Pacific Science Review B: Humanities and Social Sciences 2 (2016) 22–28

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Pacific Science Review B: Humanities and Social Sciences

journal homepage: www.journals.elsevier.com/pacific-science-review-b-humanities-and-social-sciences/

Study of agile methodology with the cloud

Shariq Aziz Butt

Superior University of Pakistan, Pakistan

ARTICLE INFO

Article history:
Received 30 April 2016
Accepted 8 September 2016
Available online 23 November 2016

Keywords:
Information systems
Agile methodologies
Cloud computing
Software management processes

ABSTRACT

This paper reviews different Cloud computing and agile development methodologies with new technologies that have come with new approaches. In the future, we can develop better quality software and can also provide computing services. However, the synergy between the two is bonded with technical and non-technical challenges. Agile methodologies are considered very competent in software development. Agile development techniques are very practical for accepting information and are necessary when the industry atmosphere changes.

1. Introduction

The techniques of agile software development have become popular during the last few years. For traditional Project Management, the agile methodology is widely used in software development to increase the quality of a project as well as to enhance customer satisfaction. Agile methods are light-weight software schemes (Misra et al., 2006). The old traditional techniques (Waterfall, Unified Process, prototyping Model and Spiral Model) are not capable for software development nowadays because new requirements are taking place in the market. The new software development techniques include XP, Scrum, Crystal, FDD, DSDM, and ASD (Mani and Deebitha, 2014).

Cloud computing is entirely based on Internet facility. During the recent past, cloud computing has reached great esteem and developed an important style in IT. Cloud computing is a model that enables suitable, on-demand networking access to a pool of public and configurable computing resources that are quickly provisioned with the negligible management attempt or services supplier contact (Bogdan et al., 2014). A cloud, in fact, is a network of computers serving as a "service-oriented" construction to bring software and data. The cloud research design at high speed focuses only on newly joined customer (Gangadhar et al., 2015).

A number of organizations all over the world are trying out a variety of accessible agile development methods.

2. Literature review

A number of studies have been carried out in the literature show that agile methodologies and cloud computing has based its features on the necessities for small and large organizations as well as on an understanding of the project group. In the following section of the paper, two different methodologies for cloud computing and agile methodologies that are widely used in business are discussed. Agile development methodologies and cloud computing are an excellent combination. A brief review of important research studies carried out in last two decades has been presented in following section of the paper. Boehm and Turner (2005). Generally, agile methods are lightweight processes that utilize short iterative cycles. It enthusiastically engages users to establish, prioritise, and ensure that necessities are taken care of as well as relies on a team's unspoken knowledge as different to certification (Boehm and Turner, 2005).

According to Cho (2008), he highlights the differences connecting agile and traditional software development methods and describes the scrum framework, which is a well-known agile method. The author also discovers issues and challenges in scrum in an in-depth case study. The identified issues are comprised of certification, user participation, operational environment and scrum ceremonies. The issues and challenges exposed by the authors on the foundation of one holder study and the dialogue of nine employees is inadequate to set-up a statement concerning the method (Cho, 2008).

As stated by Cao et al. (2009), the agile advancement is predictable for considering the challenges connected to the developers, which could influence its achievement considerably. The

E-mail address: shariq2315@gmail.com.

Peer review under responsibility of Far Eastern Federal University, Kangnam University, Dalian University of Technology, Kokushikan University.

http://dx.doi.org/10.1016/j.psrb.2016.09.007

method generally depends on the contained knowledge of the developers. The developers might not document vital decisions, and the deficiency of prescribed record-keeping regarding the project may possibly make things more complicated for the teams to sketch and comprehend the structure (Cao et al., 2009).

As per Hneif and Ow (2009), the discussion on three agile methods of extreme programming, scrum, and agile modelling illustrate the differences amongst them. The authors also present a number of suggestions concerning how these methods should be followed by the practitioners (Hneif and Ow, 2009).

