Current Journal of Applied Science and Technology

35(6): 1-12, 2019; Article no.CJAST.49500
ISSN: 2457-1024
(Past name: British Journal of Applied Science & Technology, Past ISSN: 2231-0843,
NLM ID: 101664541)

Modeling Distributed Agile Software Development Utilizing Cloud Computing: A Holistic Framework

Ali Sever[1*]

[1]*Department of Computer Information Systems, Pfeiffer University, Charlotte, NC, 28109, USA.*

Author's contribution

The sole author designed, analysed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/CJAST/2019/v35i630206
Editor(s):
(1) Dr. Rodolfo Dufo Lopez, Professor, Department of Electrical Engineering, University of Zaragoza, Spain.
Reviewers:
(1) M. Mudasir Kirmani, Sher-e-Kashmri Unviersity of Agricultural Sciences and Technology, India.
(2) Anthony Spiteri Staines, University of Malta, Malta.
(3) Yulia Informatics, Petra Christian University, Indonesia.
Complete Peer review History: http://www.sdiarticle3.com/review-history/49500

Original Research Article

Received 24 March 2019
Accepted 06 June 2019
Published 11 June 2019

ABSTRACT

In distributed agile software development, to conquer the problems raised by rapid development of complex products with ever changing technological advances and risks, there is a need of a model with specific tools and technology supported approaches to manage them. Distributed environment has challenges and complexities in many ways of over collocated teams. However, they seem to enhance the basic principles in which agile methodology thrives on. Agile projects and distributed projects are compatible and therefore practices, principles, tools, and techniques for distributed agile software development are introduced to the distributed teams to ensure that the agile projects become successful. Although cloud computing is well established for software system developments, its effect on agile methodology has not been extensively studied. This study outlines how agile projects can be modeled in the cloud-computing era, namely, seamless agile-cloud integration. We propose a model to improve Agile Methodology in distributed computing platform. This model has uncovered a gap between the distributed agile development phases that are overemphasized and the phases that require actual attention. In addition to this, emphasis on appropriate phases would help build desired software that can exploit the benefits of cloud computing in numerous ways.

**Corresponding author: E-mail: delafener@gmail.com, ali.sever@pfeiffer.edu;*

Keywords: Distributed agile software development; agile methodologies; cloud computing; machine learning; agile software development.

1. INTRODUCTION

A distributed team is commonly defined as a group "of geographically, organizationally, and time dispersed workers brought together by information and telecommunication technologies to accomplish one or more organizational tasks" [1]. The distributed team system, thus, faces many issues like communication gap, lack of understanding, decreased productivity due to slow process, improper communication, difference in time zones and cultures.

The problem we studied in this paper is how cloud-computing framework can be applied to distinct agile software development applications. Our aim is to discuss potential improvements in distributed agile methodology while understanding the limitations of current processes. Initially, it is important to consider current implementations of agile methodology to understand where software companies currently stand. It is very important to identify constraints that are holding this form of development back. Additionally, it is critical to understand the difficulties, which software engineers face when developing agile software projects. With the phenomena of cloud computing, it is becoming an increasingly difficult challenge to manage agile software development projects. Mainly, our goal is to identify the great potential that exists for agile methodology in the realm of cloud computing in regards to the software development projects in a holistic framework. Although cloud computing is well established for software system developments, its effect on agile methodology has not been extensively studied. A great discussion in [2] and [3] emphasize that a little work has been carried out to research this issue and previous works have not comprehensibly considered cloud agile integration. Indeed, at present, general research in agile-cloud integration systems is still in its infancy.

Agile Project Management is a unique approach in developing software projects. It is a way to manage research and development of new products and projects that have high risk and are time sensitive. This approach needs rapid development and deployment of products, for that we need team members to be positioned at a location to work on the project. Although collocated team structure is the center of the agile project management, it cannot be possible all the time. There might be situations where teams would be in different locations and might need to work together towards a common goal of the project. The teams in this scenario are referred to as distributed teams. Distributed teams are a cost-efficient way of getting talented individuals from various locations. The below discussion is going to cover how both agile methodology and distributed teams can work together in the cloud computing framework (Fig. 1).

