International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 3 | ISSN : 2456-3307

A Survey on Agile Software Development Methodologies

Santosh Kumar Yadav, Naveen Kumar Jangid, S Rajeshwari

Department of Information Science and Engineering, New Horizon College of Engineering, Bangalore, India

ABSTRACT

Agile software development is rapidly gaining a lot of interest in the field of software engineering. Agile software development, despite its novelty, has become an important domain of research within the software engineering discipline. Agile software development methods have attracted many software engineers and researchers worldwide. But, unfortunately there is scarcity in scientific research, there has been little detailed reporting on the usage, penetration and success of agile methodologies in traditional, professional software development organizations. This paper aims to analyse, organize and make sense out of the dispersed field of agile software development methods.

Keywords : Agile Methodologies, Software Development Approaches, Agile Modelling, Extreme Programming

I. INTRODUCTION

In the 1980s and early 1990s, it was believed that the best way to achieve better software was through careful project planning, the use of analysis and design methods supported by CASE tools, formalized quality assurance and controlled and rigorous software development processes. This viewpoint came from the software engineering community that was accountable for developing large, long-lived software systems.

Discontent with the heavyweight approaches led a number of software developers in the 1990s to propose new ‘agile methods’. These allowed the development team to concentrate on the software itself rather than on its design and documentation. Agile methods generally rely on an incremental approach to software specification, development and delivery. They are appropriate for application development where the system requirements usually change rapidly during the development process. They are envisioned to deliver working software quickly to customers, who can then recommend new and changed requirements to be included in later iterations of the system.

Probably the best-known agile method is extreme programming (Beck, 1999;Beck, 2000). Other agile approaches include Scrum (Cohn, 2009,Schwaber, 2004,Schwaber and Beedle, 2001), Adaptive Software Development (Highsmith, 2000), Crystal (Cockburn, 2001,Cockburn, 2004), Feature Driven Development (Palmer and Felsing, 2002) and DSDM (Stapleton, 1997, Stapleton, 2003).

PRINCIPLES

Principles	Description
Customer involvement	Customers should be meticulously involved throughout the development process. Their role is to provide and prioritize new system requirements and to assess the iterations of the system.
Embrace change	Assume the system requirements to change and so design the system to accommodate these changes.
Incremental delivery	The software is developed in increments with the customer specifying the requirements to be included in each increment.
People not process	The skills of the development team should be known and exploited.

CSEIT172347 | Received : 12 May 2017 | Accepted : 24 May 2017 | May-June-2017 [(2)3: 557-559]

	Team members should be left to develop their own ways of working without prescriptive processes.
Maintain simplicity	Concentrate on simplicity in both the software being developed and in the development process. Wherever possible, actively work to remove complexity from the system.

II. AGILE METHODOLOGY

Agile methodology supports a broad range of the software development life cycle. Some of them focus on the practices (e.g., XP), while some focus on managing the flow of work (e.g., Scrum). Others support activities for requirements specification and development (e.g., FDD), while some seek to cover the full development life cycle (e.g., DSDM).

Types of agile methodology are as follows:

1. Agile Scrum Methodology: Scrum is one of the leading agile software development processes. It's a proven, scalable process for managing software projects. Rather than doing all of one thing at a time Scrum teams do a little of everything all the time. In scrum product is designed, coded, and tested during the sprint. Scrum Team presents what it accomplished during the sprint.
2. Lean and Kanban Software Development: LEAN thinking is important because it can reduce error rates to one per million units. It has been shown to have the potential to at least double the productivity of both manufacturing and service organizations. Kanban offers a visual process-management system and it also supports decision-making about what, when and how much to produce.
3. Extreme Programming (XP): Perhaps the best-known and most widely used agile method. Extreme Programming (XP) use an 'extreme' approach to iterative development of product. New versions of software product may be built several times per day and Increments are delivered to customers every 2 weeks. All the tests must be run for every build and the build is only accepted if tests run successfully.

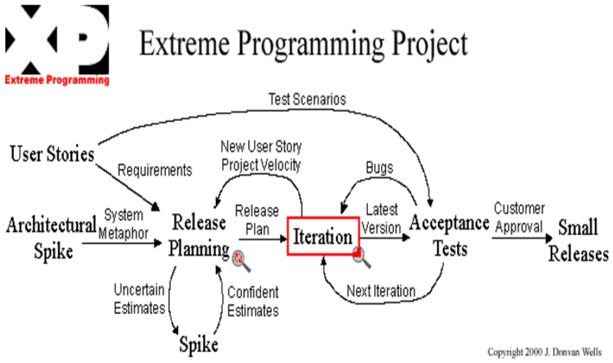


Figure 1: The flow in a project using XP [9]