According to Pallis (2010), in view of present demand from employment, the individual requirements for online appointments, and the expansion of the net, cloud computing could be a demonstration of an innovative paradigm of a large-scale circulated computing efficacy for business and social order solutions (Pallis, 2010).

Consistent with Hochmuller (2011), in agile software development, the purchaser or client ambassador is responsible for explaining supply. For a while, clients did not contain the essential skills to illustrate their needs. To rise above this subject, the prerequisite engineer should co-operate with the developers and customers. The author also highlights several reimbursements that can be achieved if the engineer works shoulder by shoulder with the customer. This move towards customer–developer relationships is the foundation for the agile software development process. To achieve real benefits, the engineer should be a specialist in other areas of software engineering, such as cost estimation, requirements confirmation, and corroboration (Hochmuller, 2011).

According to Mazni et al. (2011) the agile methodology has an important impact on software development. The methodology positively influences the developers since it can be introduced and included into the services offered for the team members (Mazni et al., 2011).

As per Khajeh-hosseini et al. (2012), development methodologies, such as agile methodologies, emphasise detailed practices that may perhaps carry issues of concern in the form of non-technical and technical problems related to cloud computing. The user and developer communicate the restrictions, and the programming environment incorporates examples of non-technical and technical problems, respectively (Khajeh-Hosseini et al., 2012).

In view of the existing demand from work and personal requirements for online arrangements and the expansion of the web, cloud computing could demonstrate a new paradigm of a large-scale disseminated computing efficacy for business (Venkatraman and Wadhwa, 2012). Cloud computing has many benefits, but it also has challenges, such as security concerns; data ownership concerns; lock-in and interoperability concerns; enterprise support and service development issues; condition for online connectivity; and there is concern among developers about a new cloud computing stage without suitable supervision, which is focuses on how to successfully make use of typical cloud computing architecture (Venkatraman and Wadhwa, 2012).

Consistent with Werfs et al. (2013), in principle, the cloud computing environment facilitates agile development teams also add their expertise by continuously developing software manufactured goods (Werfs et al., 2013).

3. The significance of the study

This article provide an overview of various agile methodologies with an overview on the literature of this methodology, which identified the key practices of agile methodologies and comparatively analysed the five methodologies on the basis of identified practices. This paper provides an analysis of unique practices for agile methodologies, a comparison between the methodologies, a summary of the practices that are common to agile methodologies, and provides an understanding of appropriate methodology combinations.

4. Agile modelling

Agile modelling is a methodology that will help for further document software systems based on optimum practices. It is an assembly of standards and ethics that agile software increasing uses (Moniruzzaman and Hossain, 2013). Moreover, agile modelling is similar to other methodologies, such as Scrum, Excessive Programming (XP), and Rational Unified Process (RUP). Like XP, agile modelling expounds that changes are simple in software development. Agile modelling also highlights the difference between everyday models whose solitary reason is to assist face-to-face communication and models for system documentation (Rajasekhar and Mahammad Shafi, 2014).

5. Agile methodologies

The difference between agile development and traditional software is not negligible and code oriented. The most important agile methods are listed in (Singh and Chana, 2013; Tuli et al., 2014; Werfs et al., 2013) (Fig. 1).

- Crystal methodologies family
- Extreme Programming
- Feature-Driven Development
- SCRUM

5.1. Crystal methodologies family

In 2000, Alistar Cockburn discovered the Crystal Methodologies families. They focus on effectiveness and habitability of tools for project safety (Moniruzzaman and Hossain, 2013). There are different types, including Clear Crystal, Yellow Shining, and Orange Crystal. Moreover, Red Crystal is good due to the different number of people. It is also best depending on the load of projects. All of the Crystal methodologies urge especial roles, platform standards, and lines must be adopted. Crystal Clear, which is one of the Crystal methodologies, is for development teams of six to eight members with a focus on activity on non-life sensitive systems (Hasaba and Faraahi, 2014; Kumar and Bhatia, 2012).