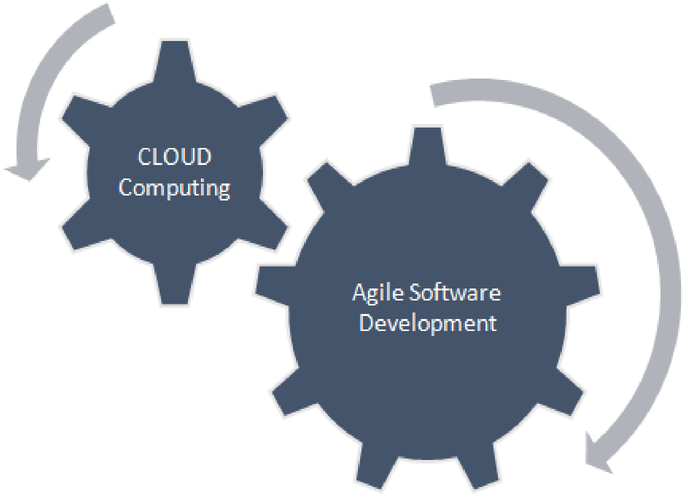

Fig. 1. Agile-Cloud integration

Business environments are, nowadays, extremely agile and new technology commands industries to react faster in order to maintain a competitive edge. Organizations are looking for cost-effective ways to allow their workforce to collaborate on the cloud and still accomplish tasks with the same velocity as face-to-face communication [4]. The trend towards developing a globalized software was originated by software intensive high-technology businesses. In larger enterprises, global software development has become the norm involving distributed development. To achieve this, multinational companies and large business organizations use the Agile Methodology [5].

One also has to consider agile methodology from a project management perspective. Agile Project Management is a dynamic approach that is widely used in a fast moving business environment. It facilitates in quick development and delivery of a product. To achieve this with distributed teams, several methods have been studied [6,7] and certain discussions [8] have been made, eventually leading to conditional results [9,10, 11,12]. Common repositories were used to share the work progress so that efficiency is maximized during that development process. Distributed agile team enhances agile project management since there is huge scope to select people with proper skills. Some factors such as cultural differences, language compatibility, and time zones have to be taken into consideration [13]. Some of the results conclude that distributed teams can improve or worsen distributed agile project management depending on how well they are implemented [14,15,16].

Agile methodology must incorporate rapid development and faster delivery of the product to clients and this requires a strong communication system among the team members. It was initially introduced for clustered teams. As teams are distributed globally, creating strong communication becomes very difficult. This happens as people work in different time zones that increases the wait time. In addition, they belong to different cultures and speak different languages. At first, it was believed that distributed teams might not be utilized in agile project development, however, it was found to be essential as in agile methodology. Team members may not always be collocated across the world. Therefore, distributed team members might be needed in agile development.

Even though the agile development is a productive approach, it has several complications that impede the growth of the final product. Information gap, lack of professional trust, loss of non-verbal cues and so many other reasons can be cited [17-21]. Nevertheless, this paper covers the benefit of a distributed team in the agile development in the cloud-computing context and provides an unique approach that can be used to avoid these common issues and to mitigate their impact.

A group of researchers presented a paper [22] in 13th International Conference on Global Software Engineering, which describes the common problems in managing agile methodologies among distributed teams. The way we approached these issues is that project teams will need to be created with a cloud mindset in future projects. In particular, the issues that distributed project teams faced being comprised of members from different regions were conflicts involving time zone differences, version control, asynchronous contributions, and members belonging to multiple teams. As shown in [23,24,25] distributed teams can work on projects around the clock and proper talent can be found in less competitive markets.

The remaining document is structured as follows: Chapter 2 emphasizes on agile methodology and cloud environment, existing systems and their disadvantages and challenges. Chapter 3 provides a proposed model and its architecture within the framework of multi cloud architecture. It explains in detail the system architecture, discusses the challenging tasks and solutions to manage these distributed teams. Chapter 4 depicts the results and discussions related to development projects and gives the description of modules. Chapter 5 gives conclusion and future enhancement and explains the need of it.