4. Crystal: Crystal is a human powered method used for small projects or small teams. It requires frequent delivery of usable code to users, reflective improvement and Osmotic communication. Crystal includes Personal safety, Focus, Easy access to expert users, automated tests, configuration management, and frequent integration.
5. Dynamic Systems Development Method (DSDM): Dynamic Systems Development Method (DSDM) is primarily used as a software development method as agile project delivery framework. Earlier DSDM originally used to provide some discipline to the rapid application development (RAD) method. In later versions DSDM was revised and became more of a generic approach to project management and solution delivery rather than being focused specifically on software development and code creation and could be used for non-IT projects .

A. When to Use Agile Methodologies

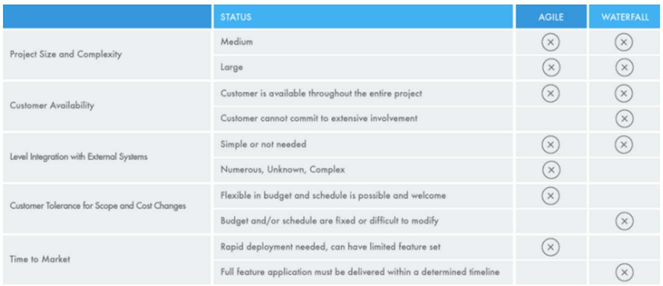
- ✓ When new changes are needed to be implemented.
- ✓ To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- ✓ Unlike the waterfall model in agile model very limited planning is required to get started with the

project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world.

- ✓ Both the system developers and stakeholders alike, find that they also get more freedom of time and options, than if the software was developed in a more rigid sequential way.

III. BENEFITS

- ✓ Satisfaction of customer by rapid, continuous delivery of useful software.
- ✓ People and interactions are focused rather than process and tools. Customers, developers and testers regularly interact with each other.
- ✓ Working software is delivered in increments frequently (weeks rather than months).
- ✓ Face-to-face conversation is the best form of communication.
- ✓ Close, daily cooperation between business people and developers.
- ✓ Continuous attention to good design and technical excellence.
- ✓ Regular adaptation to changing requirements.
- ✓ Even late changes in requirements are welcomed.



	STATUS	AGILE	WATERFALL
Project Size and Complexity	Medium	⊗	⊗
	Large	⊗	⊗
Customer Availability	Customer is available throughout the entire project	⊗	⊗
	Customer cannot commit to extensive involvement		⊗
Level Integration with External Systems	Simple or not needed	⊗	⊗
	Numerous, Unknown, Complex	⊗	
Customer Tolerance for Scope and Cost Changes	Flexible in budget and schedule is possible and welcome	⊗	
	Budget and/or schedule are fixed or difficult to modify		⊗
Time to Market	Rapid deployment needed, can have limited feature set	⊗	
	Full feature application must be delivered within a determined timeline		⊗

IV.LIMITATIONS OF AGILE METHODOLOGIES

There are some limitations to apply agile methodologies. Firstly, agile methodologies are not suitable for greenfield engineering and maintenance, since there will not be much documentation for the systems. Secondly, the success of the project will depend on the cooperation and communication of the user. So, much of user involvement is required. There is a set of assumptions that are assumed to be true, to get the advantages of applying agile methodologies in the development. To mention some are: evolving and changing requirements of the project; cooperation and face to face relation between the customers and the development team; developers having good individual skills and experiences; in addition to many more. It is better to look for other methodologies to apply for the development process, if these assumptions do not apply to a software development project, in order to get better results.

V. CONCLUSION

Software development methodologies have evolved since the 1970s. Agile methodologies were developed after the need for a light way approach to do software development due to changing requirements environment. Agile software development methods have evoked a substantial amount of debates and literature. Still, academic research on the subject is still less, as most existing publications are written by consultants or practitioners. The aim of this paper is to attempt to make sense out of the vast number of emerged agile software development methods. Based on the outcome of the analysis, practitioners are in a better position to understand the various properties of each method and make their judgment in a more informed way. Comparative analysis approach was chosen for the purpose.

VI. REFERENCES

[1]. Extreme Programming. What is Extreme Programming? [Online] Retrieved 18th April 2017.Available at: www.extremeprogramming.org

[2]. Agile Methodology. What is Agile Methodology? [Online] Retrieved 18th April 2017.Available at: http://agilemethodology.org/

[3]. An Approach using Agile Method for Software Development, IEEE 2016 978-1-5090-2084-3/16

[4]. Software Engineering (8th edition) – Ian Sommerville; PEARSON Education