ISSN 2348-1196 (print)
International Journal of Computer Science and Information Technology Research ISSN 2348-120X (online)
Vol. 3, Issue 1, pp: (199-203), Month: January - March 2015, Available at: www.researchpublish.com

Comparative Analysis of Different Agile Methodologies

Shelly

M. Phil (CS), Department of Computer Science, Punjabi University, Patiala-147002, Punjab, India

Abstract: **Today‘s business, political and economic environment is very much dynamic, and customers are adapting their software requirements to adjust with these new environment. In particular software needs to cater to the change in business environment due to the current economic down trend situation. Agile methodology satisfies the customer through early and continuous delivery of software. This research paper helps in understanding various agile methods; comparison among agile methods: XP, Scrum, FDD, DSDM, ASD and to find out relationship among them to help future developers to get new ideas, methods to develop software and also help to choose the right methodology. This study in general aims to contribute to the agile body of knowledge.**

***Keywords*: Agile, XP, Scrum, ASD, FDD and DSDM.**

I. INTRODUCTION

Software development methodologies are perpetually important in the field of information system. Agile is a software development methodology that develops a product using both iterative and incremental fashion. Agile is a topic of growing importance and nowadays lot of customers wants their projects to be executed using agile methodologies due to its “Deliver value early” principle [1]. Agile overcomes the problem of waterfall methodology.

This article provide an overview of various agile methodologies with key publication of these methodology, identify the key practices of agile methodologies then comparatively analyze the five methodologies on the basis of identified practices This paper provides an analysis of unique practices to agile methodologies, the comparison between the methodologies, the practices common to agile methodologies, and provide an understanding of appropriate methodology combinations.

II. VARIOUS AGILE METHODOLOGIES

Agile is an iterative and incremental approach of software development using shorter development cycles. Agile methodologies are adaptive rather than prescriptive. That is, the methodologies encourage evolution – modifying and or incorporating new practices and discarding practices that no longer work. This research work includes five agile methods: XP, Scrum, FDD, DSDM and ASD.

Extreme Programming (XP):

Kent Beck initially introduced the Extreme Programming approach. Fundamental principles of XP are communication, simplicity, feedback, courage and respect [2].XP initiates with collecting user requirements. On the basis of these requirements the whole development process is divided into small number of cycles. The next phase is iteration planning i.e. deciding the number of cycles, requirements are prioritize and the amount of effort required to implement each cycle is estimated. To develop each iteration pair programming is used. New User Requirements may come during the development phase and the iteration plan should be adjusted according to that. In the next step the latest developed version is tested for bugs, if detected; the bugs will be eliminated in the next iteration. After every acceptance testing project tracing should be done in which feedback is taken from the project that how much job has already been done **[3].**

Scrum:

SCRUM finds its real origins in 1993 at the Easel Corporation where it was successfully applied to a software project. In each sprint, the team creates finished portions of product. Prioritized requirements that go into a sprint come from the product backlog. Sprint planning meeting is conducted taking the release backlog and creating a sprint backlog. Development is time bound such that the sprint must end on time whether all requirements finishes or not. If some of them are not completed for any reason they are turned down and returned to the product backlog. Daily scrum meetings are held after a sprint is finished [4]. At the end of a Sprint, a *Sprint Review Meeting* conducted by the Scrum Master.

Feature Driven Development (FDD):

FDD developed by Jeff De Luca and Peter Coad. According to Palmer and Felsing FDD does not cover whole development life cycle but rather focuses on design and building phases. In *Develop an Overall Model* phase a team consists of development team members and experts are formed to build an overall model of the domain, in order to establish the scope of the system and provide a common understanding the domain. In *Build a Features List* phase The team then identifies a comprehensive list of features and features are grouped together into feature sets, and then major feature sets. In *Plan by Feature* phase Collected feature list is then prioritized and a development plan is establishes that includes the order in which feature sets will be realized. In *Design by Feature and Build by Feature* phase the team launches into a series of Design by Feature and Build by Feature iterations where they break into Feature Teams and design, build, and test features in two-week time boxes. Repeat this step until no more features exist [6].

Dynamic Systems Development Method (DSDM):

The DSDM is a proven framework for agile project management and quick product delivery with a guiding of how to control the process at the same time. Moscow rule is a technique used for prioritizing requirements which are assigned on the basis of Must have, Should have, Could have, Want to have but will not have this time round [6]. When a project is developed with the dynamic systems development method, the feasibility study and the business study must be done sequentially. These two phases is the ones that decide the ground for the rest of the project. The last three phases are iterative and incremental. It is during these phases that the actual development is done in the project.

