

## Agile Methodologies: Working Mechanism with Pros and Cons

Rajiv Kumar<sup>1</sup>, Dr. Ashu Gupta<sup>2</sup> and Dr. Hardeep Singh<sup>3</sup>

### Abstract

Now days the choice of right software development life cycle (SDLC) has become big challenge for the organizations which are involved in the development of software(s). There are number of models like Waterfall, Prototyping, Spiral etc. are available with different set of advantages and disadvantages. Directly or indirectly these models are based upon the traditional Waterfall model. One different approach which is given the name Agile Modeling is a practice-based methodology based upon V model which is used for modeling and documentation of software. Agile software development technology also called “agile technology or methodology” is not a set of tools but is an philosophy which came in late 1990s. Instead of relying on heavy documentation Agile focused on to be a collection of values, principles, and practices for modeling software that can be applied on a software development project in a more flexible manner than traditional Waterfall modeling method. The comparative study of agile methodologies and traditional methodologies assists different organizations to choose optimum methodology for their projects under consideration. All around the world XP and SCRUM the most popular *agile methodologies* have widespread reputation. The core of agile methodologies is people, customer and each team member in agile development teams which are the key success or failure factors in agile process. In this paper we have discussed the XP and SCRUM along with strength and weakness points in agile methodologies and corresponding affects of strength and weakness factors on the overall results of agile development process.

**Keywords:** SDLC; Agile methodologies; Waterfall; Agile Strength; Agile Weakness; XP; SCRUM.

### 1. Introduction

Cost, Quality and Schedule are the three main concerns associated with the development of the Software projects. Different surveys, researches, models and working mechanisms are suggested all over the world to improve and enhance the overall development process. With the growth of industry and dynamic behaviour of market scenario nothing is stable now days. Traditional methods are not sufficient to deal with such speedy changes in the environment and distributed geographical conditions. To deal with these rapid changes in organization and business methods a new mythology emerges. This methodology is called agile method or agile methodology. Agile methods mainly focus on people, customer’s satisfaction and rapid response to change (Boehm et al., 2003).

---

<sup>1</sup> Assistant Professor, Deptt. of Computer Applications, Apeejay Institute of Management Technical Campus, Jalandhar. Research Scholar of Punjab Technical University, Kapurthala.  
Email: r\_pundrik2003@yahoo.com

<sup>2</sup> Assistant Professor, Deptt. of Computer Applications, Apeejay Institute of Management Technical Campus, Jalandhar. Email: guptashu1@rediffmail.com

<sup>3</sup> Professor, Department of Computer Science & Engineering, Guru Nanak Dev University, Amritsar.  
Email: hardeep\_gndu@rediffmail.com

In February, 2001, the creators and supporter of agile methodologies met first time to talk and prepare the schema of Agile Software Development (ASD). In the meeting seventeen people discussed and talked about the future of software development methodologies. Further they noticed that they share some common characteristics in general and all of the main characteristics are mainly focused on people. The result of the meeting gave a new technology to the world given name “Agile Methodology OR Agile Alliance”.

## 2. Agile Vs Traditional Approach

In the old traditional methodologies such as System Development Life Cycle (SDLC) which is a structure imposed and develops the software product in sequential phases. There are several models for such processes, each describing approaches to a variety of tasks or activities that take place during the process. Traditional software development methodologies has been quite famous and been used by many software development teams and enterprises across the world. SDLC includes several stages from preliminary development analysis to post development software testing and evaluation (Mohammad et al., 2013).