2. AGILE METHODOLOGY CHALLENGES

First, the business value of rich, high context face-to-face communication needs to be acknowledged. With that in mind, distributed teams do not have strong communication among the team members. Asynchronous teamwork increases the communication times. In addition, language, culture, leadership, and trust are major drawbacks for distributed teams. Some of the most common problems faced by distributed teams may be listed as follows:

1. Release management and continuous integration.

2. As the distributed teams are located in different locations, there is lack of communication, which indeed effects the integration of the project as the teams have different time locations.
3. There is no sense of ownership for the members of the distributed teams, as they do not have proper recognition for their works.
4. Distributed teams require physical presence for communication quality
5. The hierarchy of authority in an organization and functional structure need to specify ground rules and correct functioning of an organization.
6. The trade-off between flexibility and product deliverables would take its toll on documentation in agile methodology, which in turn creates difficulty in recruiting new project members in later stages.

The problem of utilizing passive means of communication model is that it places a risk on the project performance. A small error made by any of the distributed team members ends up affecting the whole project. Passive communication is one of the main challenges faced by distributed teams. That may create a loss of cohesiveness and the team members in a distributed environment may become individuals working on their own tasks instead of acting as a team.

Agile team members are all positioned working side by side making it easy for them to discuss their opinions, ideas and have a direct face-to-face communication whenever required and forming lightweight groups and planning, prioritizing, solving issues together and acting quickly to develop or to release a product. Agile project management is highly efficient and flexible, but in this world of development, there are many people so that positioning everyone to work on a project is not possible. This is the generation for distributed teams. This way people from varied locations, languages, culture and creativeness can collaborate their work through distributed teams, therefore, adapting distributed teams into agile methodology is needed in this era of technology.

With the advent of the development of Internet and cost-effective workplaces, collaborative software has changed the way we look at project development; it has given us a way to assign an agile project to a distributed team. Distributed teams are the new core of the several companies. As stated below, there are many reasons for the need of distributed teams.

- Not all the talent can be aligned on a project management model based only on collocated teams.
- No single team can develop every segment of the product. Teams need different members of distributed team to help them to get through the issues.
- The recent problems of rapidly developing complex new products, turbulent and dynamic market conditions, exponential rate of technological change, ever increasing competitive threats, and seemingly insurmountable risks are faced daily in order to satisfy enterprise, organization, and firm-level needs.
- Most of the current products are demanding frequent changes either in their workflow or the outcome or the look and feel based on the customer feedback. Distributed teams with resources having such adaptability will be apt in such scenarios.
- Even distributed teams having members who are from different cultures can help with proper feedback while planning/deciding on the services offered by the product.
- Creating a single team for the development of a project and working to satisfy the customer needs is not possible.
- Even cohesiveness among the team member’s tasks will be lacking. As the team members
  pay attention on their individual tasks instead of dealing with the single body of work.
- Team members can sometimes miss crucial information due to the differences in the time zones.

Even though agile methods have been thriving in the last decades, their implementation in globally distributed teams, still have some big challenges. However, many professionals argue that agile methodologies yield better results when there is a team collaboration and face-to-face communication. In today’s growing world, the needs of the client keep changing rapidly and the software engineers should be able to keep up with the pace of change and deliver the product, which gives complete satisfaction to the end users. With changing needs, the required talented professionals cannot be found at single location. In addition, with shorter deadlines, the teams must work continuously, which is possible with teams from different time zones. Hence, the importance of distributed teams has increased since the skilled professionals are distributed globally. Therefore managing distributed teams without compromising the agile philosophy is very important to success. In this paper, we present how company productivity may be increased by incrementally designing the product by using distributed teams over the cloud. In addition, we are discussing the challenging task of agile-cloud integration and solutions to manage these distributed teams.

In [13] and [14], the following are the proposed materials and methods to avoid the different challenges that distributed teams face when we adopt agile methodology. We should state clear objectives, work-plans and scope. We should assign proactive and skillful leaders for the project success. With this, the manager has access to all the information and can-do necessary changes to make sure the task is completed by the deadline, effectively. Even though the teams are located remotely, the work is being monitored continuously, which makes sure there is no delay or any kind of misconceptions in the project. Furthermore, we may have to use the experienced mentors for tracking the progress of the project. In addition, gather people periodically to discuss the progress of the project. All these have to be done with minimum synchronous activities while ensuring maximum effective communication as much as possible.

