

AGILE SOFTWARE DEVELOPMENT AND ITS SUCCESSFUL FACTOR: A LITERATURE REVIEW

Raheela Nasim*

Zaib-u-Nissa**

Tasleem Mustafa***

Abstract:

Agile software development methodologies is an incremental and iterative approach widely accepted these days. Self organizing teams, contribution of clients in development process, high quality projects, less documentation bundles and reduced time to market are some properties of agile development. Traditional software development processes are not well-organized and efficient method when requirements change frequently. A systematic literature review of ASD and its success factors present in this paper. Study of existing literature, identify a list of possible critical success factors of agile development. Quality, scope, time and cost are four success dimensions of software. Agile methods may produce software faster but we also need to know how we develop best quality software in less time and cost.

Keywords -Agile Software Development ASD, Success Factors

* Lecturer department of computer science, University of Agriculture, Faisalabad

** Department of computer science, University of Agriculture, Faisalabad, Pakistan

*** Chairman department of computer science, University of Agriculture, Faisalabad, Pakistan

Introduction:

The use of agile software development model attain quickness because of its some valuable properties like self organizing teams, contribution of customer in development process, high quality projects, less bundle of documentation and reduced time to market [4].

There was a bulk of documentation in ancient techniques for software development. In ancient development methods the thinking of project managers was that the gathering of requirements and their specification in early stages because scheduling and design could save a high amount of cost, resources and time for development of project as compared to the changes in later stages which cause a lot of increase in cost, scope and time [5].

These ancient approaches did not agreed to the changes in late phases e.g. during implementation and confirmation, many software under these approaches still unsuccessful to meet their mission because of quantum of necessary specifications and the demand of a bundle of documentation. But ASD approach presents the solution to these issues by the facility of changes integration and close customer collaboration in development process and with development teams. It also presents the concept of iterations that can integrate changes in requirements even in the stage of implementation.

In this research, there is a detailed study of different agile methodologies and its success factors that affect the quality, scope, time and cost of projects. These methodologies can be different or may be a mixture of two or more customized methodologies use according to organizational environment and project requirements.

Agile Software Development Methodologies:

Agile Methodologies (AM) is an incremental and iterative methodology for efficient modeling and documentation of software systems. Objective behind the use of Agile Methodologies is the production of high quality software in less time and less cost. The purpose behind the development of AM was to reconstruct the traditional development process and take away the hurdles to accept changes in requirements during development phases [1]. There is no restriction in AM to lock the industry requirements and design descriptions to be fixed for development period.

Some trendy agile development models are Scrum, Extreme Programming (XP), Feature Driven Development (FDD), Crystal Methods, Test Driven Development and Dynamic Systems Development Method. All these models work in a different way from ancient software development models and facilitate the organizations to face the challenges of market [3].

1. Extreme Programming

XP can get better quality programming while reduction delivery schedules. XP based on conception and practices or performance including the client's collaboration with the developers, pair programming, shared ownership in coding and the use of images to portrait the business conditions [6]. Other XP doctrines are: developing tests before developing code, managing an open workspace, rebuilding code, and work to 40 hrs in a week. XP include practices that are newly for many developers. Pair programming, the 40 hour workweek and open workspaces may lead to disagreement from developers and management. Extreme Programming model is one of the major pillars of ASD framework [2,3].

In short XP is basically designed for the team size between 3 to 10 but the team is enhanced with one or more stakeholders to supply architectural inputs and to work as coding counselor. The development team in XP can be located in one or more contiguous rooms. After the completion of iteration a running software version is released for user testing. A set of user stories produced at a planning meeting is a programming process held at beginning of all iteration. We can say that meetings are the pool of stories where next iterations are finalized on the priority of their functions. In meetings the order of priorities can be changed when needed [2].

It is a key point that client is a creator of the user stories. Each user story has a significant importance as it is a small crumb of functionality for final product gathered by the end user point of view. When it is gathered by the client, the developers purify the user story with the aspect of estimating progress required to implement the functionality in separation of all other stories [11]. The ideal situation for user story is that it must be capable to be implemented in two to three days as it is a crumb of functionality but if the estimate is longer than the developer divides it into series of smaller pieces if it is necessary. When user stories are collected and agreed then it is again the customer who settled the sequence of priority that dictates the flow of implementation.

The customers or stakeholders fix the business worth of each story and which are most worthy for next release. After finalizing the set of stories for next release the time is estimated which indicate when that release can be projected. In this way the time required for next iteration is estimated. In all scenarios the client performs a leading role for decision making and the developer provides all important estimates [6].

2. Scrum

Scrum methodology is used for frequently changing requirements. Scrum word is derived from the method used in 'rugby game'. By using scrum technique project is dividing it into "sprints" [8]. In scrum more concentration is on process development then coding. It helps creating friendliness in development team. Best thing about scrum is that it maximizes the productivity. It is flexible in a way that it can be implemented to single project as well as to whole organization.