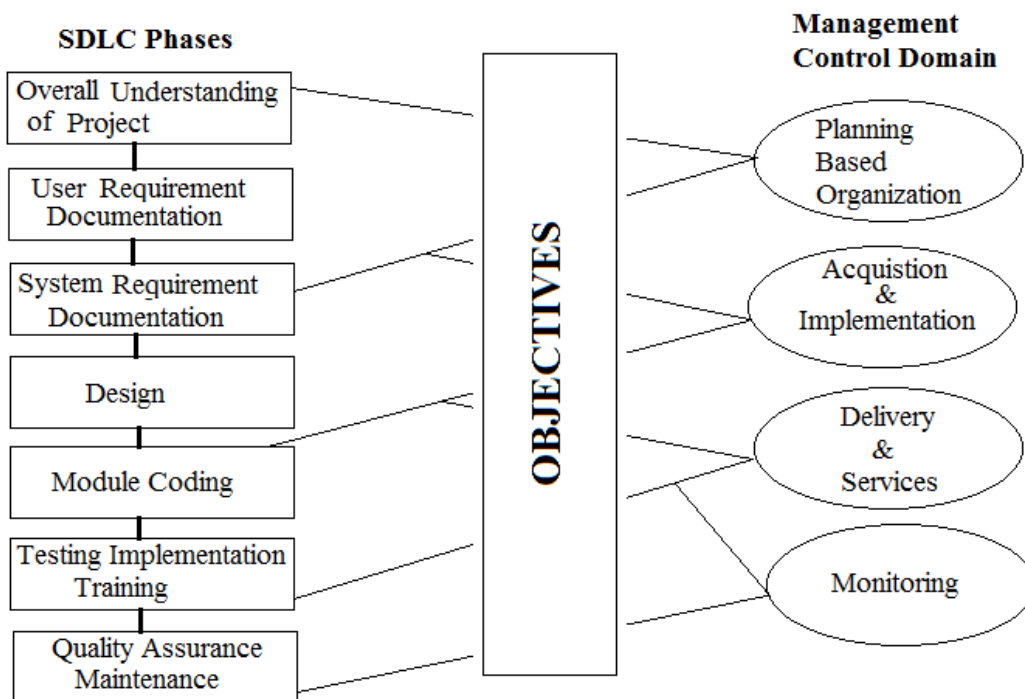


Fig 1. SDLC Activities\*

\*Adapted from: [www.en.wikipedia.org/wiki/File:SDLC\\_Phases\\_Related\\_to\\_Management\\_Controls.jpg](http://www.en.wikipedia.org/wiki/File:SDLC_Phases_Related_to_Management_Controls.jpg)

Traditional approach is having different models like waterfall, spiral, step increment etc. with corresponding working methodologies generally used by the developers for the development of a particular software. These complex working mechanisms are not easy to handle, due to this reason developers were interested to have some flexible method instead of using traditional methods. These efforts for the flexible methodology search lead to the birth of new methodology which came in existence in the start of 2000 and known as agile methodology (Larman, 2003). This methodology is an iterative based methodology.

Agile software development process or agile software methodologies are a group of software development process that based on incremental development or iterative development. In agile development process requirements and solutions are changed and evolved through collaboration and communication between organization and teams (Mohammad et al., 2013). The main concentration is given to the people (client / end users or team member). Agile software processes are capable to deal with rapid changes from user side or due to changing environment effects.

Out of different possible attributes of agile methodology one of the main and core characteristics of agile methodologies is that it addresses the problems of rapid change and it is based on best practices and their previous success and failures stories. In addition agile methodology team must be more effective and efficient in responding to changes. These days, agile software methodologies or lightweight methodologies generate a huge debate between traditional methodologies followers and agile methodologies followers (Mohammad et al., 2013). But later are found to be in strong position due to number of evidences in support of success gained by the agile methodologies in different fields and positive feedback from the persons in the field of agile technology.

By using traditional approach first preference is given to conversion of overall goal to corresponding possible scope of the software with reference to present available resources of the organization but in case of agile technology the emphasis is given to achieving the final goal by breaking a big problem into number of small problems. The size of each sub problem is decided on the basis of the capacity of the team and number of iterations. With the passages of iterations the dynamic scope will become clear to all the concerned either from developer side or client side.