3. PROPOSED MODEL

The main problem here is adapting agile methodology to the distributed teams in the cloud platform. Agile feature selection has to be based on high-level principles on which agile methodology thrives. As stated in the agile manifesto, these principles essentially focuses on people, rather than tools, the product over its extensive documentation, collaboration with the customer rather than a contractual relationship, and dynamic change response over plan-based systematic development. Therefore, our proposed model for approaching this problem can be divided into four main categories. The model that is considered for distributed agile software development will be based on the simulation of the cloud operations on the following four pillars of agile development:

1. Dynamic change response
2. Product (working software)
3. Collaboration with customer
4. Team collaboration

In Fig. 2, the general architecture of the proposed approach shows a holistic view of the overall system. By the term holistic, we mean thinking about the many aspects that comprise the characteristics of agile and cloud methodology. The development process is broken down into three tiers. The first one is the process space that deals with project's central repository. The second tier is the model space that consists of agile models, domain models, and AI models, which help automation and integration techniques. The last one is the decision making tier. By employing the tier model, a more holistic approach to software and development operations is adopted. This way model guaranties that other important aspects aside from mere software integration become the focus and that a well-rounded approach to continuity and operations is provided. The model explores the decision-making process in agile teams using predefined model space that influence the decision-making process during the agile progress meetings. This ensures that development is open to change. Therefore, a more proactive approach is utilized in which the flow concept of agile principles 1-4 is prioritized. This will provide a smooth and continuous software system development as well as an improved awareness of the actual progress. The proposed holistic approach to developing a progress tracking system may also help support identifying and coordinating the various components on the agile-cloud integration.

Model operates in the simulation mode of DEPSKY [23] (Fig. 3). That is a multi-cloud architecture, which is accessed by its users by invoking operations in several individual clouds. Distributed systems are used to store data in different locations and brought together via software. This will mitigate the limitations of loss of availability, loss and corruption of data, and loss of privacy.

We attempt to establish several connections between pillars of agile methodology (1-4) and cloud computing that do not exist in the literature. We provide this under four sections.

3.1 Change to Response on the Cloud

The main idea of agile methodology is to quickly and efficiently respond to a change to develop and create a working software. Therefore, a quick turnaround is expected while developing a software. As the teams have global presence in the Cloud environment, working in different time slots project time-off is hardly a possibility. To succeed, the team need to be flexible and adapt to moving targets. A logical way to deal with this issue is to create the teams with the local workforce but it may not be possible in many cases to form collocated teams. In case of a high priority production issue, teams existing across different landscapes proves to be a better idea. Moreover, production plants run continuously and a software glitch or need to change a process urgently, may work better. A team, which is always available on call or on demand, is always preferred to handle such issues. This bypasses the local time zone issue and provides a fix to the problem in turn providing a smooth business operation. Since the team is distributed across the globe, at least one team member is most likely working on the project at all times. There is hardly any project off time. Hence, responding to the change is faster in the agile-cloud model