In Feasibility Study problem is defined and assessments of costs and technical feasibility of delivering a system to solve the business problem is done. Feasibility report and development outline plan is output of this phase. A fast prototype, to make the decision whether to proceed to the next phase or not, can be made [7]. In business study phase business and technology analysis is done. Affected users and presented business processes are described called as business area definition. Outcome of this phase is system architecture definition and development plan that includes prototyping and testing strategy and a configuration management plan. Functional model iteration is the first incremental and iterative phase of this development process. Analysis, coding and testing all are done in this phase. Outcomes are prioritizing functions, Non-functional requirements and Risk analysis of further development, functional prototyping documents [8]. In Design and Build Iteration, a high standard computer system is engineered to be securely placed in the hands of the users [7]. Objective of this phase is to refine functional prototypes to meet non functional requirements. Lastly, in implementation phase the product is delivered and installed in its real environment [8].

Adaptive Software Development (ASD):

Highsmith developed this methodology and states requirements may be fuzzy in the beginning in e-business project. *Speculation* consists of setting project mission and objectives, identifying requirements, Project Time box which based on the feature set requirements, scope, estimates and availability of resources. Length of iteration is decided which depends upon the overall project size and degree of uncertainty. Then time box is assign to the each iteration. Team member write an objective statement for the each iteration. In last developers and users assign features to the each iteration. *Collaboration* requires teamwork which is full of trust and respect. Team must collaborate on rapid decision making, requirements, and problems. *Learning* will help them to improve their level of real understanding. In this phase, focus groups provide feedback, formal technical reviews, and postmortems. [9].

III. CHARACTERISTICS OF VARIOUS AGILE METHODOLOGIES

Characteristics of various agile methodologies are tabulated in following table.

Table 1: Characteristics of various agile methodologies

Characteristics	XP	Scum	DSDM	FDD	ASD
Approach	Iterative increments	Iterative increments	Iterative	Iterative	Incremental
Time Period for Iteration	1 to 6 weeks	2 to 4 weeks	80% solution in 20 % total time	2 days to 2 week	4 to 8 weeks
Team Size	small teams (<20 members)	All Sizes	all sizes independent teams	many members and more than one team	smaller team (5-9 members team)
Suitable Project Size	For smaller Project	all types of projects	all types of projects	more complex projects	smaller projects
User involvement	user highly involved	involvement through the product owner	involvement through frequent releases	involvement through reports	involvement through frequent releases
Documentation	only basic documentation	only basic documentation	Documentation exist	Documentation is important	only basic documentation
Major Practices	User Stories, Test driven development, Refactoring, Pair Programming	Sprint, Product and sprint Backlog, Scrum meetings	Prototyping, feasibility and business study	UML Diagrams	Learning Cycle

IV. EVALUATION OF VARIOUS AGILE METHODOLOGIES

Each of these methods under study were analyzed in detail and 51 different practices were identified and based on these 51 practices these five agile methods were evaluated (Table 2). Hence it is possible to find out whether the practice is unique to specific agile method or not. Below table lists the engineering practices of various agile methods under study. "Y" indicates that particular engineering practices are being used in that agile method predominantly as per literature study. For example: "40 hour week" is engineering practice of both XP and DSDM and "High level Class diagram" is practice of FDD only.

Table2: Engineering Practices of Agile Methods

S.No.	Engineering Practices	XP	Scrum	FDD	DSDM	ASD
1	1-4 week iterations	Y				
2	All types of Testing		Y		Y	
3	Customer driven development		Y			
4	High level Class diagram			Y		
5	Time boxing				Y	Y
6	Iteration of phases	Y			Y	Y
7	Incremental development	Y			Y	
8	40 hour week	Y			Y	
9	User stories usage	Y				
10	collaboration and learning					Y
11	Daily builds and tests		Y			
12	Reporting of Results			Y		
13	Develop only what has direct business value				Y	
14	JAD sessions				Y	Y
15	Onsite Customer	Y	Y			
16	Individual code ownership			Y		
17	Feasibility study				Y	
18	Business study				Y	
19	Regular meetings (preferably daily)				Y	Y
20	Small releases	Y				
21	Component development					Y
22	Software inspections					Y
23	Requirement envisioning			Y		
24	Object Modeling			Y		
25	Frequent Inspections			Y		
26	Regular Builds			Y		
27	Feature based Teams			Y		
28	Evolutionary prototyping				Y	Y
29	Simple design	Y				
30	Customer focus groups					Y
31	Resource requirements analysis					Y
32	Coding standards	Y				
33	Collective ownership	Y				
34	Metaphor	Y				
35	Function point counts					Y
36	Dedicated meeting space					Y
37	Refactoring	Y				
38	Continuous integration	Y				
39	Collocated team			Y		Y
40	Test first development	Y				
41	Post mortems (reflective workshop)					Y
42	Continuous Risk assessment					Y
43	Product Backlog usage		Y			
44	Sprint Execution		Y			
45	Pair programming	Y				
46	Planning game	Y				
47	Sprint Backlog usage		Y			
48	Sprint planning meeting		Y			
49	Daily Scrum meeting		Y			
50	Sprint Review meeting		Y			
51	Release Backlog usage		Y			