The Traditional planning method uses approximately 60 percent time and budget of the project with repeated updates in the requirements due to which rest of the phases suffer. Agile planning is layer based, here planning is divided into different layers ranging from a high level layer to low level layer at the start of any particular iteration. Throughout the whole project development customer changes in requirements are welcomed and appreciated, so dynamic changes flexibility is ensured in this approach.

The traditional approach is pipe lined in nature having different phases in sequence in such an arrangement that input of next phase is dependent upon the output of the previous phase, client is supposed to interfere only in initial and final phase but agile always prefer to have client as a team member and take suggestion from the client by providing him / her some working model after the completion of each iteration.

No rigid hierarchy with respect to management is followed in agile methodology. The team members are assigned roles and responsibilities on the basis of their skill and experience. The overall objective will remain at higher priority to get optimum solution to each problem in the way. Proper team work is appreciated and every team member is promoted to work for the common goal.

Agile methodology supports decentralized approach regarding management and decision making and motivate all individuals to give their best to achieve central goal well in time by keeping in mind the dynamic scenario of the present market. In every iteration, attempt is made to provide maximum cohesion and least coupling among the different sub modules going to be integrated in future. Traditional risk management is descriptive, identifying the starting point of the potential risks along with mitigation actions. In the agile approach, risk management is based on experimentation. Learning takes place over the iterations. And within the iterations, the impact of failures is limited (ISACA, 2014).

### 3. Agile vs. waterfall: Working differences in methodology

The waterfall (software development) methodologies were hardly methodologies at all, but a basic technique for those organizations which struggled to profit from new computer-related technologies. With the passage of time as the organizations learned more about different possible ways to get profit from developing software, certain techniques for managing and controlling the cost of software development projects came into use. Out of different methodologies the methodology which dominated software development projects for decades is called “waterfall” (Serena Software Inc., 2007).

Agile software development (ASD) is relatively a new term used in software engineering. Agile processes or agile methodologies actually represent a new approach for planning and managing software projects (Rao et al., 2011). The working mechanism of both waterfall and agile methodology are totally different which are described below in tabular form:

Waterfall Model	Agile Technology
1. Believe in depth analysis of the problem.	1. Believe in individuals and interactions among people.
2. Prepare detailed requirement specifications (SRS).	2. Development of working software instead of documentation.
3. Implementation of specifications in sequence.	3. Collaboration among different teams.
4. Testing of different types at different levels.	4. Iterative responses to dynamic changes.
5. Delivery and maintenance of finished product.	5. Flexible and unstable methodologies

### 4. Types of Agile Methodologies

There are several agile methodologies that are available. The major types of this technology are: Extreme Programming (XP), Scrum, Lean and Kanban Software Development, Crystal, Dynamic Systems Development Method (DSDM) and Feature-Driven Development (FDD)

Out of all above specified types first two approaches i.e. XP and Scrum are more popular among the organizations which are shifting their focus from traditional methodology to agile methodology, so here in this paper XP and Scrum methodology is discussed in detail.

### 5. Extreme Programming (XP)

XP stands for extreme programming. It concentrates on the development rather than managerial aspects of software project. XP was designed so that organizations would be free to adopt all or part of the methodology (Serena Software Inc, 2007). Beck and Fowler (2000) described this technology in the market. Out of different possible agile technologies XP is the most popular technology but side by side it is also considered as one of the most controversial agile methodologies. XP is a much disciplined approach which is committed to deliver high-quality software very quickly and continuously. It is based upon the principle of high customer involvement, quick feedback, continuous testing, continuous planning, quick response to changes and close teamwork to deliver working software at very frequent intervals, typically every 1-3 weeks. The original XP recipe is based on four simple components, viz. **simplicity, communication, feedback and courage** (Williams, 2007).

**5.1 Working Mechanism:** In XP, all the members from the Customer / Client side or some time end user are very closely involved with development team during the development period to define and discuss the "User Stories". The development team estimates, plans, and delivers the

highest priority user stories in the form of working, tested software on an iteration by iteration basis. In order to maximize productivity, the practices provide a supportive, lightweight framework to guide a team and ensure high-quality software (Guha, 2011).