5.2. Extreme Programming (XP)

Extreme programming (XP) is programming that helps agile, lightweight software development. It became popular during the

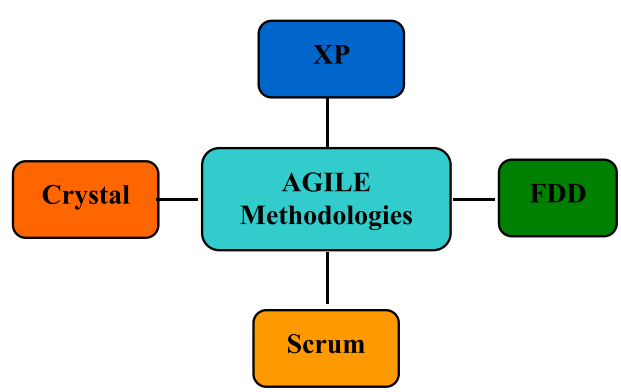

Fig. 1. Agile methodologies.

modern age. Kent Beck introduced XP in 2000 (Nawrocki et al., 2002). XP gives a list of straightforward, definite, and apparently inexperienced features and points that lead to software development procedures. The four stages of software development are given below:

- Planning
- Coding
- Designing
- Testing

The main purpose is to fulfil the needs of the customer when the requirement arises (Kumar and Bhatia, 2012) (Fig. 2).

Proof. C-MWPC can be easily reduced to the MWPC problem assuming a coalescence graph without any edge or a fully connected interference graph. Therefore, each C-node is an uncoalesced live range after value separation and C-PC is equivalent to PC. A fully connected interference graph is made possible when all live ranges interfere with each other. Thus, the C-MWPC problem is NP-complete (Table 1).

5.3. Feature-driven development

Feature Driven Development is a process that has five steps and no training is required. The first three steps explain and define the overall design of the model and its uses. These also help for desired properties, and the list of principles for the implementation of the plan. The last two steps are where the development may need to occur (Hasaba and Faraahi, 2014; Rajasekhar and Mahammad Shafi, 2014).

5.4. SCRUM

Ken Swaber was the person who first introduced the term SCRUM in 1995. This methodology was designed to solve the repeated changes in business requirements. SCRUM methodology is only used to simplifying the project during easy procedure, easy to update documentation and highly mutual relationship between the members of the team (Misra et al., 2006; Hasaba and Faraahi, 2014; Kumar and Bhatia, 2012) (Table 2, Fig. 3).

6. Cloud computing

In academia and also in the industry, cloud computing is used on a large scale. Using a computer network, its requirements are passed on directly. For running applications, it allows business to access and share data centre space. Due to the key advantages cloud computing is upcoming in the world (Gangadhar et al., 2015) (Fig. 4).

6.1. Advantages

- No Need to change hardware and software
- Speedy and easy use.
- Very few chances for failures.
- Work environment is customizable.
- A project solution with combination is effortlessness (Nazir et al., 2016).

6.2. Categories of models

- **Infrastructure as a Service (IaaS):** abstracted hardware and virtual machines and operating systems provided.
- **Platform as a Service (PaaS):** Clients allow to expand different applications by means of APIs, implementation. Platforms consist of improvement tools, improvement platforms and arrangement supervision and deployment of platforms (Bogdan et al., 2014).
- **Software like a Service:** third party presented this type of software. Software will be providing when its need require. It is provided through a web browser. This software is a remote

