International Journal of Information Technology and Management

Vol. VII, Issue No. X, November-2014, ISSN 2249-4510

REVIEW ARTICLE

STUDY OF THE AGILE SOFTWARE DEVELOPMENT METHODS FOCUS ON APPLICABILITY AND IMPLICATIONS IN INDIAN INDUSTRY

International Journal of Information Technology and Management
Vol. VII, Issue No. X, November-2014, ISSN 2249-4510

Study of the Agile Software Development Methods focus on Applicability and Implications in Indian Industry

Paramjeet Singh

ADIO, NIC, Collectorate, Sri Ganganagar

---------------------------♦-----------------------------

INTRODUCTION

Software development is an organized thrives to deliver products in faster, better and cheaper ways. Agile software development (ASD) is a relative new term within software engineering. Agile processes, or development methods, represent an apparently new approach for planning and managing software development projects. ASD differs from traditional approaches as it puts less emphasis on up-front plans and strict plan-based control and more on mechanisms for change management during the project.

WATERFALL vs. AGILE METHODOLOGY

Waterfall

A classically linear and sequential approach to software design and systems development, each waterfall stage is assigned to a separate team to ensure greater project and deadline control, important for on-time project delivery. A linear approach means a stage by stage approach for product building, e.g.

1. The project team first **analyses**, then determining and prioritizing business requirements / needs.

2. Next, in the **design phase** business requirements are translated into IT solutions, and a decision taken about which underlying technology i.e. COBOL, Java or Visual Basic, etc. etc. is to be used.

3. Once processes are defined and online layouts built, code **implementation** takes place.

4. The next stage of data conversion evolves into a fully **tested** solution for implementation and testing for evaluation by the end-user.

5. The last and final stage involves **evaluation** and **maintenance,** with the latter ensuring everything runs smoothly.

Agile software development methods can be seen as a new idea to plan-based or traditional methods, which emphasize a rationalized, engineering-based approach in which it is particular those problems are fully specifiable and that best and anticipated solutions exist for every problem. The “traditionalists” are said to advocate extensive planning, codified processes, and rigorous reuse to make development an efficient and predictable activity. By contrast, agile processes address the challenge of an unpredictable world by relying on people and their creativity rather than on processes” .In 2001, the agile manifesto” was written by the practitioners who proposed many of the agile development methods. The manifesto states that agile development should focus on four core values:

1. Individuals and interactions over processes and tools.

2. Working software over comprehensive documentation.

3. Customer collaboration over contract negotiation.

4. Responding to change over following a plan.

To synopsis the difference between the two, one can say the classic waterfall method stands for predictability, while agile methodology spells adaptability. Agile methods are good at reducing overheads, such as, rationale, justification, documentation and meetings, keeping them as low as is possible. And, that is why agile methods benefit small teams with constantly changing requirements, rather more than larger projects.

Agile methodology means cutting down the big picture into puzzle size bits, fitting them together when the time is right e.g. design, coding and testing bits. So, while there are reasons to support both the waterfall and agile methods, however, a closer look clarifies why many software and web design firms make the more appropriate choice of employing Agile methodology. The following reasons are there for choosing Agile methodology over the Waterfall method.

1. Once a stage is completed in the **Waterfall method,** there is no going back, since most software designed and implemented under the waterfall method is hard to change according to time and user needs. The problem can only be fixed by going back and designing an entirely new system, a very costly and inefficient method. Whereas, **Agile methods** adapt to change, as at the end of each stage, the logical program, designed to cope and adapt to new ideas from the outset, allows changes to be made easily. With Agile, changes can be made if necessary without getting the entire program rewritten. This approach not only reduces overheads, it also helps in the upgrading of programs.

2. Another **Agile method** advantage is one has a launch able product at the end of each tested stage. This ensures bugs are caught and eliminated in the development cycle, and the product is double tested again after the first bug elimination. This is not possible for the **Waterfall method,** since the product is tested only at the very end, which means any bugs found results in the entire program having to be re-written.

3. **Agile's** modular nature means employing better suited object-oriented designs and programs, which means one always has a working model for timely release even when it does not always entirely match customer specifications. Whereas, there is only one main release in the waterfall method and any problems or delays mean highly dissatisfied customers.

