

Need of Agile Development

Rupinder Singh, Amandeep Bakshi

Abstract— There are numerous developing methods in the manufacturing and software development industries which are broadly classified into traditional and agile methods. This paper reviews the important aspects of traditional and agile methods of software development. It also explains the properties of both methodologies by making comparison between them. We will discuss the flaws and challenges faces by traditional development processes and how will agile overcome them which leads us to the conclusion that agile have an edge upon traditional methods.

Index Terms— Agile, SDLC, Traditional methods, Iterative approaches

I. INTRODUCTION

In any industry whether it is some manufacturing company or Software industry there is some define processes, tools and technique to develop the project. The success of the project or product with high or low profit is totally dependent on these processes [1]. In IT industries there are so many processes are following by Software Development companies but among all of them in recent years, There has been much discussion is going on in the software press about ‘agile methods’.

Agile processes have the capacity and quality to bear the change in requirement in Software specification at any time of SDLC (Software Development Life Cycle). By adopting the Agile approaches organizations can respond swiftly to customer needs, speed solution delivery and adapt efficiently to change & also able to make more money as compare to traditional software development models like Projects will become faster – early to market, faster product evolution, Cash flow starts earlier, profits are higher, the customer learns by using the product in the market which keeps commitments of IT to become trustworthy.

Agile software development aims to develop and implement software quickly in close cooperation with the customer in an adaptive way, so that it is possible to react to changes set by the changing business environment and at the same time maintain effectiveness and efficiency. Steps to accomplish this are to put an emphasis on tacit knowledge and sharing it by frequent face-to-face communication and by concentrating on producing working software instead of documentation. Agile Manifesto and agile methods claim to provide guidelines for such development efforts.

II. TRADITIONAL METHODS

Every software engineering organization should describe a unique set of framework activities for the software processes they adopts. they should have to populate each framework activity with a set of software engineering actions, and define each action in terms of a task set that identifies the work (and work products) to be accomplished to meet the development goals. It should then adapt the resultant process model to accommodate the specific nature of each project, the people who will do the work, and the environment in which the work will be conducted. Regardless of the process model that is selected, software engineers have Traditional chosen a generic process framework that encompasses the following framework activities like as communication, Planning, Modeling, Constructing and Development [2].

So Software developers & gurus examine a number of perspective software process models. We call them “prescriptive” because they prescribe a set of process elements framework activities, software engineering actions, task, work products, quality assurance and change control mechanisms for each project. Each process model also prescribes a workflow—that is, the manner in which the process elements are interrelated to one another [2].As a result traditional model like Waterfall, Incremental and Evolutionary model came to existence and are classified into the heavyweight methodologies [1]. The traditional software development methods are dependent on a set of predetermined processes & on-going documentation which is written as the work progresses and guides further development

But in practice there are some limits and draw back in traditional methods like Traditional Methods are presumptuous [2]:

1. Assume all aspects can be defined prior to the start of work.
2. Requirements are stable.
3. Technology is well known to the team and matures in its implementation.
4. There will be no surprises, no changes, and no deviations.
5. Enforces specific organizational structures
6. Up front design cannot accommodate change
7. Escalation can not affect Product Development.

In practice, traditional process model causes the development teams to struggle to specify requirements fully and correctly up front, in the early stages of the project, before they have finished learning. The majority of change requests happen near the end of the testing phase when it is very expensive and time consuming to do the rework. To prevent this, many changes request are therefore rejected in order to save work and to protect the project’s delivery dates.

Revised Manuscript Received on 30 March 2013.

* Correspondence Author

Rupinder Singh, Computer Science and Engineering Department, Thapar University, Patiala, India.

Amandeep Bakshi, Computer Science and Engineering Department, Thapar University, Patiala, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Need of Agile Development

PMBOK [3] recommends that 28% of an effort should be dedicated to planning, prior to any design or development, because it provides false comfort in a plan, “Planning cannot define reality”, “Planning cannot control reality”, regardless of how detailed it is. So Waterfall no longer solves our problems. The Traditional methodologies were good at managing the known but they are terrible at managing the unknown conditions & actions.

III. AGILE PROCESS METHODS

Agile process methodologies evolved out of a need to address the ominous problems facing by software industry due to the failure of traditional process models to tackle the unknown and unpredictable needs, at the later end of development. Agility is dynamic, content specific, aggressively change embracing, growth oriented and value-driven approach that allows Project Managers to deliver high-priority, high-quality work—and look like rock stars to their stakeholders.

In agile development, rather than following a single large process model that have implemented in conventional SDLC, the development life cycle is divided into smaller parts, called “increments” or “iterations”, in which phases within a development life cycle are revisited over and over again [4]. According to Agile Manifesto [5], the major factors of agile factors include the following four:

1. Early customer involvement
2. Iterative development
3. Self-organizing teams
4. Adaptation to change

In the existing era of Agile process models the current agile software developments methods are Extreme Programming, Scrum, Crystal family of methodologies, FDD (Feature Driven Development), Rational Unified Process, Dynamic System Development method, Adaptive Software Development and Open Source Software Development method[6]. Among all the methods Scrum are widely used in industries because it does not require any specific engineering practices, it can be adopted to manage whatever engineering practices are used in an organization.

IV. TRADITIONAL VS AGILE

Major difference between agile development and traditional method is the ability to successfully deliver result quickly and inexpensively on complex projects with imperfect requirements. Agile methods stress on teams, working software, customer collaboration, and responding to change while the traditional methods stress on contracts, plans, processes, documents, and tools with a predefined set of rules and believes that there is no such undefined requirements. Unlike waterfall, agile project management continuously evaluates time and cost as primary constraints.