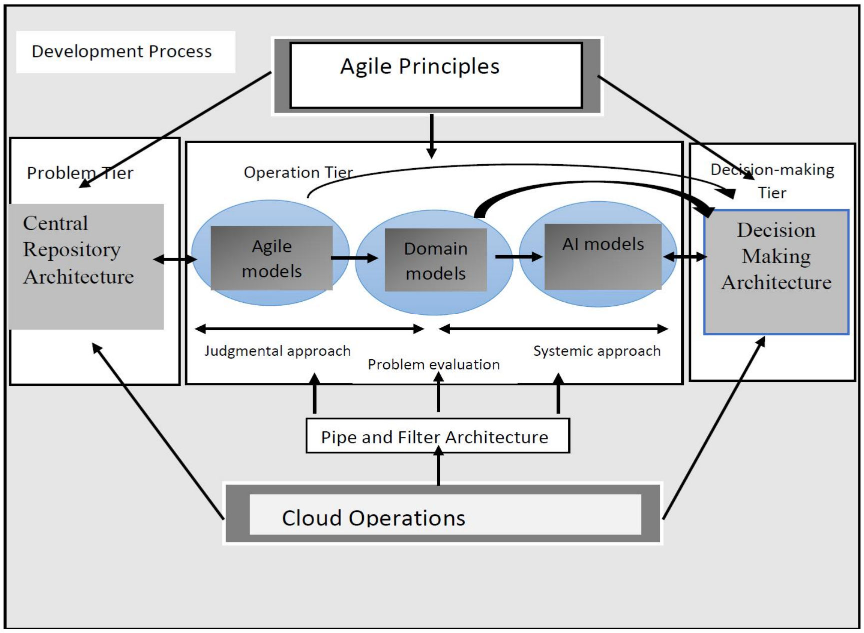

Fig. 2. Holistic approach to agile-cloud development

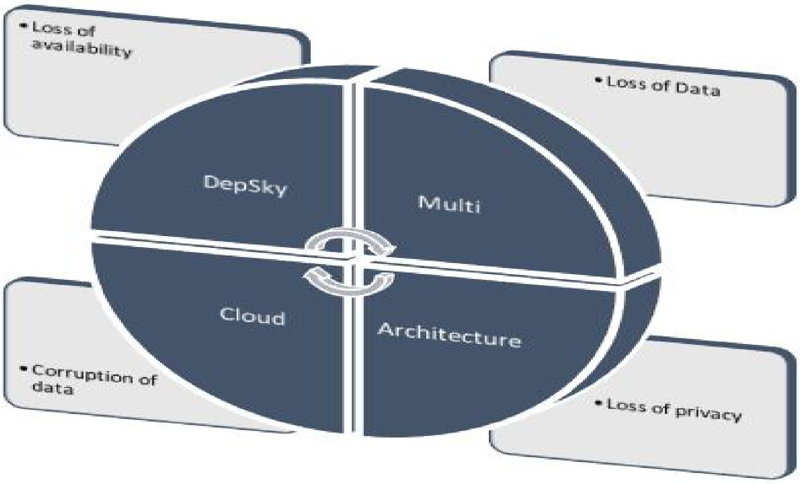

Fig. 3. DepSky - Cloud architecture

As an agile methodology requires skilled people within the team, the development team in the cloud may have an expert in testing, database, user interface to respond to any change request. With distributed agile teams, the probability of finding such highly skilled individuals increases as the teams are geographically dispersed. In a distributed setting, team managers make sure the team has required work and there is proper communication among individuals in a distributed environment as the communication happens via technology and there is a record of all communications.

In distributed agile development environment, since code reviews are, naturally, cross-team and cross-location, it is easy to respond to change quickly. In addition, due to distributed knowledge in the agile-cloud model, product support and maintenance becomes easier.

3.2 Product (Working Software) on the Cloud

Agile focuses on working software over documentation. That means agile methodology does value documentation but it streamlines it in the form of short user stories, which are easy to understand and such an environment is most favorable for distributed development teams.

Individuals working in distributed teams work differently and have different working style and degree of comfort with using electronic communication technology. It is the project manager's responsibility to assess each person's level of comfort and skills. Team leaders across the globe can have meetings before the start of the project to gain better understanding of requirements and, therefore, reducing the overall need of documentation. As agile focuses on a good quality product with limited documentation, there may be a need for highly skilled and well organized people in the team. Using distributed team concept, one can take the advantage of the skills and deliver a good product without need of detailed documentation. Despite whether the distributed group is residential or all-inclusive disseminated, certain practices are recommended. This is needed to do with limiting pointless synchronous exercises, while guaranteeing the most methodology efficiency as it focuses on customer satisfaction and quality product deliver which only needs a skilled team, which has good communication. Properly managed distributed teams create as good or even better product as collocated teams. All team members had access to the same amount of documentation using online repositories like GitHub. Centralized communication is visible to all team members using Slack and daily online standup meetings have maintained transparency across teams. This will help production of desired quality of working software in the agile-cloud model with limited documentation needed.

