologies are used. Waterfall and incremental models are most popular traditional software methodologies.

Agile development Preference over Heavy Software Development.

Agile development gives huge advantage over heavy software development when it comes to development of small and medium scale software developments. Agile methodology are highly appreciated due to lack of documentation and emphases on coding rather than detail documentation for small scale software development. While development of medium scale softwares agile methodology is prefer over traditional methodology due to heavily involvement of customer in the development process rather than doing focus on processes and contracts.

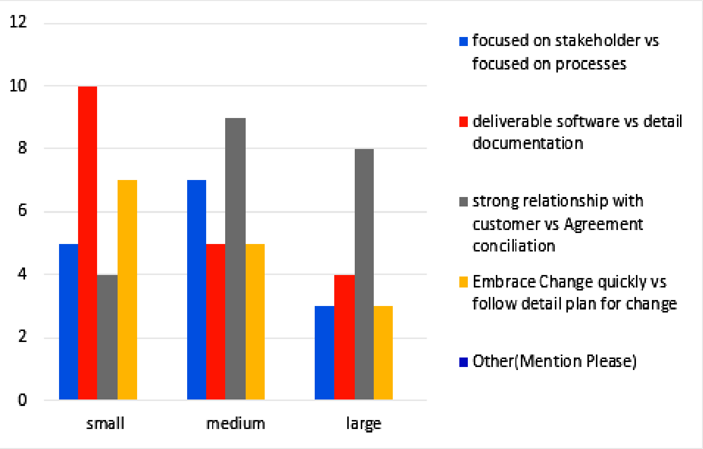

Fig 1. Agile preference over heavy methodologies for difference level of softwares development.

Pros and Cons of Agile and Heavy Software development

When developing small level softwares, features which are more appealing to developers are less documentation and more focus on delivering working software. While considering medium and larger software development, mostly developers gives high rate to customer negotiations over details contracts. Whenever projects become complex and larger in size it becomes difficult to control it, only possible way to manage the project is to involve customers/stakeholders in the development process. Agile has a great feature over traditional development as it includes customer as a part of development team, which is most like when developing medium, complex and larger softwares. Other features are giving more importance to peoples rather than following details process which further make development process slow.

When considering the dislike of agile development, it is observed that when developing small scale softwares agile works perfectly. But when talking about middle and larger software development, agile development usually lacks in detail project management planning and project structure. In Larger software development usually software is divided into structure to make the end product. Now for handling such a large product proper project management activities are required to run the project successfully. According to the survey 70% participants thinks due to less management and product structuring agile is not suitable for large scale development.

In case of heavy methodologies data collected from the experts, it is observed for small scale software development that too much documentation is always disliked. Detail documentation is lengthy activity and their always lack of resources (time and cost) in small scale development. Therefore almost 70% thinks that heavy documentation is problem with small scale development also 20% oppose the lengthy process of change management in heavy methodologies. For medium and large scale development it is concluded that developers does not feel free and relax while commanding management environment also allot of initial planning take allot of time in medium scale software development. Results are shown in Fig 3.

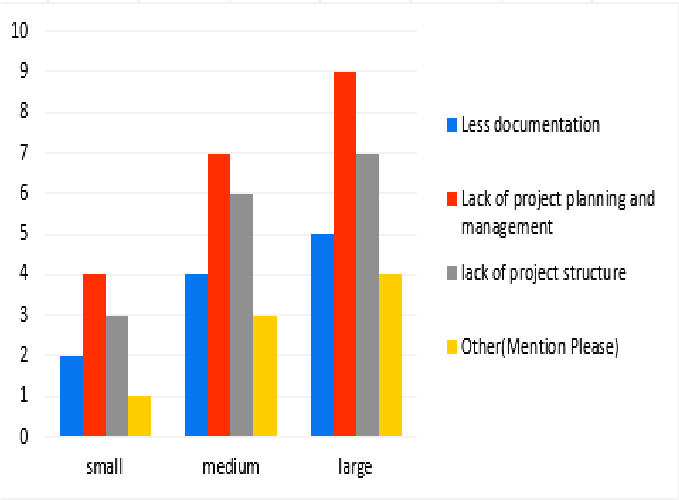

Fig 2. Agile dislikes according to expert inputs.

