ISSN : 2231-0711

Assessment of Manifestation of Agile Methodology and Computing

T.Venkat Narayana Rao[#1], D.Hiranmayi [*2], G. Shravani Priya [#3], G.Ravi[#4]

Professor[#1], Post Graduate Scholars [# 2#3], Assistant Professor[#4]
Department of Computer Science and Engineering
Sreenidhi Institute of Science and Technology
R.R District, T.S, India

***Abstract*—Agile methodology is a modern approach in software process to achieve a common business goal. In today's business environment, normally organizations choose to deliver the projects very fast. They also require adjusting as well as adapting to changing environments. In this regard, traditional software methodologies fail to achieve these requirements. Traditional software mechanisms are sequential which takes much time to deliver the projects. Though, some of the traditional methods are used to deliver the software projects faster, agile methodology have some new challenges in delivering the software products and satisfies the customers by the valuable accurate software products. Agile software methodologies are an incremental and iterative approach used in delivery of accurate software product and assist to respond unpredictability. As numerous areas are overblown, there is huge requirement of applying the agile methodology in software process. This paper focus on study of agile methodologies, a comparative study between traditional methodologies with agile methodologies and process involved in agile methodologies . This paper also emphasis on the best practices of agile methodologies and its exemplars to the readers i.e. how to develop software product in a innovative and agile means and yet with a enhanced business advantage outcome.**

***Keywords*—Agile, Iterative, Traditional methods, Extreme programming, scrum, Feature driven development.**

I. Introduction

Agile methodology is a software development methodology used to improve the speed in delivery of software projects [4]. The majority of the organizations use agile methodology over the past few years. In the past years most of the organizations like Google, Microsoft and Siemens have adopted agile methodology in some major projects.

A. Definitions

The definition of agile methodology varies with context and resources. Some resources defined agile methodology as: "An iterative and incremental (evolutionary) approach to software development which is performed in a highly collaborative manner by self-organizing teams within an effective governance framework with service that produces high quality software in a cost effective and appropriate manner which meets the changing needs of its stakeholders". It is also defined as , "A belief and a methodology which is evolved as an response to the long growing annoyance of the waterfall SDLC concepts". This term proposes an iterative approach to software development using shorter and lightweight development cycles and some diverse deliverables [2].

B. The meaning of agile

The term "agile" is taken from the dictionary word called agility which means "moving swiftly and easily". Agile Computation is a logical depiction used to process the software more swiftly and naturally. At each stage of a software development life cycle the process includes additional requirements and make changes according to the requirements. As most of customers are less able to articulate their requirements initially, hence traditional approaches fail to satisfy their requirements. Attainment of agile methodologies is very useful because these methodologies are iterative as at the end every step in the software life cycle these methods include iterations and review patterns about the software requirements. Most of the agile methodologies like scrum, extreme programming, crystal, and dynamic system development methods are increasingly evolving as they are iterative and unpredictable positive approaches. The programmers or developers of the software should be continuously interacting with the customers at the end of each iteration. Agile methodology is time bonded approach and develops the software incrementally by adding additional requirements from the starting stage of the project instead of adding all the requirements at the end. Agile Computing Methodology involves phases called sprints or iterations as shown in figure 1. Each sprint has its unique and own duration and deliverables. Deliverables are defined, prioritized by the customers. If work is not completed or customer need to add some additional requirements to the project, the sprint is repeated and incremented and the future sprint is planned. Work completed by each sprint is repeated and evaluated by the customers, since customers play a major role in software development. As the requirements of customers are dynamic there should be a proper interface with the customers at each stage of software development.\

TABLE I. Comparison of traditional approaches with agile

Issues	**Traditional approach**	**Agile**
Development life cycle	Sequential or linear.	Iterative and incremental
Style of development	Anticipatory	Adaptive
Requirements	Knowledgeable early, stabled, clearly defined and documented.	Emergent, rapid change discovered during the project.
Architecture	Heavy weighted architecture for current and future requirements.	YAGNI precept (you are not going to need it).
Management	Process centric, command and control.	People centric, leadership and collaboration
Goal	Predictability and optimization.	Exploration or adoption.
Team organization	Pre-structured teams	Self-organized teams
Measure of success	Conformance to plan	Business value delivered.