3.3 Customer Collaboration on the Cloud

The main idea of agile methodology is customer interaction. Constant feedback from the client on the status of the project is the key to a successful software development. Since the teams are spread across globe, customer communication becomes easier as some of the other team shares the same geographic location. This makes communication easier due to various factors like time zone, culture, language barrier. Distributed teams are very dynamic groups working on the same project, distanced by space and time. To work harmoniously, every team member, team leader, and project manager has to rely heavily on technology to communicate and collaborate. This, of course, introduces several problems in team dynamics and rapport and adversely affects the project development and consequently the product. Challenges such as these were believed to make it difficult to implement agile method in these teams. Some distributed teams, however, prove not only to be promising to work well with agile methodology but bonds the principles on which agile methodology was built.

Another approach to effective customer collaboration in the agile-cloud model is the use of distributed virtual machines provided by the client's end that are located at different geographical locations. The distributed teams should have the client's distributed virtual machines installed on their local desktops. The client will update every requirement on client specific portals that are accessible by the team members. Based on the requirements, team members can work on different modules and update the same on the portals. Meetings and discussions must be conducted on a regular basis to have a productive environment.

For Agile methodology to work efficiently the team has to interact with the client, record user stories and provide them sufficient data to propose changes or provide feedback in real-time. Distributed teams enhance the agile methodology practices as the distributed teams have to come up with effective methods to share any software development information to their clients whether they use agile or not because they cannot depend on a face-to-face meeting with the client at each phase of the development.

Some companies use webpages and in-house tools to notify clients regarding the project implementations and developments. Clients find it easier to track progress and provide feedback when it is required. Changes can be notified to the development team through the portal itself. Microsoft Lync and emails provide easy and detailed communication that both clients and the teams can come back to any time. Distributed teams are all but required to work in this way and it enhances the agile practices that the team adopted. Adoption of common practices and tools across site ensure easy sharing of knowledge. Knowledge sharing across sites can be costly. Nevertheless, dedicated open source platforms like the slack channels are used to maintain knowledge across different sites with relatively less cost. Daily meetings should happen through Webex, Zoom, One Meeting and bridge calls over VOIP. Continuous communication through IM tools like Skype will help reducing communication waiting time and speeding up the process in the agile-cloud model.

3.4 Team Infrastructure on the Cloud

Agile Methodology thrives on individuals within a team to succeed. Team members are required to communicate with one another to take ownership at every phase and work with the customer to come up with the best possible options.

In the agile-cloud development model, some software development teams are spread over multiple countries providing software solutions to several global clients using distributed teams all over the world. Some use Skype, Microsoft Lync, and Outlook for teams to communicate with each other, which is regular and consistent among team members. These software development teams have to interact with network engineering teams and clients to deploy the best possible product. These communications are scheduled beforehand through email with the agenda attached; however, the meetings are generally light and breezy, thus, creating a rapport and better relationships. In this way, team members that interact with each other on a daily even develop friendships despite being thousands of miles away. This level of rapport is developed through intensive training in professional communication, soft skills and professional writing.

In the agile-cloud model, the interactions via technology make it easy to keep track of communications. The work of the team lead becomes more efficient as the lead can track project progress, team interactions and feedback-change management based on emails. Expertise of members across the globe is shared via technology as well. In addition, since some team members do not speak same language as their first language, e-communication makes them less self-conscious about their authority in the other language.

Tools in distributed agile development are mainly used to support cross platform teams; it must include both collocated teams and distributed teams. Proper project management tools needs to be used to track task, outcomes, time, and deadlines. Product release speed may be improved by keeping the code in a releasable state by using Jenkins for continuous integration, enabling reliable and automated releases with Ansible, and monitoring production infrastructure for failures using Consul. In distributed agile development, these tools will enable teams to deploy software releases often at least every sprint if not more frequently.

One of the major problems in distributed teams is pair programming. To avoid or minimize this conflict it is important that the team is well balanced, have a similar level of programming experience, and have a good blend of skills. Distributed Teams can work effectively around the globe using various advanced tools developed over the last decade. Some of the commonly used collaboration tools are Confluence, Jira, Bitbucket, Bamboo, and Source Tree.

By using some of these interesting tools, the productivity will certainly increase more than some other documentations and provides a single platform to all the teams distributed globally to follow the agile methods without any confusions or misunderstandings.

- Version one is a project management tool specifically designed for scrum development to add and plan releases and sprints.
- Hubstaff is another great tool, which can assist in computing the billing hours.