Scrum can be applied to one small project and can be applied to large project by dividing it into small subprojects [2].

3. Crystal Clear

Crystal clear is an agile software development methodology was developed by Alistair Cockburn. It is a lightweight methodology that applied on that software which is not life critical and having 6 or 8 co-located team members. It focuses on people not on tools or processes. Crystal clear can be applied on small projects which are not life critical [2].

4. Feature Driven Development FDD

FDD consist of 5 steps and there is no need of training to use it. In first step develop a model of whole project, second list required features and third prioritize the main features for implementation plan. In fourth and fifth step development iterations occur [2]. Feature is design and released, priority of feature catalog is changed so that the developers continually develop the highest priority features. FDD work with XP technique very efficiently. Fourth and fifth step can be time boxed in FDD for better development and customer satisfaction [6].

5. Test Driven Development (TDD)

Test Driven Development (TDD) is a methodology to develop software in which the actual development process is done in iterations after writing the test. Development of software based on these tests. Every iteration generate code essential to pass that iteration's tests. Finally, the developer or team rebuilds the code to provide accommodation any changes [2].

Implement of new feature firstly arise a question that whether the existing design is the best possible design that enables to implement these functionality. The quality of your feature design is improved and makes it easier to work with in the future [10].

6. Dynamic Software Development Method DSDM

Dynamic Systems Development Methods DSDM is an agile methodology for developing software. It is an incremental and iterative framework that draws attention to continuous customer involvement. Adjusting the changing requirement and delivering the software systems on time and on budget is the ultimate goal of the DSDM, along the development process. DSDM is about people, not tools. Actual understanding of a business requirements, software delivering process that work and deliver as quickly as possible [9].

In DSDM divide projects in three steps: pre-project, project life-cycle, and post project. Nine rules of DSDM: user participation, frequent delivery, addressing current business requirement, empowering the project team, incremental and iterative development, before project starts high-

level scope being fixed, allow for reversing changes, testing during the lifecycle, and effective and efficient communication [6].

Success Factors of Agile Software Development:

According to Quality, Cost, Time, and Scope the success of Agile Methodology depend on management obligation, friendly environment in organization, pleasant project team environment, high quality team capability, strong user involvement, systematic project definition process, style of software engineering techniques, style of project management process, right delivery plan, Non life serious project nature, changeable scope project type, and dynamic, accelerated process schedule [7].

These factors categories in to five main types such as

- Organizational factor
- People factor
- Technical factor
- Process factor
- Project factor