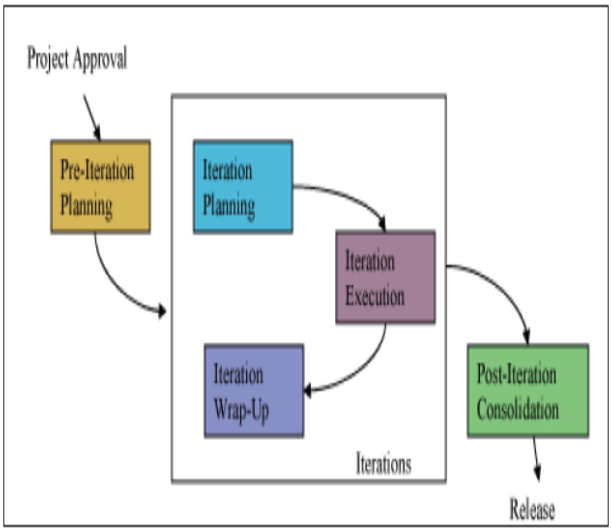


Figure 1: Agile Computing Process

C. Overview of agile system

Agile methodologies chiefly include the relationships and communality of software developers. Software developers include programming practitioners, testers, managers etc. Agile methodologies comprise of the relationships and proper communication between various entities of the process. To develop software more accurately and quickly there should also exist a cooperation and good communication between the customers and developers of the software. Software teams should communicate at the end of each sprint of software development and they should collect the responses from the customers. If the customer want to alter or add additional functionalities, then the developer have to rebuilt the software on incremental basis. Agile methodology is commonly known as empirical process. Empirical process is used in research oriented, high change, unstable and rationally intensive domains which require constant monitoring and exploratory work [7]. These conditions impose short "inspect-and-adapt" cycles, regular and short feedback loops. The short inspect-and-adapt cycles are prominent in agile mechanism that would help development teams to handle the conflicting and unpredictable demands of some projects emerging time to time.

II. TRADITIONAL VERSES AGILE APPROACHES

A .Issues with traditional approaches

The traditional software methodologies are plan driven and follow sequential approaches[1]. In traditional software methods the requirements of software are gathered at an early stage of development cycle and according to those requirements the software can be developed. According to customer requirements planning, modelling, coding and testing phases are carried out. But it is not possible for a customer to change the requirements or add some additional functionality. Once the requirements are taken the developer doesn't have interaction with the customers. Customer does not have the knowledge about how much amount of process is concluded until the software is developed wholly. The most known sequential approach is waterfall method. The table I. Illustrates and Compares sequential approach and agile approaches on the technical issues and parameters.

Sequential approaches fail to develop the accurate software products. Sequential approaches take much time to develop the products. This problem could be resolved by iterative approaches like spiral model. Spiral model is a traditional iterative approach which has a iterated pattern until the end product is developed. But this model takes three to four months to complete all iterations. By this model the software obtained is accurate and would be satisfactory to the customer but the duration of the process is very high.

B. Why is agile computing inevitable?

All the traditional methods existing for software development are not fulfilling the needs of customers or clients particularly in terms of time or requirements. Hence, to overcome these problems agile methodology is useful. Agile methodology is an iterative and has flexible incremental approach.

C. Waterfall versus Agile

1) Waterfall method

Waterfall method is a sequential approach used in developing of software. The approach include each stage to be in a sequence and move to further stages. After the completion of all stages only it will move on to the next iteration [8]. Waterfall method include following stages in development of software:

- Gathering of requirements
- Design
- Coding
- Testing
- Fixing of issues raise
- Delivery of product

2) Agile method

Agile method is an iterative and incremental approach in which every stage of software development process is repeated and incremented. At the end of each stage it review and incrementing is done based on review status. The figure 2 and table II depicts differences between waterfall and agile models.