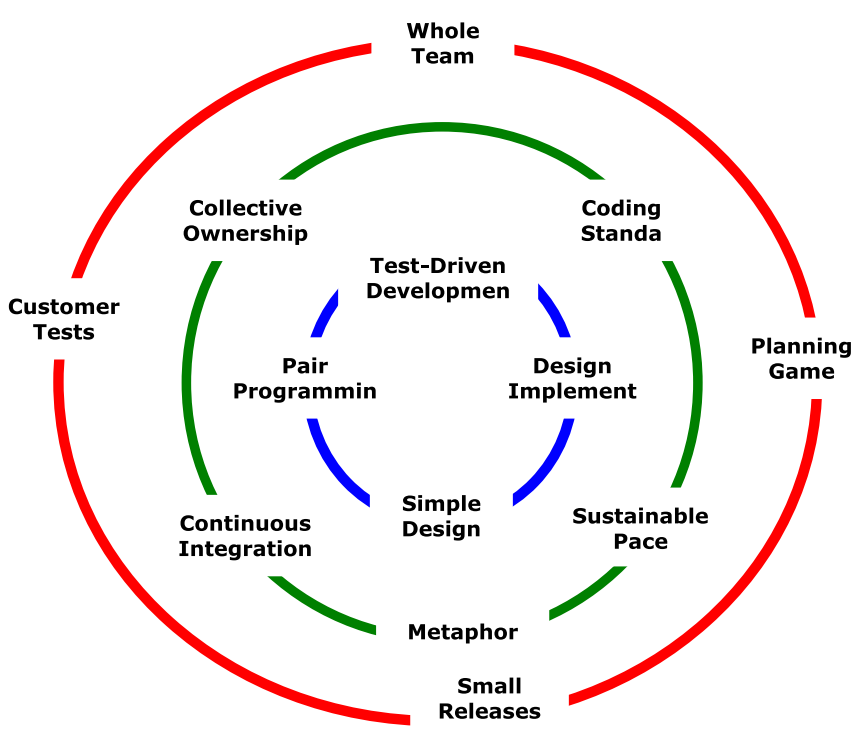

Fig. 2. Xp practices.

Table 1
Extreme programming.

Practice	Description
The planning game	This practice suggests a secure relationship between the client and technical team of the project. Every component is dependent on identifying a place in the project and particular attributes
Small Releases	The objective of this practice to quickly put in production to implement each new small set of attributes. Each release should be as soon as possible and also contain the most important aspects for the client
Metaphors	Metaphors permit describing the characteristic by creating a general point of view of the client and the technical team on how an item for consumption should work.
Simple design	The code including all units test and architecture should be as simple as possible
Test Driven Development	The entire implemented features should be exposed in a unit test, which must always be satisfied, in an attempt to remove unit level and regression bugs for the period of development
Refectory	Refastening aims make simpler to implement code by removing the code uncertainty and ambiguity.
Pair programming	That type of practice containing two types' programmers functioning at the same time on the same computer. Every programmer has his own specific role
Collective code ownership	All team members are optimistic to perform every type of essential changes in the code if required
Continuous integration	When new features are implementing or any type of code is fixed, and if all tests are effectively executed, a new release must be created that facilitates all the changes
One site customer	XP not only proposes a mutual relationship with the client, but in addition to that client should all the time be available throughout the life cycle of the project

Table 2
Scrum methodology.

Practice	Description
Burn down chart	This type of chart needs to be updated daily. It shows the revamped work day by day. This chart is used both to decide and sprint progress
Product backlog	This provides a complete list of the requirements that included bugs, requests, usability and performance not currently in use in the product release
Scrum Master	The person responsible for controlling and managing the SCURM project
Sprint backlog	The complete list of backlog items assigned to the sprint, but it yet not complete

control operating system (Rajasekhar and Mahammad Shafi, 2014; Bandana and Harshitha, 2013).

6.3. Cloud deployment models

Cloud deployment models use private cloud and public cloud IaaS for central applications and workloads and for customer-facing operation. Hybrid cloud might exist, which is an organization seeking a sense of balance between public cloud, which the federal (or localized) IT has powered over it; whereas others may change from public clouds, which are inherited applications and datacentre systems, to conclude their life cycle and switch to an agile application instead (Rajasekhar and Mahammad Shafi, 2014; Singh and Chana, 2013; Bandana and Harshitha, 2013).