The first phase of the XP projects is release planning phase, followed by several iterations, each of which concludes with user acceptance testing. When the product has enough features to satisfy users, user will decide priority of development among different features. Corresponding to these priorities the team will develop some small working software releases. Initially end users or clients describes their needs or requirements which software should fulfill at least without going in details. These written small documents from user side are given the name "User Stories". On the basis of user stories development team will finalize their tools and resources required to develop the software. The persons who will provide these user stories are considered as a part of the development team and are authorized to give their suggestions as well as requirement changes while the process of development.

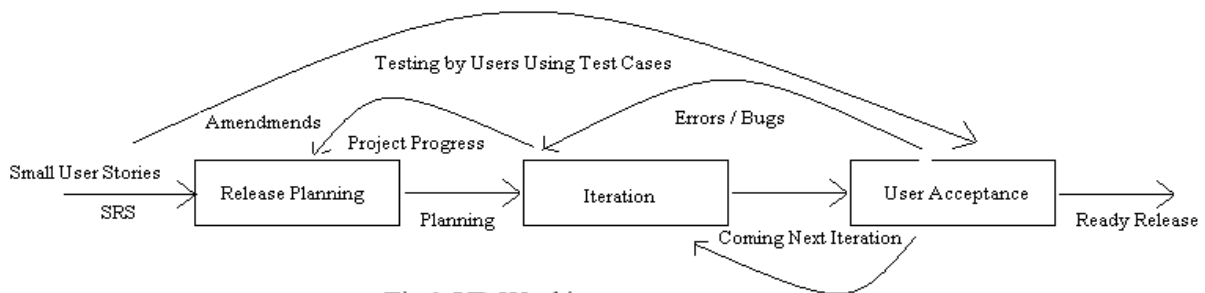


Fig 2. XP Working

Source: [www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf](http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf)

For the better understanding of a release plan, the team uses the strategy of breaking big module into small sub modules according to which the overall development tasks is further divided into different iterations. The release plan is basically a complete charted out work planning of the particular iteration which is used before the start of that particular iteration. When iteration will end this release plan will be compared with the user stories given by the client, if some bugs are found during these comparisons, are noted down and removal of these bugs will become the priority in the coming next iteration. This iterative testing, after each iteration, is serving the purpose of acceptance testing and will result in final release of the developed software. Now there will be a time when all the concerned members will agree upon that now there is no further requirement to any user story, it mean it will be the time to stop the overall project and will move towards the finishing of the project.

**5.2 Important Rules:** Although there is a very long list of XP methodology rules but some very common and useful rules are described below:

**5.2.1 User Story:** Basic requirements written in simple language given by the client to the development team or the services client wants to be given by the system going to be developed are known as User Stories. User story should not be in the form of a long document but it should not contain more than six lines. User stories will provide complete detail of the services and functionalities which are expected from the purposed software but will never provide the solution to these problems. Instead, a sample user story might go like this: Search for customers. The user tells the application to search for customers. The application demands from the user to specify which customers. After the user specifies the search criteria, the application returns a complete list



of customers meeting those criteria. Because user stories are short and somewhat vague, XP will only work if the customer representative is always in touch with the development team to review and approve user story implementations. This is one of the main objections to the XP methodology, but also one of its greatest strengths (Serena Software Inc., 2007).

**5.2.2 Pair programming:** Instead of coding by single programmer for the release, XP always support coding by team of programmers from the same skill field. Team should be consist of minimum two programmer but not more than four programmers. With the involvement of more than one programmer chances of better results with more suggestions and good quality end product will get increase. Further processing speed will get increase and it will be finished in time. Further two persons can support each other with their experience and assistance to other person in the team.

**5.2.3 Continuous Integration often:** Under XP methodology it is recommended that all the members of software development team as well as members from client side must meet to each other at least once in a day at prescheduled time (generally in the evening) to discuss all the suggested changes. Further with such type of integrations among team members overall performance regarding issues are also given considerations. It is not a hard and fast rule but such type of small and regular meetings are highly recommended.