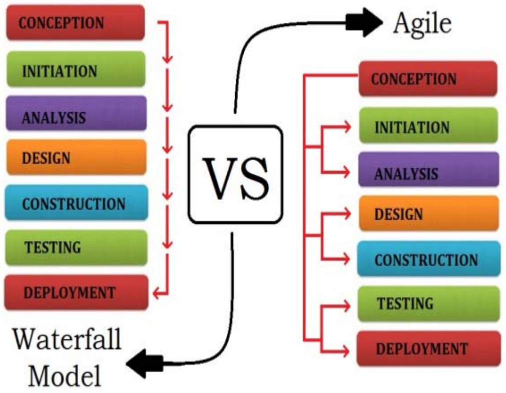
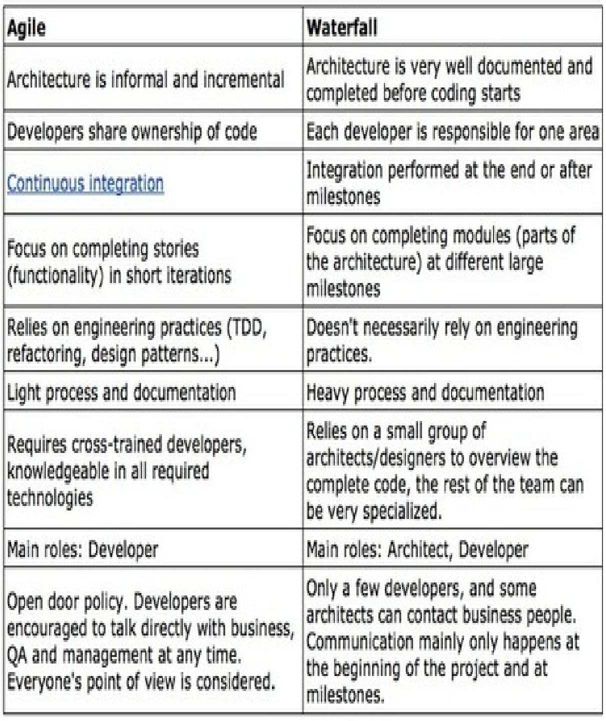


Figure 2: Comparison of processes involved in waterfall and agile.

TABLE II. Comparison of agile and waterfall with different functionalities



Agile	Waterfall
Architecture is informal and incremental	Architecture is very well documented and completed before coding starts
Developers share ownership of code	Each developer is responsible for one area
Continuous integration	Integration performed at the end or after milestones
Focus on completing stories (functionality) in short iterations	Focus on completing modules (parts of the architecture) at different large milestones
Relies on engineering practices (TDD, refactoring, design patterns...)	Doesn't necessarily rely on engineering practices.
Light process and documentation	Heavy process and documentation
Requires cross-trained developers, knowledgeable in all required technologies	Relies on a small group of architects/designers to overview the complete code, the rest of the team can be very specialized.
Main roles: Developer	Main roles: Architect, Developer
Open door policy. Developers are encouraged to talk directly with business, QA and management at any time. Everyone's point of view is considered.	Only a few developers, and some architects can contact business people. Communication mainly only happens at the beginning of the project and at milestones.

III. Agile Process and Principles

A. *Process:* Agile development involve following steps to develop a software:

1) *Development is iterative:* Agile development is iterative and incremented that means, at each and every step the process is iterated. Requirements are gathered at the end of each step and the process should be reviewed and increments the functionality of process. The requirements and architecture of the process are continuously changing until the concluding product is developed.

2) *Planning is adaptive:* Planning is done following gathering the requirements. But whenever it is difficult to process according to the planning then the planning can be changed according to the process thus making the planning very flexible.

3) *Roles blur:* Agile process does not have specific roles to the persons. In traditional approaches roles are given to the employees according to the skills they have. But in agile roles can be blur, and roles may change dynamically while developing the projects.

4) *Scope can vary:* In agile methodologies the scope of project can change according to the customer requirements. Duration of the project development is also flexible and can vary while developing the software. Agile development team integrates their plans, and works in groups to enable flexible budget.

5) *Change in requirements:* At each stage developers would interact with the customers and thus customer can add or remove the requirements based on their intent. Hence, the requirements are always dynamic in agile methodology. Customers need not worry about developer's threat or organizations budget.