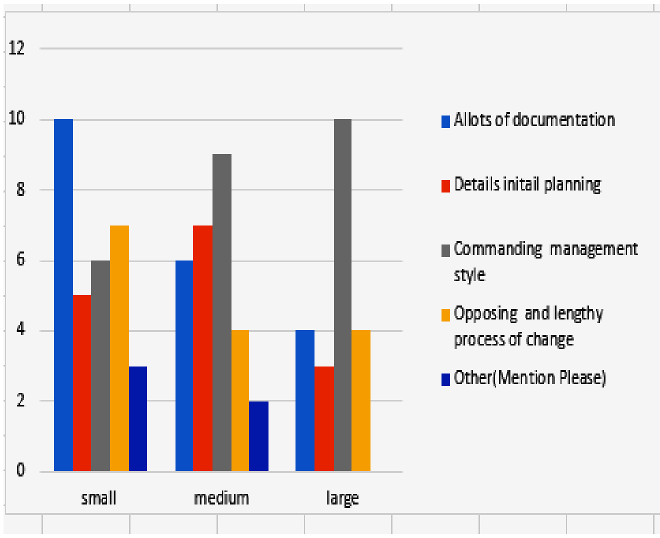

Fig 3. Dislikes of Heavy Software Development methodologies.

4.1.1. Problems with agile methodologies

In order to improve the performance of agile methodologies it is also asked from the participants that what are the main problems which are coming while practicing agile methodologies for developing different scales of softwares. 70% of the participants thinks that availability of agile experts are the main problem in agile development. Agile methodologies need experts which are very difficult to find. According to agile manifesto software engineers are needed for agile development who are good in every phase of software development but he must be expert in any phase of software development. It is also concluded that larger size and project complexity are also the hurdle in agile performance. When project size increases it is difficult to handle the project by smaller teams. Therefore larger team are made to get control over the project but with larger team agile does not give good performance, as allot of communications are need which can't be controlled by agile methodologies. Some participants thinks that for larger projects lack of controlled structure and upfront planning are the obstacles in practicing agile development. The statistics are shown in below diagram.

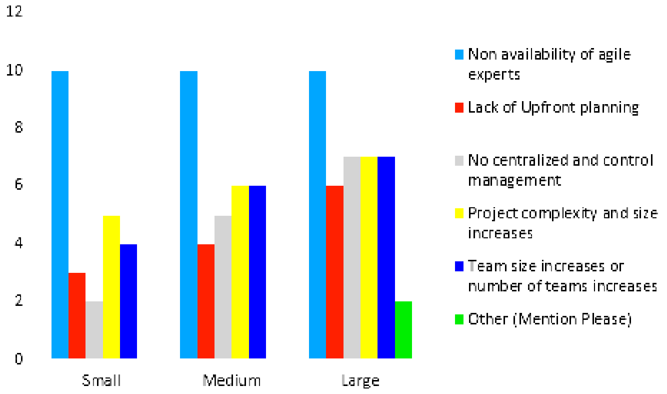

Fig 4. Problems coming in practicing agile methodologies.

Effect of agile on Quality and Cost

When talking about small scale software development, almost 90% participants agree on the point that software Cost decreases when used agile methodology instead of heavy software methodologies. When developing medium scale softwares with agile cost of the software product remain constant as traditional software development but when it comes to larger software development with agile methodologies software cost becomes started to increases as compared to traditional software development methodologies. Details are shown in below Fig 5.

When question about quality was asked from the participants, mostly believe that quality increases when using agile methodology instead of heavy software methodology. Quality come from nonfunctional attributes and continues testing, which are carried out throughout agile development process. In agile development software development comes in increments also customer is considered as a part of development team which further assist in quality development. However due to larger software development some time quality is effected by agile development. 50% of the participants thinks that product quality decreases when follow agile methodology for larger software development. Fig 6 shows the statistics.

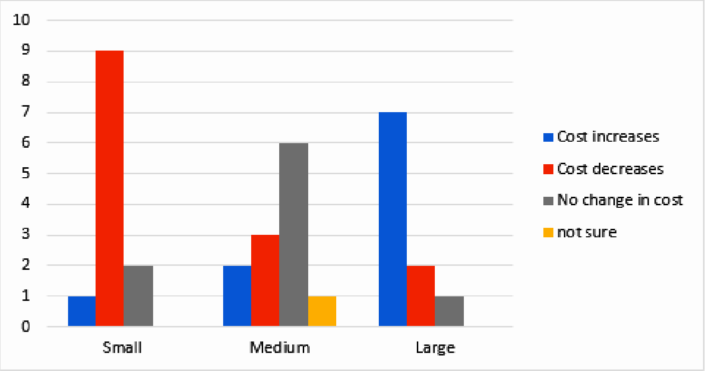

Fig 5. Effect on Cost when using agile instead of Heavy software methodologies.