**5.2.4 Constant Project Progress:** Regular meetings among team members and exchange of views after each release is also considered very use full tool to measure progress of the project. This measure is also known as **velocity** of the project. Velocity measure tells us that how much work is getting done on the project. This important metric drives release planning and schedule updates.

## 6. Scrum

Scrum is a lightweight agile project management technology with broad applicability for managing and developing iterative and incremental projects of all types. Ken Schwaber, Mike Beedle, Jeff Sutherland and others have contributed significantly to the evolution of Scrum over the last decade (VersionOne, Inc., 2014). In rugby, 'Scrum' (related to "scrimmage") is the term for a huddled mass of players engaged with each other to get a job done (Serena Software Inc., 2007). Due to its simple development mechanism and managerial attributes Scrum technology's popularity is increasing day by day all around the world.

**6.1 Working Mechanism:** Scrum is based upon backlog to be completed in next iteration. Here in scrum, the "Client or End User" works closely with the development team to identify and prioritize system functionality in form of a "Product Backlog". The Product Backlog consists of composition of functional and nonfunctional features. Functional features include interface priorities, bug fixes, data mismatch etc. Non-functional requirements include quality measures; additional functionalities, standards etc. - whatever needs to be done in order to successfully deliver a working software system. With priorities driven by the Product Owner, cross-functional teams estimate and sign-up to deliver "potentially shippable increments" of software during successive Sprints. Sprint is the name given to small iterations in which modules are divided to break down the complexity of the product. These sprints are decided immediately after the finalization of broad scope and high end design of the product. After deep analysis and number of meetings with concerned persons, once backlog of particular sprint is completed, no other additional services can be added to the final product of the sprint. After the delivery of particular sprint, its backlog will be re analyzed and priorities are reset according to the changes required. Scrum is quite useful to deal with rapid changes in the different phases of the product development.

At the start of each sprint project backlog is propagated among the team members and persons from client side to decide the scope of the next release. Each sprint sets its goal to fix backlog bugs reached to it from the previous sprint. Before the implementation of each sprint, the team members identify the backlog items for the sprint. After the completion of sprint, the team reviews the sprint to collect findings in the form of lessons learned and check progress. Like XP even in scrum, daily small team meetings are recommended. These small meetings are given the name scrum. During these scrums every team member describes the work to be done that day, progress from the day before, and any blocks that must be cleared. These meetings will be formal but in the informal environment, so the scrum is supposed to be conducted with everyone in the same room—standing up for the whole meeting.

When enough of the backlog has been implemented so that the end users believe the release is worth putting into production and nothing left to be covered, management closes development. The team then writes the code, performs integration testing, training, and small documentation as necessary for product release.

**6.2 Important concepts:** Again there is a very long list of Scrum methodology concepts in real practice but some very common and useful concepts are described below:

**6.2.1 Product backlog:** Every sprint is based upon product backlog which is basically a complete list of requirements—including bugs, enhancement required in different interfaces, usability and performance improvements—that are not currently in the product release provided but in final product which are expected.

**6.2.2 Scrum Master:** The project leader is generally given the designation of the Scrum Master and he/she is the person responsible for managing the Scrum project teams and overall activities associated with the development of different releases. Sometimes it refers to a person who has become certified as a Scrum Master by taking Scrum Master training.

**6.2.3 Sprint backlog:** Sprint backlog is the list of backlog items assigned to a sprint, but not yet completed. In ideal conditions it is recommended that no sprint backlog item should take more than two days to complete. The sprint backlog helps the team to predict the level of effort required to complete a sprint and meet the targets which were set before the implementation of any sprint.

**6.2.4 Burn down chart:** Burn down chart is as important in scrum as velocity of project is in case of XP. This chart, updated every day, shows the complete details of sprint progress and used to check the current status of work remaining within the sprint. The burn down chart is used both to track sprint progress and to decide when items must be removed from the sprint backlog and deferred to the next sprint.