7. Why agile?

Agile development methodology offers a number of ways to evaluate the trend of the project throughout the advancement of the lifecycle. If we want any type of achievement, it requires regular work, such as sprints. In an agile model each and every aspect is for this purpose, and it is frequently revisited throughout the lifecycle (Gangadhar et al., 2015; Jain and Rani, 2011) (Table 3).

8. Cloud computing and agile, a great combination

Athletic development processes improve (as much as possible) the opportunity given by computers that do work for you, but that are stored somewhere else and maintained by other companies as a result of releasing software over and over again. The clients point is to assist the organizations and to check and observe each and every fragment for upgrading standards (Jain and Rani, 2011; Rajasekhar and Mahammad Shafi, 2014; Singh and Chana, 2013; Hasaba and Faraahi, 2014) (Fig. 5).

Combining agile development and cloud computing brings the best of both worlds. Different organizations are using different methods for computing. It makes software faster than before. The cloud has provided quality. There are different advantages when we combine both agile and the cloud (Singh and Chana, 2013; Khan et al., 2011; Tuli et al., 2014).

8.1. Distributed application development

The cloud assists conveyed application advancement with groups scattered topographically. Any colleague can have rights to get data about the application at any minute.

8.2. Data sharing

Assets are obliged to share a colossal measure of information between diverse machines for some applications. The cloud assumes a significant part by giving a medium to share a great deal of information according to the prerequisites.

8.3. Prioritizing tasks

It is important to consistently organize and prioritize errands. In a conveyed framework, there is a necessity to redesign devices. This expends vitality and requires some investment (Nazir et al., 2016; Tuli et al., 2014).

8.4. Transparency

Straightforwardness is important. In a cloud-based agile model, the engineer can be straightforward and live up to expectations.

8.5. Infrastructure

Dexterous programming advancement obliges the infrastructure to be set up from the beginning. The cloud helps by providing a base (Werfs et al., 2013).

8.6. Benefits for business users

The cloud encourages spry practices with a specific end goal to meet the clients' prerequisites faster. The cloud underpins the idea of “pay per use” (Table 4).

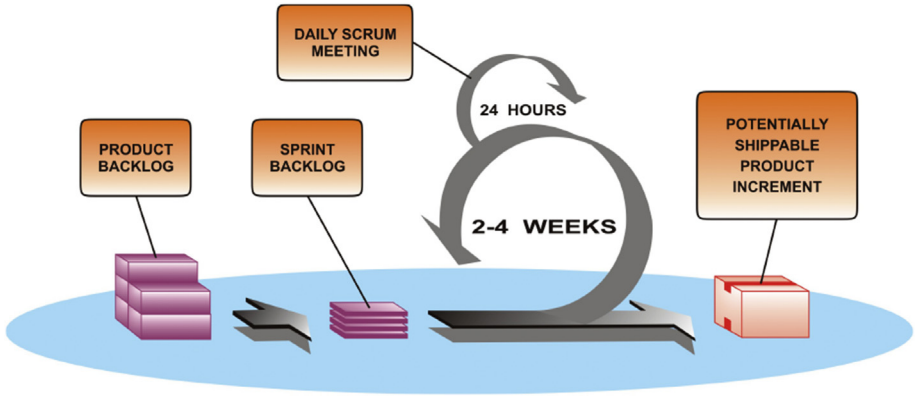

Fig. 3. Scrum practices.

Fig. 4. Cloud computing services.

Table 3
Principles of agile methodologies (Kumar and Bhatia, 2012; Mani et al. March 2014; Tuli et al., 2014 and Nazir et al., 2016).