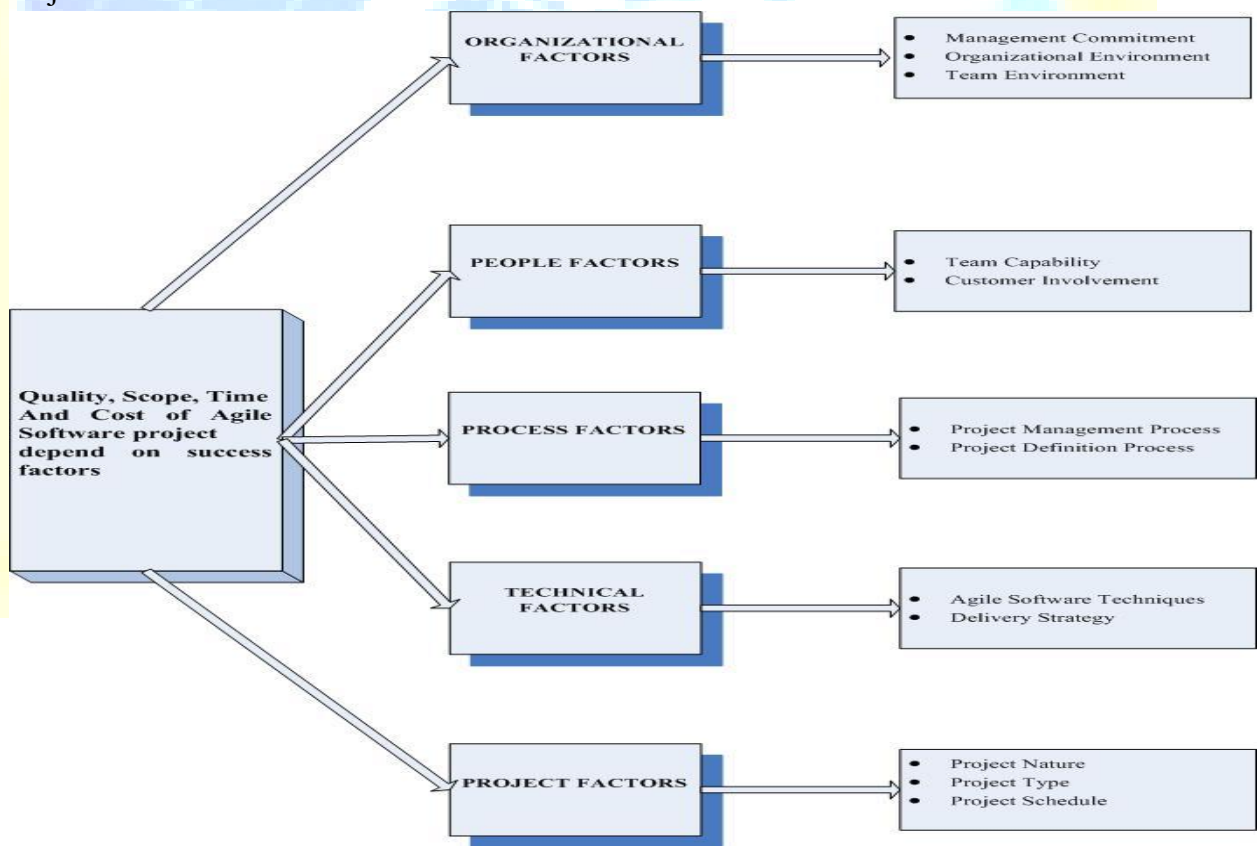


Figure1: Success Factors

1. Organizational factor:

Strong Management obligation, pleasant Organizational and Team Environment plays important role on project quality. Collocation of the entire team, self organizing and coherent teamwork and friendly environment develop the efficient and well organize software [7].

2. People factor:

In software development two types of people involve development team and customer. High competence and capability of team members achieve their goal with great enthusiasm and knowledge. Managers who have adaptive agile management style and trained technically his team improve the quality of software with less cost and time. Agile methodology support continues contribution of customer in development process. Friendly customer relationship, Strong customer obligation and presence improve the project quality [7].

3. Technical factor:

Some technical factor like regular delivery of Software, Delivering prioritize features first, understandable coding principles, Pursuing easy design, accurate rebuilding activities, accurate quantity of documentation and exact integration testing effect the quality, scope, time and cost of project [7].

4. Process factor:

Management process of agile-oriented requirement, project management method, management process of configuration, high quality progress tracking mechanism, powerful communication and contact through daily face to face meetings and honoring regular work plan improvement the development procedure [7].

5. Project factor:

Small and simple project require less affect to achieve goal as compare to large and complex. Quality of project depends on project nature, type and schedule [7].

Conclusion:

The use of agile software development model attain quickness because of its some valuable properties like self organizing teams, contribution of customer in development process, high quality projects, less bundle of documentation and reduced time to market. Many direct or indirect factors involve the success of ASD. In this paper some possible critical success factors of agile methodology were identified on the bases of literature review. Quality, cost, time and scope are four success dimensions of software. Team capability and environment, customer participation, project management process and engineering methods, project nature and software delivery are some major factor that improve the quality of project with less time and cost

References

- [1] Schatz, B. & Abdelshafi, I. Primavera gets agile: A successful transition to agile development. *IEEE Software*. 22(3). 2005
- [2] Livermore, J. A. (2007, March). Factors that impact implementing an agile software development methodology. In *SoutheastCon, 2007. Proceedings. IEEE* (pp. 82-86). IEEE.
- [3] Boehm, B. & Turner, "R. Management challenges to implement agile processes in traditional development organizations" *IEEE Software*. 22(5), 30-40. 2005.
- [4] Dingsøyr, T., Nerur, S., Balijepally, V., & Moe, N. B. (2012). A decade of agile methodologies: Towards explaining agile software development. *Journal of Systems and Software*, 85(6), 1213-1221.
- [5] T. Stober, U. Hansmann. "Overview of Agile Software Development", Agile Software Development: Best Practices for Large Software Development Projects, Springer. 2010.
- [6] Dybå, T., & Dingsøyr, T. (2008). Empirical studies of agile software development: A systematic review. *Information and software technology*, 50(9), 833-859.
- [7] Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, 81(6), 961-971.
- [8] Mann, C. & Maurer, F. A case study on the impact of scrum on overtime and customer satisfaction. *Proceedings of the Agile development Conference (ADC'05). Denver, CO.* 70-79. 2005.
- [9] Mead N. R., V. Viswanathan and D. Padmanabhan, 2008. Incorporating Security Requirements Engineering into the Dynamic Systems Development Method, *COMPSAC 32nd Annual IEEE International Computer Software and Applications Conference*, pp: 949-954.
- [10] Scott, W. Ambler, "Introduction to Test Driven Design (TDD)" Available: <http://www.agiledata.org/essays/tdd.html>
- [11] Rees, M. J. (2002). A feasible user story tool for agile software development?. In *Software Engineering Conference, 2002. Ninth Asia-Pacific* (pp. 22-30). IEEE.