V. RESEARCH METHODOLOGY

The Secondary data gathered from literature survey was analyzed to find various Engineering practices. The questionnaire was then prepared on the basis of engineering practices identified. Personal Interviewing Survey method was used for collection of data. In survey respondents were asked to Rank the individual engineering practices under study. The data thus collected have been tabulated first. Then, the data have been analyzed with the help of statistical formulas.

VI. RESULTS OF STUDY

Based upon practices (**Table 2)** data is collected from five IT companies using survey and the statistical analysis is done to find out the relationships among methodologies and their practices. In the first analysis Percentage of engineering projects follow their engineering practices fully, partially and not used any practice at all is calculated. Results shows percentage of DSDM projects using DSDM practices fully is highest, percentage Scrum Project using Scrum Engineering practices partially is highest, percentage of FDD Project not using FDD Engineering practices is highest than others. Percentage of FDD projects using FDD practices fully is lowest, percentage XP Project using XP Engineering practices partially is lowest, percentage of Scrum Project not using Scrum Engineering practices is lowest than others. In second result we calculate which practices of methodology are highly used. As per our survey results Iteration of phases, Incremental development, gets the first two spot in terms of usage in XP projects. Sprint Execution, Sprint backlog usage, Release backlog usage gets the first three spot in terms of usage in scrum projects. Feature based teams, Requirement envisioning, High Level class diagram gets the first three spot in terms of usage in FDD projects. Regular meetings, develop only what has direct business value gets the top two spot in terms of usage in DSDM projects. Collaboration and learning, Software inspections gets the top two spot in terms of usage in ASD projects. Thirdly, five agile methodologies Scrum, XP, ASD, FDD and DSDM are compatible to each other. All other methods prefer XP in combination. XP provides many suitable practices.

VII. CONCLUSION

The field of agile project execution is growing, and the opportunity and need for research is great. It is hoped that this study will provide new data and possible insights to the ongoing efforts at understanding the role of engineering practices and its impact on recommended best practices, project manager behaviors, and organizational policies. Results could be of interest to organizations for identifying leadership traits that should be encouraged or recognized, instructive to project leaders as they attempt to produce desired results in challenging and dynamic environments, and finally, to other researchers that are attempting to create more effective models and approaches in the field of agile. It is concluded that only identifying & counting the number of practices of particular method may not clarify about how the method used on the project. It also requires gathering the data about the extent of practices used as well. In future studies, the same research procedure should be conducted in large geographical area and should include other agile methodologies that are not included in this study due to time constraints.

REFERENCES

[1] DeMarco, T., and Lister, T., (1987) - “People ware: Productive Projects and Teams”, Dorset House Publishing Co., NY.

[2] Mike Holcombe (2008), “Running an Agile Software Development Project”, Wiley

[3] Sheetal Sharma (2012) “Agile Processes and Methodologies: A Conceptual Study”, Vol.4, 894-8. International Journal on Computer Science and Engineering (IJCSE).

[4] Chhavi Malhotra, Anuradha Chug, Agile Testing with Scrum-A Survey , International Journal of Advanced Research in Computer Science and Software Engineering 3(4), March - 2013, pp. 452-459

[5] E. Mnkandla and B. Dwolatzky, “A SURVEY OF AGILE METHODOLOGIES”, The Transactions Of The SA Institute of Electrical Engineers, vol.95, no.4, pp 236-247, 2004.

[6] Diana E. Strode (2005), "The Agile Methods: An Analytical Comparison of Five Agile Methods and an Investigation of Their Target Environment", Unpublished Master of Information Sciences (Information Systems), Massey University, Palmerston North, New Zealand.

[7] Benjamin J. J. Voigt (2004), "Dynamic System development Method", dept. of Information and Technology, University of Zurich, Zürich, Switzerland.

[8] Abrahamsson, P., Salo, O., Ronkainen, J. and Warsta, J. (2002), "Agile Software Development Methods: Review and Analysis", VVT Publications, no. 478, pp. 7–94.

[9] J Highsmith (2002), "Agile Software development Ecosystems", Addison-Wesley.