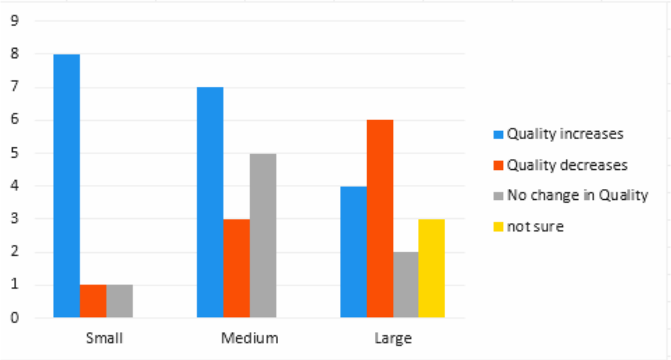

Fig 6. Effect on Quality when using agile instead of Heavy software methodologies

Risk Assessment in agile and traditional development techniques.

Risk is important factor which is consider during the development of software. If risk is not identified at earlier stage during development, it will cause the software to failure. When it is asked from the participants which development methodology assist more when dealing with software risk. 70% of the participants agrees that traditional software development techniques provide platform to deal with the risks. While 20% participants remains neutral. In case of agile development 50% participants agrees that agile provide mechanism to cope with risks while 25% of the participants shows disagreement with agile handling risks.

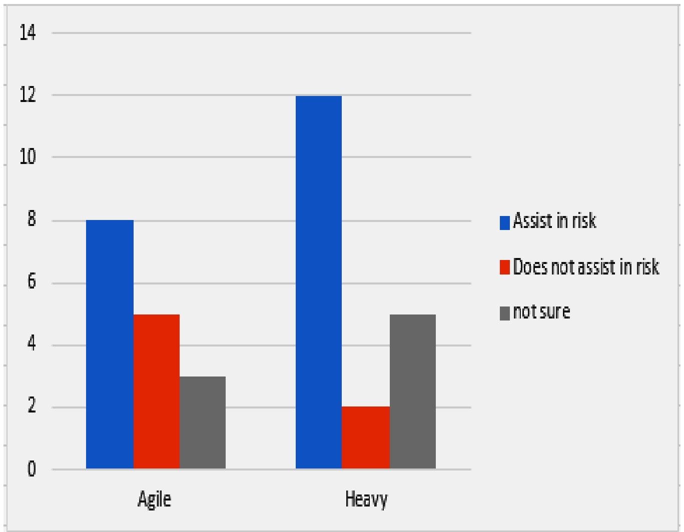

Fig 7. Response of risk in agile and heavy software development.

CONCLUSION AND FUTURE WORK

In this paper, agile and heavy software methodologies are compared based on certain parameters. Both methodology are critically reviewed and advantages and disadvantages of both methods are identified and listed. To support my study I conducted close ended survey in Pakistan software organization to gather useful information about agile and traditional software development techniques.

It is concluded from the results that software experts are happy and satisfied from both heavy and agile software methodologies. Like development of small and medium scale products and software where customers are heavily involved, then experts prefer to use agile methodologies. Softwares having huge budget, allot of risks involved and expected good quality then traditional software development techniques are used. To satisfy customers demands any method can be used.

Agile development is mature enough now and is defined by small set of practices and techniques. Agile development have allot of success stories from which it is concluded that agile development defines a tactical ability, an ability to respond quickly to change, ability to deliver quick product within budget and time, ability to group a high creative peoples and get maximum creativity and innovation from them, and ability to grooms the organization in uncertainty and instability conditions. Whereas traditional software methodologies have rich processes support and having strong organizational structure. Traditional methodologies are best suited in large and risky projects such as defense. Nowadays software nature has been change due to dynamic nature of the customer and customer uncertainty of what features he wants, in such cases agile methodology seems to be best suited. Market strategy has been change, allot of competitors exist therefore every company want early introduction of their products to market. Quick plan is developed for achieving the objectives. Customer is important factor nowadays in software development, so as agile manifesto allow to include customer as a part of team. Customer is focused instead of processes. Also agile allowed to give early version (rough) of the software in order to remove any ambiguities.

However, expert are happy with both agile and traditional software development techniques but its process still needs to be evaluated by a large number of professional from software industries. Research results can be applied to organization to get more profitability and outcomes. Furthermore, studies are required on agile methodologies to make it more useful and efficient in larger software development.

REFERENCES

[1] Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64-69.

[2] Boehm, B., & Turner, R. (2004, May). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*(pp. 718-719). IEEE. pdf

[3] Boehm, B. W., & Papaccio, P. N. (1988). Understanding and controlling software costs. *Software Engineering, IEEE Transactions on*, *14*(10), 1462-1477.

[4] Cho, J. (2008). Issues and Challenges of Agile Software Development with Scrum. *Issues in Information Systems*, *9*(2), 188-195.

[5] L. Williams and A. Cockburn, ―Agile Software Development: It‘s about Feedback and Change,‖ IEEE Computer, June 2003, pp. 39-43.