B. *Principles:* Agile manifesto has documented the prime principles it includes [6][3]. There are around twelve key principles in the agile manifesto:

- Offers top priority to satisfy the customer's through the delivery of continuous and valuable software.
- Though the development cycle is late but agile methodology grants more importance to the customer requirements and their competitive advantage.
- Deliver working of software frequently from weeks to months, so that customer would get more confidence on the product.
- Business people and developers must work jointly to develop a precious product.
- Build projects around enthused individuals and maintain them to deliver correct product.
- To be effective by allowing communication between developers with customers.
- Working software is the perfect measure of software product.
- Agile process let sustainable development, allowing a steady speed to be maintained.
- Continuous attention towards industry for enhancing agility.
- The art of maintaining the work done should simple and vital.
- The best architectures, requirements and designs emerge from self organizing teams.
- At regular intervals team should be effectual and return according to the changes.

IV. Examples of Agile Methodologies

The most popular software development methodologies which are using agile are scrum, extreme programming, lean development, crystal methods and dynamic systems development methods. These five methods follow the key principles and processes of agile methodology.

A. *Scrum Methodology*

Scrum methodology is a lightweight agile framework which includes proper applicability of agile principles. Scrum maintains effective iterative and incremental approach in all types of software development life cycles [5]. In scrum methodology, product owner works with team to recognize the system functionality and prioritization of functional units along with the product backlog. Product backlog consist of bug fixes, non-functional requirements etc. It provides a facility offered by the product developers to work according to those priorities and user delivery priority during each sprint within 30 days. Once the product backlog is delivered, the software team reprioritize the sprint and move to commence other sprints with earlier functionalities; hence, the process is iterated at every sprint as shown in the figure 3. Once the product backlog is fully delivered no additional functionalities are permitted to be added by the customers.

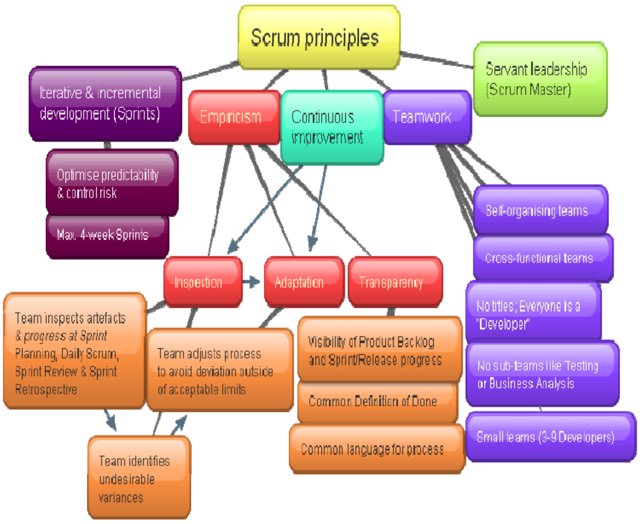

Scrum principles
Iterative & incremental development (Sprints)
Optimise predictability & control risk
Max. 4-week Sprints
Empiricism
Continuous improvement
Teamwork
Servant leadership (Scrum Master)
Self-organising teams
Cross-functional teams
No titles; Everyone is a "Developer"
No sub-teams like Testing or Business Analysis
Small teams (3-9 Developers)
Inspection
Adaptation
Transparency
Team inspects artefacts & progress at Sprint Planning, Daily Scrum, Sprint Review & Sprint Retrospective
Team adjusts process to avoid deviation outside of acceptable limits
Team identifies undesirable variances
Visibility of Product Backlog and Sprint/Release progress
Common Definition of Done
Common language for process

Figure 3: Scrum methodology and its process

B. *Extreme programming*

Extreme programming is powerful and emerging software agile methodology [9]. It is an appropriate method to deliver the software very quickly and continuously. It promotes continuous testing, continuous planning, high customer involvement; rapid feedback loops and delivers software process work continuously at small intervals (1-3 weeks). Xp involves mainly-simplicity, communication and feedback as shown in figure 4.