1.	The first priority is to satisfy the clients through timely and fast deliverance of important software as soon as possible
2.	Changing requirements are welcomed for the customer's competitive advantage. Agile processes harness change
3.	From a few weeks to few months, software delivered quickly, with shorter time periods
4.	Through this project business people and developers must work together
5.	Motivated individuals with projects must be supported and they need to be trusted
6.	The best and efficient way of providing information is face-to-face discussion within a development team
7.	For primary measure the advancement functioning of software is in progress
8.	Indefinitely the facilitator, developers, and users are supposed to be able to retain a regular tempo through sustainable development, which agile processes promote
9.	We increase technical superiority and agility with permanent awareness.
10.	A self-organized team must have the best architecture, needs and designs
11.	Self-organizing teams merge design with the best architecture
12.	The team must know how to become more successful due to usual intervals rather than tuning and correcting its manners

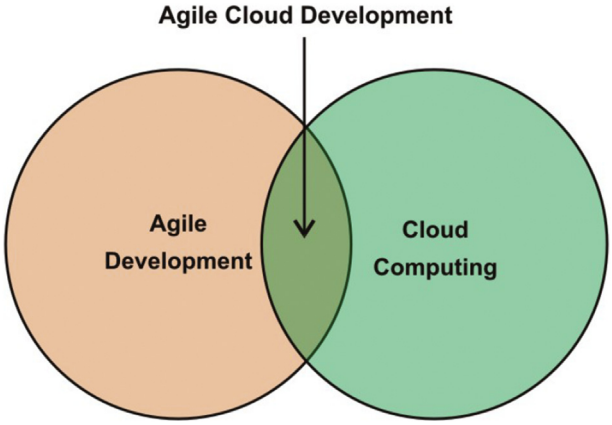

Fig. 5. Software development processing.

9. General features and comparison of methodologies

In this article, we develop the first multi-frequency MAC protocol for WSN applications in which each device adopts a single radio transceiver. The different MAC design requirements for WSNs and general wireless ad-hoc networks are compared, and a complete WSN multi-frequency MAC design (MMSN) is put forth. During the MMSN design, we analyse and evaluate different choices for frequency assignments and also discuss the non-uniform back-off algorithms for the slotted media access design.

10. Recommendations for future research

- Cloud computing combined with agile development will become very useful for the world. Cloud computing, describes the way of delivering software applications. Nowadays, it is efficient to consider the delivery of cloud computing software applications.
- Due to advancement in technology, there is an increasing range of dealing with systems and increasing the storing superiority of networking since everywhere has a high-bandwidth network. Right uses, and greater precautions and dependability than before the Internet.
- Ranging from increased processing due to the most advanced technology, increasing the area of networks data storage, and the most remarkable thing is the consistency in Internet protection.
- The upcoming possibility of this effort is to be examined and to include threat factors in using Cloud Development scientifically

Table 4
Comparison of agile methodologies.

Characteristics	Xp	Scrum	FDD	Crystal
Development approach	Iterative Increments	Iterative Increments	Iterative	Incremental
Recommended iteration time period	One to six weeks	Two to four weeks	Two days to two weeks	Depending on method belong to family
Project team	Smaller teams (less than twenty members)	All sizes (concept of scrums)	Many member more than one team	All sizes Depending on method belong to family
Team Communication	Daily Stand up meetings (Informal)	Daily Stand up meetings (Informal)	Documentation Based	Face-to-face Informal
Customer Involvement	Customer Involvement	Through Customer the role of Product owner	Reports through Customer	Customer through incremental releases
Project Documentation	Basic documentation	Basic documentation	Importance of Documentation	Basic documentation
Specialties	Refactoring, User stories, TDD	scrum master, planning poker, product, Sprint and sprint backlog,	Diagrams of UML	Adaptable method from family All kind of projects and size of team
Quality	Good Quality	Good Quality	High Quality	Good Quality
Time	Short Time Duration	Short Time Duration	Long Time Duration	with real-time question and answer

From the above table we conclude that it is very practical, and is intended to be informative as to how successful the adoption of each method has actually been.