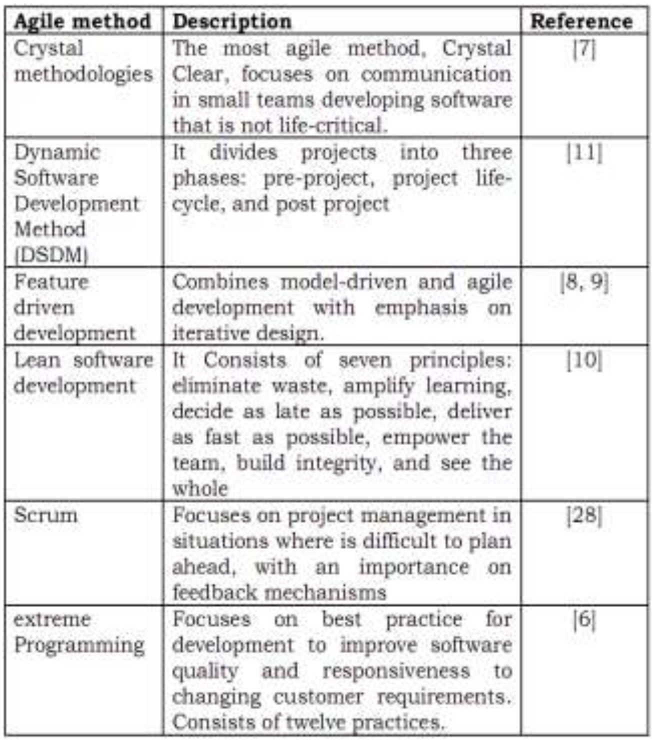
4. Agile methods allow for specification changes as per end-user's requirements, spelling customer satisfaction. As already mentioned, this is not possible when the waterfall method is employed, since any changes to be made means the project has to be started all over again.

5. However, both methods do allow for a sort of departmentalization e.g. in waterfall departmentalization is done at each stage. As for Agile, each coding module can be delegated to separate groups. This allows for several parts of the project to be done at the same time, though departmentalization is more effectively used in agile methodologies.

In conclusion, though on the plus side, waterfall's defined stages allow for thorough planning, especially for logical design, implementation and deployment, Agile methodology is a sound choice for software development and web design projects. More and more firms are becoming Agile!

There are a number of agile software development methods. Methods for agile software development represent a set of practices for software development that have been created by experienced people [1]. The most common methods are extreme Programming (XP) [4], Dynamic Software Development Method (DSDM) [5], Scrum [2], and Crystal [3]. The highest criterion of these methods is development of the software and customer satisfaction through continuous delivery of software. This is achieved by having short iterations in the development process. The iterations focus on timely delivery of working code that provides substantial value to the customer. More importantly, is a very limited work that has been produced to assess the applicability of agile methods.

Table 1: Description of Main Agile Development Methods with References"



Agile method	Description	Reference
Crystal methodologies	The most agile method, Crystal Clear, focuses on communication in small teams developing software that is not life-critical.	[7]
Dynamic Software Development Method (DSDM)	It divides projects into three phases: pre-project, project life-cycle, and post project	[11]
Feature driven development	Combines model-driven and agile development with emphasis on iterative design.	[8, 9]
Lean software development	It Consists of seven principles: eliminate waste, amplify learning, decide as late as possible, deliver as fast as possible, empower the team, build integrity, and see the whole	[10]
Scrum	Focuses on project management in situations where is difficult to plan ahead, with an importance on feedback mechanisms	[28]
extreme Programming	Focuses on best practice for development to improve software quality and responsiveness to changing customer requirements. Consists of twelve practices.	[6]

RESEARCH QUESTIONS

The paper is being carried with the following objectives:

- To examine, gain insight into the agile methods and practices

- To find out the issues in where, when and how agile methods are used
- Strengths and weaknesses of agile methods
- Understanding the applicability of agile methods
- Understanding transition from traditional methods to agile methods and its effects on the organization

To get good understanding of the above mentioned, we formulated research questions. These are discussed as follows.

A. What are the current agile methods and practices?

This research question attempts to discuss the agile methods. It is understood that some agile methods are not popular, so it is of immediate concern to know how the agile methods are differ from each other. Also, this contributes to one of the fundamental tasks of the paper. The outcome would be to identify the companies that have incorporated the agile philosophy and methods. This will give an understanding of the most commonly used agile methods.

B. What are the benefits of the agile methods and their level of applicability in industry?

We primarily focus on understanding what has been published on all the agile methods, benefits and limitations. It would give a broader perspective on the following:

- ▪ Is there any relation between type of software developed and the agile philosophy adopted?
- ▪ Is there any relation between size of the project/company and agile methods?
- ▪ Benefits with respect to agile methods chose and type of project.