This approach provides following supporting practices:

- *Planning Game:* clientele and developers interact with each other and develop the story regarding the projects.
- *Small Releases:* Every release should be small and contain only business requirements.
- *Customer Acceptance Tests:* Customers should agree to the design and coding primarily.
- *Simple Design:* The developed release should be as simple as possible.
- *Pair Programming:* Instead of periodic reviews two practitioners review their codes and optimize codes accordingly.
- *Test-Driven Development*: Test cases should be developed and required changes can be made.
- *Refactoring: Deals with* optimization of internal coding and architecture.
- *Continuous Integration*: The developed code should be frequently integrated.
- *Collective Code Ownership*: The code developed should be associated to the single owner.
- *Coding Standards*: While developing the code some standards must be followed.
- *Metaphor:* Establish collaboration between developers and customers.
- *Sustainable Pace*: Developers put in the best efforts once they are committed.

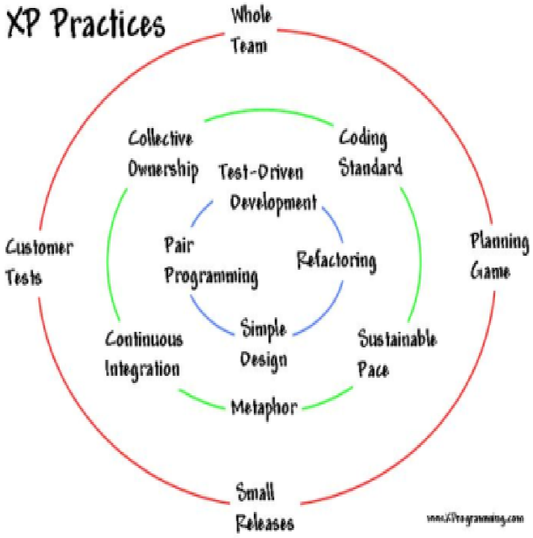


Figure 4: Extreme Programming

C. *Lean development*

Lean software development methodology focuses on delivering value to the customers, and on the efficiency of the "value stream" the mechanism delivers value stream. The main principles of lean development involve:

- *Eliminating Waste*: Eliminates waste through selecting only valuable and prioritize them and release in small batches.
- *Amplifying Learning:* Learn in developing o software products and optimizing them.
- *Deciding as Late as Possible:* Decision should depends on individuals hence late.
- *Delivering as Fast as Possible:* Deliver the products as fast as possible.
- *Empowering the Team:* Lean uses team resources.
- *Building Integrity In:* Built integrity between teams.
- *Seeing the Whole:* concentrates on entire product while developing the individuals.

D. *Crystal methods*

Crystal is a comprise model of agile family methodologies such as crystal yellow, crystal clear , crystal orange and others whose exclusive characteristics are determined by team size, system criticality and project priorities. Crystal family addresses the recognition that each project should involve different set of policies and practices to reach their unique characteristics. Crystal model also includes teamwork, communication and simplicity as well as manifestation of frequently adjustment with respect to the requirements and process. Crystal Methods are the collection of methodologies based on two fundamental assumptions:

(1) Teams can restructure their process as they work and become a more integrated and optimized.

(2) Projects are unique and active and require methods that are exclusively designed for each endeavour.

E. Dynamic system development methods

DSDM is defined as: ”A project delivery framework that assists in the development and delivery of business solutions to firm timescales and fixed budgets.” The DSDM Manual narrates that "the current step needed to be completed only sufficient to move to the next step," this gives a good representation of the incremental, iterative approach common to agile methods. DSDM is different from all agile methodologies in which members paying yearly fee to remain as member and able to access the licensed methodology. DSDM is an intertwined set of processes, each of which is iterative and incremental. While the model is complex and discouraging, it is fundamentally a three-step iterative model, consist of Modelling, Design-Build and Implementation phases.

DSDM requirements are prioritized using MoSCoW rules:

M – Must have requirements
S – Should have if at all possible
C – Could have but not critical
W - Won 't have this time, but potentially later.