[6] Yeo, K.T. Critical failure factors in information system projects, International Journal of Project Management, Vol. 20, Issue 3, pp. 241-246. 2002.

[7] Arias, G.,Vilches, D.,Banchoff, C.,Harari, I.,Harari, V.,Iuliano, P., The 7 key factors to get successful results in the IT Development projects, Procedia Technology, Vol. 5, pp. 199-207, 2012.

[8] Marques, A.,Varajão, J.; sousa, J.; PERES, E., Project Management Success I-C-E Model - A Work in Progress,Procedia Technology, Vol. 9, pp. 910-914, 2013.

[9] Eder, S.,Práticas de gerenciamento de projetos de escopo e tempo nasperspectivas das abordagenságil e tradicional,School Engineering of São Carlos - São PauloUniversity- USP, 2012. Verzuh, E. MBA Compacto, Gestão de Projetos, Campus: São Paulo, 2000.

[10] Fernandez, D.J.,Fernandez, J.D., Agile project management: agilism versus traditional approaches, Journal of Computer Information Systems, Vol. 49, pp. 10–17, 2008.

[11] Cruz, F., Scrum e PMBOK: Unidos no Gerenciamento de Projetos, Rio de Janeiro: Brasport, 2013.

[12] Wazlawick, R. S.,Engenharia de Software: Conceitos e Práticas, Rio de Janeiro: Elsevier, 2013.

[13] Špundak, M., Mixed Agile/Traditional Project Management Methodology: Reality or Illusion?,Procedia - Social and Behavioral Sciences, Vol. 119, pp. 939-948, 2014.

[14] K. Peterson, A Comparison of Issues and Advantages in Agile and Incremental Development between State of the Art and an Industrial Case. Journal of System and Software. 2009.

[15] Cho, Juyun. *Issues and Challenges of agile software development with SCRUM*. Issues in Information System. vol ix, No. 2. 2008.

[16] Vijayasarathy, Leo R. *Agile Software Development: A survey of early adopters*. Journal of Information Technology.

[17] Management Volume XIX, Number 2. 2008.

[18] Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, *35*(1), 64-69.

[19] L. Williams and A. Cockburn, ―Agile Software Development: It‗s about Feedback and Change,‖ IEEE Computer, June 2003, pp. 39-43

[20] Highsmith, J., & Cockburn, A. (2001). Agile software development: The business of innovation. *Computer*, *34*(9), 120-127.

[21] Boehm, B., & Turner, R. (2004, May). Balancing agility and discipline: Evaluating and integrating agile and plan-driven methods. In *Software Engineering, 2004. ICSE 2004. Proceedings. 26th International Conference on*(pp. 718-719). IEEE.

[22] Southwell, K. (2002). Agile process improvement. *TickIT International Journal*, 3-14.

[23] Coram, M., & Bohner, S. (2005, April). The impact of agile methods on software project management. In *Engineering of Computer-Based Systems, 2005. ECBS'05. 12th IEEE International Conference and Workshops on the* (pp. 363-370). IEEE.

[24] Highsmith, J. A. (2002). *Agile software development ecosystems*. Addison-Wesley Professional.

[25] Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. *Communications of the ACM*, *48*(5), 72-78.

[26] Dyba, T., & Dingsoyr, T. (2009). What do we know about agile software development?. *Software, IEEE*, *26*(5), 6-9.

[27] Söderlund, J.; Geraldi, J., Classics in project management. Revisiting the past, creating the future,International Journal of Managing Projects in Business, Vol. 5,Issue 4, pp. 559–577, 2012.

[28] Ribeiro, R. L. O.,Gerenciamento de Projetos com PRINCE2, Rio de Janeiro: Brasport, 2011.

[29] Project Management Institute – PMI, PMBOK Guide: A Guide to the Project Management Body of Knowledge, Pennsylvania: Project Management Institute, 5th ed, 2013.

[30] Hass, K. B., The Blending of Traditional and Agile Project Management, PM World Today, Vol. 9, Issue 5,pp. 5-07, 2007

[31] Agile, Manifesto for Agile Software Development, 2011.

[32] Augustine, S., Payne, B., Sencindiver, F., Woodcock, S., Agile Management: Steering From the Edges, Communications of the ACM - Thesemantic e-business vision,Vol. 48, Issue 12, pp. 85-89, 2005.

[33] Highsmith, J.,Agile Project Management: creating innovative products, Addison- Wesley: Boston, 2009.

[34] M. A. Awad, A Comparison between Agile and Traditional Software Development Methodologies, *School of Computer Science and software Engineering,The University of Western Australia, 2005.*

[35] Mehar Ullah, Comparison and problems between Traditional and Agile software development methods, Department of Master Degree Program in Computer Science, Lappeenranta University of Technology, November 13, 2014

i