C. What are the impacts of using the agile methods in industry?

The subjects of relevance to this research question would be to collect information on the changes that the introduction of agile methods has brought to the organization. This also helps in knowing the following:

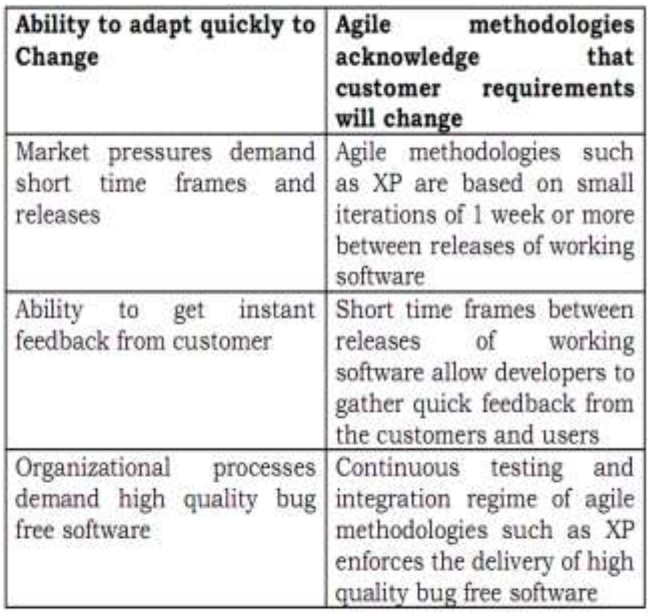
Practicing agile methods is always beneficial to employees in terms of job satisfaction, communication etc.?

Level customer satisfaction achieved and issues, if any, encountered among the customers

AGILE METHODOLOGY KEY ASPECT

Agile methodologies like XP are capable of quickly adapting to the changing requirements of the customers. XP accommodates frequent changes in the software product and is tailored made for such purposes. XP facilitates frequent releases of the working software. Developers generally believe a positive outcome in obtaining regular feedback on the software being built. Another important aspect of XP we found is its role in testing. XP maintains a very high quality throughout the design and testing. As software is released in intervals, defects can almost be perceived as public and are easy to rectify. "

**Table 2: Reasons Motivating Adoption of Agile"
Reasons motivating to adopt agile methodology**

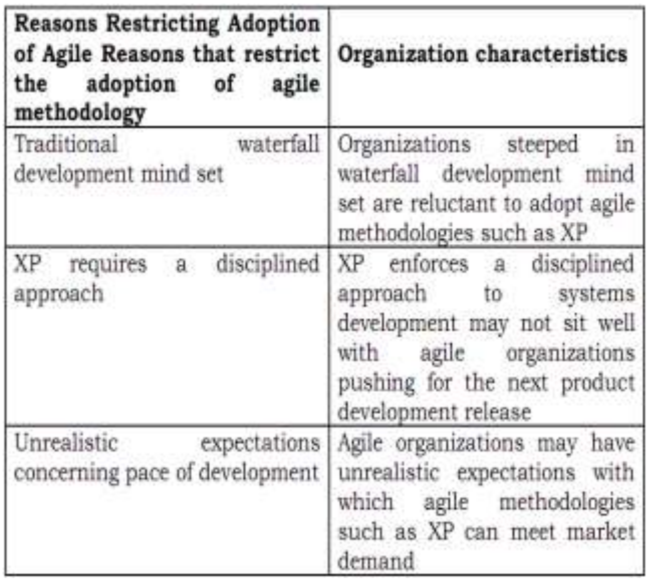


Ability to adapt quickly to Change	**Agile methodologies acknowledge that customer requirements will change**
Market pressures demand short time frames and releases	Agile methodologies such as XP are based on small iterations of 1 week or more between releases of working software
Ability to get instant feedback from customer	Short time frames between releases of working software allow developers to gather quick feedback from the customers and users
Organizational processes demand high quality bug free software	Continuous testing and integration regime of agile methodologies such as XP enforces the delivery of high quality bug free software

The agile methods are designed to serve particular purposes. These are mentioned here.

- DSDM provides a framework for RAD.
- Scrum is like a custom made methodology for project management of iterative development.
- XP aims at software development in frequent change environments. Teams are usually small.
- ASD provides a framework for managing software projects that are under intense time pressure and where requirements are changing rapidly

Table 3: Reasons Restricting Adoption of Agile



Reasons Restricting Adoption of Agile Reasons that restrict the adoption of agile methodology	Organization characteristics
Traditional waterfall development mind set	Organizations steeped in waterfall development mind set are reluctant to adopt agile methodologies such as XP
XP requires a disciplined approach	XP enforces a disciplined approach to systems development may not sit well with agile organizations pushing for the next product development release
Unrealistic expectations concerning pace of development	Agile organizations may have unrealistic expectations with which agile methodologies such as XP can meet market demand

Environmental or organizations factors for agile methods

Situations where agile methods are most effective

1. Based on Scrum only, not for large complex team structures

Based on XP, Scrum, Agile Unified process, Agile Modeling Internet application domains Significant time-to Markey pressure Cost of upgrade to the next release is minimal Not suitable for long-lasting, large, complex systems

A management style of leadership and collaboration Project manager acts as facilitator or coordinator Teams capable of self-organizing Developers are competent, above-average people Small teams High change environments Where communication is formal Where the customers is prepared to accept a critical role in development Organizational structure is flexible, participative and encourages cooperative social action Object-oriented technology

Extreme Programming (XP)

Pros The basic process structure (life cycle) of XP looks like it could adapt also to very small projects also. However, when having only one developer, pair programming and continuous review practices have to be dropped out.