V. ADVANTAGES OF AGILE METHODOLOGIES

Agile methods have paved the dynamic path of software professionals who have experienced the challenges and limitations of traditional approaches. Agile methods help the organizations to reduce the risks in the entire software development life cycle. Agile methodologies have more advantages. The advantages of agile methods include:

- High customer approval through rapid continuous delivery of results of the development.
- Interactions of customers and developers give more\importance rather than process and tools. This leads to make accurate and valuable software.
- Working on software delivered more frequently which increases customer confidentiality towards the product.
- Agile methodologies embrace face to face conversations which gives more clarity to the developers as well as customers.
- Daily mutual aid between developers and business people who are managing the software process.
- Continuous awareness to technical skills and good design.
- Regular adaption to changing technologies and requirements of customers.
- Customer can change requirements late. It allows customers to change the requirements during any phase of the development by which customer can get most valuable product.

VI. CONCLUSION

Innovative software development methodologies have been evolving since the 1980s. Agile methodologies came into focus by the virtue of need for holding changing requirements in terms of resources, skills and environment etc. Agile methodologies offer some practices that aid communication among the developer and the customer, and undergo a cycle of develop-deliver-feedback. The core endeavour of agile environment is to deliver what is needed when at a appropriate time. Agile methodologies include a set of software development approaches. The main agile methodologies that are discussed in this paper are XP, Agile Modelling, and SCRUM etc. Though, popular but agile methodologies are not best suitable for some projects, where communication between the developer and the customer is difficult or when team is young. Thus, much planning and emphasis is needed to adopt and apply these methodologies. This paper describes agile development methods, the process included in agile development and a detail account of comparative study between traditional approaches and agile methodologies. When there is a big chance for misinterpretation in customer’s requirements, deadlines and budgets are rigid, then Agile methodologies are among the best software development approaches to be considered.

REFERENCES

[1] MARK KENNALEY, SDLC 3.0: BEYOND A TACIT UNDERSTANDING OF AGILE, FOURTH MEDIUM PRESS,2010.
[2] GatherSpaceTeam, “Agile software development,” Retrieved January 15, 2011, January 2011.
[3] Barabino, G., Grechi, D.,Tigano, D., Corona, E., &Concas, G. (2014). Agile Methodologies in Web Programming- XP
[4] Vijayadevi.; Traditional and agile methodologies: An interpretation(2013).
[5] Capers Jones., Evaluating Agile and Scrum with Other Software Methodologies.(2013)
[6] Beck, Kent; et al. (2001). "Manifesto for Agile Software Development" Agile Alliance. Retrieved 14 June 2010.
[7] Beck, Kent; et al. (2001). "Principles behind the Agile Manifesto". Agile Alliance. Archived from the original on 14 June 2010. Retrieved 6 June 2010.
[8] Kimberley Chan on January 31, 2013 Waterfall vs. Agile.
[9] K. Beck and M. Fowler, Planning Extreme rogramming, Addison Wesley Longman, Reading, Mass., 2001.

BIOGRAPHIES:

Professor T.Venkat Narayana Rao, received B.E in Computer Technology and Engineering from Nagpur University, Nagpur, India, M.B.A (Systems), holds a M.Tech in Computer Science and Engineering from Jawaharlal Nehru Technological University, Hyderabad, A.P., India and a Research Scholar in JNTUK. He has 23 years of vast experience in Computer Science and Engineering areas pertaining to academics and industry related I.T issues. He is presently working as Professor, Department of Computer Science and Engineering, **Sreenidhi Institute of Science and Technology,** Ghatkesar , R.R.Dist.,T.S, INDIA. He is nominated as an Editor and Reviewer to 38 International journals relating to Computer Science and Information Technology and has published 72 papers in international journals. He is currently working on research areas, which include Digital Image Processing, Digital Watermarking, Data Mining, Network Security and other emerging areas of Information Technology. He can be reached at tvnrbobby@yahoo.com.

Dhappuri Hiranmayi is a P.G Scholar in Technology from Sreenidhi Institute of Science and Technology in the branch of Computer Science and Engineering. She has developed in projects related in java, Android apps and web Designing. Her research interests include Data mining, security issues related to data mining as well as software processes. She can be reached at hiranmayidhappuri.2.9@gmail.com.

Gadasu Shravani Priya is a P.G Scholar in Technology from Sreenidhi Institute of Science and Technology in the branch of Computer Science and Engineering. She has developed projects in java, perl and web designing. Her research interests include language translations and emerging technologies. She can be reached at priyashravani@gmail.com .