## 7. Strengths of Agile methodologies

**a) Working philosophy:** Agile methodology is not a set of tools but is a very simple philosophy which is based on the idea that projects are developed in short iterations. Iterations are useful to breakdown the complex problem into small sub modules in the form of small iterations for different sub modules. At the end of any iteration the user can see a working version of the software before moving ahead to the next iteration which means that the overall project will be more flexible.

**b) Quick response to rapid changes:** Traditional methodologies are very rigid in practice. Once requirements are finalized then any small change is not admired too because it will require complete project rebuilding, but through applying agile philosophy it can be adapted at any time as per requirement of client or developer.

**c) Constant Evaluation:** Next iteration will come in existence after proper evaluation by the user

by applying some acceptance criteria. Constant suggestions from user side are evaluated by the development team to enhance the product quality. Future planning is based upon these constant evaluations by the users.

**d) Regular testing and integration:** Each iteration has to be tested before the next release with constant coordination and integration with previous iteration result so constant testing and integration is done after each iteration which means that faults can be corrected regularly.

**e) Work progress evaluation:** The overall progress report can be generated and updated regularly. Progress of the product can be viewed and measured after each iteration.

**f) Team work:** Agile methodology is useful in improving personnel relationships among team members and motivates the persons to work together in a group for the common goal leaving behind their individual interests. Further it gives preference to people and their creativity, so it will be useful in solving emergent issues and problems immediately without following rigid policy and procedures.

**g) Direct bi-directional communication:** In traditional approaches heavy documentations is there between developer and customer but in agile technology direct communication and regular meetings are the part of the overall development process. From both sides suggestions are exchanged. Developers can communicate with customers and respond to their requirements without following rigid policies in communicating with managers to take an approval for every simple modification. Flexibility does not mean that developers can do modifications and respond to customers' needs without telling their supervisors but they can respond to customers' modifications before telling their managers.

**h) Small team structure:** Different responsibilities are distributed among different teams which are small in size generally consist of two or three persons. This means that there is always a continued channel for communication between team members which increase job satisfaction. Continued communications means identifying problems and mistakes earlier and improvement in results.

**i) Customer preference:** In agile methodologies customers are more important than tools and processes, so customers are playing an important role in the working team. Agile methodologies recommend having a full time customer working with development team. This means that any misunderstand of customers' requirements can be solved and answered immediately. Also this means the time taken for solving any emergent issues will be reduce to the minimum.

## **8. Weakness of Agile methodologies**

**a) More cooks will spoil the curry:** Regular involvement of different persons from different fields will provide regular changing suggestions, which lead to some abnormal circumstances and can increase the complexity for the iteration.

**b) Flexibility taken for granted:** Generally a very common mistake that many people assume that agile mainly based on flexibility and lack of flexibility. In reality this is not true, since agile methodologies has its own formal rules and policies. Instead of suggesting changes after thorough study everyone is free to suggest any change at any level just to show his/her presence only without proper responsibility because everybody says it is always ready to respond any change.

**c) Customer dependent:** Regular customer suggestion can become bottleneck for the progress of the project. Agile followers consider customer interaction as a merit. But in fact it may become a weak point in some circumstances such as the user or customer might not find enough time to spend with developers, or if the key customer is one of the high level managers.

**d) Daily Meetings:** It is very difficult to spend some minutes daily for the long projects especially



in those cases where customers and developers are geographically far distributed. Practically these meetings will be very costly not applicable under all situations.

**e) Lack of coordination:** Large projects need not to be completed under same roof so proper coordination difficult task. Consider the situation when two teams working in separate locations in different cities. Is there coordination equivalence to two teams sitting in the same physical location?

**f) Re-factoring:** Before the start of project overall project planning is not recommended, initially the next iteration is planned which is priority based. So there will be a regular lacking of long term planning. This means that when various components of the project have been assembled definitely this will result in some problems. In agile methodologies this is called “re-factoring”.