Features	Different process models		
	Agile process	Spiral model	RAD model
Definition	Agile process is the ability to both create and respond to changing requirements of software	Spiral Model is the software development model which focuses on managing risk	RAD model is " high speed adaptation of linear sequential model, in which component based construction is used
Adaptability	x	√	x
Testing Phase	U nit, Integration and System Testing	U nit, Integration and System Testing	Unit
Quality Factors	√	√	x
Risk Analysis	x	√	x
Off-the-Tools	x	x	√
Failure normally due to	Code	Code	Architecture and design
Knowledge Required	Product and Domain	Product and Domain	Domain
Entry & exit Criteria	x	x	x
Mock up	√	√	x
Extendibility	√	x	√
Project management involvement	√	√	x
Higher Reliability	√	x	√
Time Boxing Use of reusable components	x	x	√
Flexibility	√	√	x
Customer Involvement	√	√	**x**

Table 5
Indicate the interpretation of the goal of each methodology from Table 1.

Condition	XP	Scrum	FDD	Crystal
Small Team	√	√	X	–
Highly Volatile Requirements	√	√	√	–
Distributed Teams	X	√	√	X
High Ceremony Culture	X	X	–	–
High Criticality Systems	X	–	–	√
Multiple Customers/Stakeholders	X	√	–	–

and discover serious achievement factors of the Adaptive Cloud, including improvements as well to discover a variety of hazard factors by means of threat investigation for introducing reusability in agile cloud expansion.

- In Agile cloud development, reusability is capable of automatically using an automated apparatus. The automated form resolves or decreases the development expenditure while lifting the reusability and client fulfilment to a huge degree.
- In this study we want propose studying various projects that use unpredictable agile methods and also cloud computing. This motivation allows us to recognize further significant factors with the aim of assisting agile methods owing to their elasticity.

11. Conclusion

The aim of this paper was to give an overview of agile methodologies with cloud computing. In this review, we try to locate and describe the important changes that are required and the reason why agile software development practices and cloud computing should be used in traditional development organizations. Agile Software Development and cloud computing have brought us lots of high-quality stuff in software development. The main insight is the enhanced excellence equality of products, improved efficiency of developers and less errors. The continuously increasing acceptance of agile methods proves their efficiency and usefulness. The agile process ensures frequent interfacing between developers and customers, as well as contributes significantly to its investment. In this study, we contribute to the body of knowledge on the comparative analysis of agile methodologies. The comparative analysis of agile methodologies, specifically Crystal, XP, FDD and SCRUM, as well agile computing relationship (Table 5) with other methodologies conclude that even though the agile methodologies consist of a number of new software development approaches, but they also perform different tasks on various parameters. Along with all agile methods, SCRUM has maximum implementation, while software practitioners prefer XP.

References

Bandana, H., Harshitha, T.M., September 2013. An effective way to apply agile manifesto in development of cloud application through agile service networks. 3 (9).

Boehm, B., Turner, R., 2005. Management challenges to implementing agile processes in traditional development organizations. Softw. IEEE 22 (5), 30–39.

Bogdan, Ghilic-Micu, Marian, Stoica, Cristian Răzvan, Uscatu, 2014. Cloud computing and agile organization development, 18 (4).