In traditional waterfall project management, the project manager is burdened with balancing project scope, cost, quality, personnel, reporting, risk, and adapting as requirements change. But Agile Project Management divides these overwhelming project management responsibilities among three agile roles:

1. The Product Owner handles, setting project goals, handling the tradeoff of schedule versus scope, adapting

to changing project requirements and setting priorities for product features.

2. The Scrum Master guides the team to prioritize their tasks and remove impediments to handling their tasks.
3. The Team Members directly handle most of the task assignment, daily detail management, progress reporting and quality control of product.

The main shortcoming of traditional method is that the development team build the software in ‘one-shot’ style. If there is an issue arises in middle of any phase then entire module would have to revert, to rectify the issue. In compare to this agile methods are capable to tackle any issue at any phase of developing life cycle. Even development can begin before all requirements are known which shows that agile methods have edge on traditional methods. Agile practices allow the customer to acquire sufficient and significant details on current releases of the system during every interaction and iteration which allows the customer to give its feedback to refine the requirement provided earlier. Fig: 1 shows the comparison between agile methods and traditional methods.

Key Area	Agile Methods	Traditional Methods
Requirement	Largely emergent, Rapid change.	Knowable early, Large stable.
Documentation	Light Documentation.	Heavy Documentation.
Customers	Dedicated, Knowledgeable, Representative, Empowered Customer.	Access to Knowledgeable, Collaborative, Representative and Empowered Customer.
Customer Involvement	High Involvement	Low Involvement
Developers	Agile Knowledgeable, Collocated and Collaborative.	Plan-oriented, Adequate skills, Access to external Knowledge
Development direction	Changeable	Fixed
Refactoring	Inexpensive	Expensive
Architecture	Designed for current requirement	Designed for current and foreseeable requirement
Size	Smaller team and products	Large teams and products
Testing	At every Iteration	After Coding phase completed
Risk	Low risk and identifiable at early stage	After Testing phase
Rework	Low	High
Primary objective	Rapid value	High assurance

Fig: 1. Comparison of Agile and Traditional methods

There is high possibility for stakeholders to maximize their business return on investment by practicing agile methods in system development. Due to the fact of aggressively Change Embracing properties and low rework cost; stakeholders are always having the opportunity to revise the business factors at the beginning of iteration to include additional features into the system according to business ROI [7].

Although agile methodologies have great success over traditional methodologies in many aspects, but still there exist several difficulties in putting it into practice. One among these is that agile methods significantly reduce the amount of documentation and claim that the code itself should act as a document [8].

But there is no guarantee that the code can serve as documentation if the system was originally developed using different methods or developed by different vendors. And the other parameter is the social skills of the developer. Without the business knowledge, most he/she will not be able to deliver the product which is valuable to customers, and customer can lose their trust in development team. But these issues can be resolved by introducing the concept of automated code inspection and refactoring technique with the help of software tools and introducing the short time training program in their course to improve interpersonal and socialization skills [9].



Amandeep Bakshi is Post Graduate student in Computer Science & Engineering Department, Thapar University, Patiala, India and pursuing Master of Engineering in Software Engineering. He holds Bachelor of Technology (B.Tech) degree in Information Technology from Punjab Technical University, I.E.T. Bhaddal, Ropar, Punjab, India (2010).

V. CONCLUSION

Software Development Life Cycle is methodology that represents the entire process to develop successful project by utilizing the resources in efficient manner. As discussed earlier SDLC can be divided into two main categories out of which Agile methodology has an edge upon traditional methodologies because it promotes adaptive planning, evolutionary development with iterative approach, and encourages rapid and flexible response to change. However agile also has some disadvantages but can be rectified by taking some counter measures (like automated code inspection and refactoring) into account to support the agile development. If the organization is working on simple projects and their projects are well defined with clear scope and requirements, then traditional methods will suffice. Alternatively, Agile is the way to go when requirements are uncertain, undefined, and frequently change.

REFERENCES

1. O. Nikiforova, V. Nikulsins, U. Sukovskis, "Integration of MDA Framework into the Model of Traditional Software Development," In: Frontiers in Artificial Intelligence and Applications, Databases and Information Systems, vol. 187, pp. 229–239, IOS Press, Amsterdam, 2009.
2. C. Larman, "Agile and Iterative Development: A Manager's Guide," Addison-Wesley, 2004.
3. Project Management Institute, "A Guide to Project Management body of Knowledge," 4th Edition, 2008.
4. V. Szalvay, "An Introduction to Agile Software Development." Danube Technologies Inc, Nov 2004.
5. K. Beck, M. Beedle, A. Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Jeffries, J. Kern, B. Marick, R. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, "Manifesto of Agile Development," 2001. [Online]. Available: <http://agilemanifesto.org>.
6. T. Dyba, T. Dingsoyr, "Empirical studies of agile software development: A systematic review," Elsevier, January, 2008.
7. David F. Rico, "What is the ROI of agile vs. traditional Methods", 2008.
8. Leo R. Vijayarathy, D. Turk, "Agile Software Development: A survey of early adopters," Journal of Information Technology Management, Vol.19, No.2, 2008.
9. Yu. Leau, W. Loo, W. Tham and S. Tan, "Software Development Life Cycle AGILE vs Traditional Approaches," International Conference on Information and Network Technology, vol.37, pp.162-167, 2012.

AUTHOR PROFILE



Rupinder Singh is Post Graduate student in Computer Science & Engineering Department, Thapar University, Patiala, India and pursuing Master of Engineering in Software Engineering. He holds Bachelor of Technology (B.Tech) degree in Computer Science & Engineering from Thapar University Patiala, Punjab, India (2011).