**g) Weak Documentation:** As agile technology always believe in persons instead of heavy documentation and it is based on a verbal communication with customers this will lead to a weak documentation.

**h) Team development:** Beside that one of the key factors behind the success of agile methodologies is based on having team of experienced and highly qualified developers having the creativity and skills to works in teams and to communicate with other teams or customers. This type of developers is not easy to find and demand a huge cost. Also working in small teams guide to troubles in identifying individuals' contributions and how they can be pleased. Beside that lack of enough documentation for tracking overall progress make executive managers face huge challenges of not delivered goals.

## 9. Conclusion

Due to availability of different software development models in the market ambiguity gets arise for the organizations that which model to choose. One possible and common way of selection is market trend, simulation based study to evaluate latest technology coming in the market and another method is thorough evaluation of Pros and Cons of the different models or technologies. Business environment and market scenarios are not stable due to competition in the market, so dynamic changes and repeatedly changing client requirements software development has become very complex job. Agile technology is considered one of the best methodologies to deal with such dynamic changes due to its iteration based working mechanism. For small and big projects with frequently changing requirements Agile is recommended. But before actual implementation organization always prefer to study its Pros and Cons with respect to organization's own resources.

In this paper authors tried to explore the working of two most commonly used agile based methodologies XP and Scrum. Base will remain the same for both of the technologies so in the last section, strength and weakness of agile methodologies are highlighted. Agile methodologies are great methodologies which appears as a respond to frequent changes in market and user specifications. This paper demonstrate main strength points of agile methodologies such as its philosophy based on short iteration based development cycle, high customer involvement/satisfaction starting from first day of project to delivery of the product and a quick adaptation to change. Working strength of agile methodologies respond to changes in markets, due to which many organizations find these methodologies more suitable for market. This paper refers to some weak points too in agile methodologies. After studying the weakness and strength points of agile organizations will be able to decide the concerned required methodology.

### References

- Boehm, B. and Turner, R. (2003). *Balancing Agility and Discipline: A Guide for the Perplexed*. Boston: Addison-Wesley Longman Publishing Co., Inc.
- Beck, K. and Fowler, M. (2000), *Planning Extreme Programming*. First Edition. Boston: Addison-Wesley Addison-Wesley Longman Publishing Co., Inc.
- ISACA (2014). *Agile management: myth or reality?* Available at [www.isaca.org/Blogs/166664/archive/2010/06/08/Agilemanagementmythorreality.aspx](http://www.isaca.org/Blogs/166664/archive/2010/06/08/Agilemanagementmythorreality.aspx)
- Larman, C. (2003), "Iterative and Incremental Development: A Brief History", *IEEE Computer*, June 2003.
- Mohammad, A.H., Alwada'n, T. and Ababneh, J. (2013). Agile Software Methodologies: Strength and Weakness. *International Journal of Engineering Science and Technology*, Vol. 5 No.03 March 2013.
- Rao, K.N., Naidu, G.K. and Chakka, P. (2011). A Study of the Agile Software Development Methods, Applicability and Implications in Industry. *International Journal of Software Engineering and Its Applications*, Vol. 5 No. 2, pp. 35-46.
- Serena Software Inc. (2007). *An Introduction to Agile Software Development*. Available at [www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf](http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf)
- VersionOne Inc. (2014). *Agile Methodologies for Software Development*. Available at <http://www.versionone.com/Agile101/Agile-Development-Methodologies-Scrum-Kanban-Lean-XP/>
- Williams, L. (2007). *A Survey of Agile Development Methodologies*. pp. 209-227. Book Chapter available on <http://agile.csc.ncsu.edu/SEMaterials/AgileMethods.pdf>.
- Guha, S. (2011). *Extreme Programming (XP) What's that!* Available at [www.agilewaybysuman.blogspot.in/2011/01/extreme-programming-xp-whats-that.html](http://www.agilewaybysuman.blogspot.in/2011/01/extreme-programming-xp-whats-that.html)