Cons Pair programming is a very important practice in XP. However, it cannot be applied to one-developer-projects. Customer collaboration is not very strong. Testing and code development is done by the same person. All the possible problems may not be found because the developer tests from the same perception the product is built.

Scrum

Pros Scrum fits well into small projects. Some work releases are created and requirements can be prioritized in a well-structured manner.

Cons Customer is offsite and tight customer collaboration is not possible. Also improved team dynamics enabled by Scrum are not available in one-developer project.

Dynamic Systems Development Method

Pros This is heavier than XP and Scrum. It provides a technique-independent process and is flexible in terms of requirement evolution. It is efficient in terms of budget and time.

Cons It is based on user involvement which is not possible in every project

Feature driven development

Pros

Excellent reporting & planning

Risk Reduction

Provides a starting point for modeling using existing models

Disciplined & Clear

Customer Focused

Cons

High reliance on Tech Leads

Doesn't deal with User Interface design & build

Not as powerful on smaller projects (one developer, only one person modeling)

Doesn't cover testing & deployment

Lean Software development

Pros

Consider the system as a whole, though difficult for complex system, helps to guarantee consistency and integrity of the system. Reduces integration time since is developed as singular unit**.** The work team designs its own processes, makes its own commitments, gathers the information needed to reach its goals, and polices itself to meet its milestones.

Cons

With large system or complex system, the only way for developers to visualize its construction is through partitioning the system But it suggest the opposite, which can be difficult to accomplish. Deciding as late as possible can have adverse effect to the schedule. This can hurt parallel development and increase implementation time.

Crystal Methodologies

Pros

Software delivered incrementally, Automated Regression Testing and Direct user involvement

Cons

Need of skilled resources

CONCLUSION

Agile approaches are meant to increase flexibility, agility and to be more adjusted to the environment where software development projects are present and working today. This is a contradiction to large global project organizations with no overview and multiple interdependencies that cannot be effectively monitored. However, nothing speaks against incorporating ideas and practices from agile methods in order to increase agility even in large projects though keeping in mind that the fundamental conditions are different and that that needs to be fully understood. The ideal approach would likely be to break large projects into smaller projects which would become more flexible. Agile approaches are meant to increase fastness and flexibility in the software projects. The developers may feel stressed the following four reasons for adopting agile methods: adaptability to change, short time frames of releases, continuous feedback from customers, high-quality and bug free software.

REFERENCES

[1] P. A Gerfalk, B. Fitzgerald. Flexible and distributed software processes: old petunias in new bowls? Communications of the ACM 49 (10) (2006) 27–34

[2] Schwaber K. and Beedle M. Agile Software Development with Scrum. Prentice Hall, 2001. International Journal of Software Engineering and Its Applications Vol. 5 No. 2, April, 2011 44

[3] M. Aoyama. Web-based agile software development, IEEE Software 15 (6) (1998) 56–65

[4] P. A ˚ gerfalk, B. Fitzgerald, Flexible and distributed software processes: old petunias in new bowls? Communications of the ACM 49 (10) (2006) 27–34.

[5] S. Nerur, R. Mahapatra, G. Mangalaraj, Challenges of migrating toagile methodologies, Communications of the ACM (May) (2005) 72–78.

[6] K. Beck, Extreme Programming Explained: Embrace Change, Addison-Wesley, 2000, ISBN 0-201-61641-6.

[7] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams, Addison-Wesley, 2004, ISBN 0-201-69947-. Kajko-Mattsson, M.; Lewis, G.A.; Siracusa, D.; Nelson, T.; Chapin, N.; Heydt, M.; Nocks, J.; Snee H.,

[8] "Long-term Life Cycle Impact of Agile Methodologies," Software Maintenance, 2006. ICSM 06. 22nd IEEE International Conference on, vol., no., pp.422-425, Sept. 2006.

[9] Miller, G., “Want a better software development process'? Complement it,” IT Professional, vol.5, no.5, pp. 49-51, Sept.-Oct. 2003.

[10] M. Poppendieck, T. Poppendieck, Lean Software Development – An Agile Toolkit for Software Development Managers, Addison-Wesley, Boston, 2003, ISBN 0-321-15078-3.

[11] Raising, L., & Janoff, N.S. (2000). The Scrum software development process for small teams. IEEE software, 17(4), 26-32.