Cao, L., Balasubramaniam, R., Kannan, M., Xu, P., 2009. A framework for adapting agile development methodologies. Eur. J. Inf. Syst. 18, 332–343.
Cho, J., 2008. Issues and challenges of agile software development with Scrum. Issues Inf. Syst. IX (2).
Gangadhar, P.V.S.S., Shrivastava, A.K., Shukla, Ragini, Apr-2015. To implement cloud computing by using agile methodology in Indian E-Governance. 02 (01).
Hasaba, Seyed Amin Mousavi, Faraahi, Ahmad, 2014. An overview of applying cloud computing in addressing agile global software development challenges. 40 (1).
Hneif, M., Ow, S.H., 2009. Review of agile methodologies in software development. Int. J. Res. Rev. Appl. Sci. 1 (1).
Hochmuller, E., 2011. The requirement engineer as a Liaison Officer in agile software development. In: Proceeding of AREW.
Jain, Abhishek, Rani, Reena, 2011. Analytical Study of Agile Methodology with Cloud Computing. RTMC.
Khajeh-Hosseini, A., Greenwood, D., Smith, J.W., Sommerville, I., 2012. The Cloud Adoption Toolkit: supporting cloud adoption decisions in the enterprise. Software Practice Exp. 42 (4), 447–465.
Khan, Ahmad, Shaikh, Parez, Dhembre, Chetan, Gawali, Sushant, November 2011. Cloud services for collaborative web based project management system. 8 (6), No 2.
Kumar, Gaurav, Bhatia, Pradeep Kumar, August 2012. Impact of agile methodology on software development process. 2 (4).
Mani, Pavithra, Deebitha, S., March 2014. Analysis of agile software development utilising cloud computing capabilities. 3 (10), ISSN: 2278–3075.
Mazni, O., Sharifah-Lailee, S., Azman, Y., 2011. The impact of agile approach on software engineering teams. Am. J. Econ. Bus. Adm. 3 (1), 12–17.
Misra, Subhas C., Kumar, Uma, Kumar, Vinod, Grant, Gerald, 2006. The Organizational Changes Required and the Challenges Involved in Adopting Agile Methodologies in Traditional Software Development Organizations. IEEE.
Moniruzzaman, A.B.M., Hossain, Syed Akhter, Jul 12, 2013. Comparative Study on Agile software development methodologies.
Nawrocki, Jerzy, Jasiski, Michal, Water, Bartosz, Wojciechowski, Adam, 2002. Extreme Programming Modified: Embrace Requirements Engineering Practices. IEEE.
Nazir, Ambreen, Raana, Ayesha, Fahad Khan, Muhammad. Cloud computing ensembles agile development methodologies for successful project development, doi: 10.5815/ijmecs.2013.11.04.
Pallis, G., 2010. Cloud computing: the new frontier of internet computing. Internet Comput. IEEE 14 (5).
Rajasekhar, P., Mahammad Shafi, R., 2014. Agile Software Development and Testing: Approach and Challenges in Advanced Distributed Systems. 14 (1) Version 1.0.
Singh, Sukhpal, Chana, Inderveer, 2013. Introducing agility in cloud based software development through ASD. 6 (5) 191–202.
Tuli, Anupriya, Hasteer, Nitasha, Sharma, Megha, Bansal, Abhay, July 2014. Empirical investigation of agile software development: a cloud perspective. 39 (4).
Venkatraman, S., Wadhwa, B., 2012. Cloud computing a research roadmap in coalescence with software engineering. Softw. Eng. An Int. J. (SEIJ) 2 (2).
Werfs, M., Robert, T., Baxter, G., Allison, I.K., Sommerville, I., 2013. Migrating software products to the cloud: an adaptive STS perspective. J. Int. Technol. Inf. Manag. 22 (3).

Further Reading

Inayat, Irum, Salim, Siti Salwah, Kasirun, Zarinah M., 2013. Agile-Based Software Product Development Using, 1065–69, Cloud Computing Services: Findings From A Case Study.
Livermore, Jeffrey A., 2007. Factors that Impact Implementing an Agile Software Development Methodology. IEEE.
Mani, Pavithra, Deebitha, S., Selvakumar, Jayakumar, Rathi, Gopalakrishnan, Jan-March 2014. Enhancing agile software development using cloud computing: a case study. 1 (1).
Paetsch, Frauke, Eberlein, Armin, Aurer, Frank, Jun 9, 2003. Requirements Engineering and Agile Software Development.
Ravi, Satya Prasad, Sridhar Movva, Lakshmi, Kilaparthi, Rajasekhar, Reddaiah, B., June 2012. Agile Cloud Computing. 2 (3).
Timperi, Olli P., 2004. An overview of quality assurance practices in agile methodologies. SPRING.
Zhong, Shi., Liping, Chen., Tian-en, Chen. Agile planning and development methods, ©2011 IEEE.