- Ideaboardz tool can be used to share ideas anonymously so that team members can express their concerns honestly.
- JIRA is a bugs and project-tracking tool
- RALLY is a premium project management tool
- Slack is a communication tool that allows team to participate in a group.
- Blue Jeans, Microsoft SharePoint, Trello, Wiki, and GitHub.
- Upwork is a tool used by freelancers to collaborate with one another.
- Marketo is a tool for tracking billing hours.
- Mercurial, Slack and GitHub can also be used for sharing data
- WebEx for distributed communications.

These tools highlights the contributions made by every member. This starts with the assigning of tasks among the members and also prioritizing and arranging them accordingly. This helps the team analyze each other's work at the end of the project or during the development. It is very cost effective as well. We have summarized all of the tools with respect to the four main cloud operations discussed above in Table 1.

Some development companies may create a factory model for some of the projects spread across multiple countries where all the business units are divided into skill set towers. Under a tower, we would find employees of the same/similar skill set distributed globally yet belonging to the same distributed team working on project developments. It is also important to remember that, in agile methodology, processes and tools degrade over time. Therefore, all tools should be regularly assessed to ensure they are helping not hindering the software development project in the agile-cloud model.

4. RESULTS AND DISCUSSION

The success of agile project management as applied to distributed teams amounts to keeping up a few key values such as frequent communications, clear-cut goals and objectives, top-notch leadership, careful attention to details, and periodic face-to-face meetings. Considering the distributed teams as the center, it is important to employ the following suggestions in order to get the best of the development. In a distributed agile development, meetings must consist of proper quality of video where information conveyed is precise, sprint retrospectives are needed in every turn, as they might forget to consider the important concepts discussed before. Communication via Basecamp, Campfire, and other software might be productive when compared to the collated manner. In addition, with the distributed team, there must be a need for professional trust and friendship. Other solutions might be listed out as time zone consideration, raising a flag for the technical problem to higher officials, and rules explicitly conveyed to bridge the information gap. Considering all these minor changes done, the distributed team will be the most productive in agile software system development.

Distributed teams can be of different sizes based on the complexity of the project. We can definitely apply agile methodology in projects involving distributed teams by implementing the above stated materials and methods. In case of projects where resources are distributed in various countries across the same time zone, we do not have cultural differences in this case. We just need to establish proper communication by using modernized technology. In case of projects where resources are distributed across multiple time zones, try to decouple interdependencies if possible between the works of inter sites. In case of projects with cultural dependencies, cultural ambassadors' assignment will help in motivating and maintaining the trust among the resources. If you observe the pattern of production of successful projects involving distributed teams by applying Agile Model, the trend keeps on increasing across time as the technology is progressing. Internet has made it much simpler with its reliable video conferences and voice over calls. Globalization breaks barriers of cultural differences. Agile Methodology can be used successfully in case of distributed teams by following the proper methods and practices. Since one of the core components of agile methodology is communicating and collaborating with team members, from our perspective, distributed teams can use the right tools to communicate just as effectively as classical teams by utilizing the tools discussed above.

Many development companies also adapt "changing role" technique. This helps the individuals to keep fresh and does not make their work very dull. The Agile methodology is almost a perfect fit for distributed teams because one of the main principles of agile is the prioritization of

Table 1. Classification table for agile-cloud integration

		Cloud operations			
		1.Dynamic change response	2.Product development	3.Customer collaboration	4.Team operation
Mapping Of Agile Tools	Jira	✓	✓		
	Rally		✓	✓	
	Kira		✓		✓
	GitHub	✓	✓	✓	✓
	Slack, Mercurial	✓	✓	✓	✓
	ZOOM	✓	✓	✓	✓
	CISCO	✓	✓	✓	✓
	Webex		✓	✓	✓
	MS Lync, Share Points		✓	✓	✓
	IM (Skype)	✓	✓	✓	✓
	Version one	✓	✓	✓	✓
	Upwork		✓		✓
	Marketo	✓		✓	
	Blue Jeans	✓	✓	✓	✓
	Trello	✓	✓	✓	✓
	Confluence		✓		✓
	Hubstaff	✓	✓	✓	✓
	Ideaboardz BitBucket			✓	✓
	Bamboo	✓	✓	✓	✓
	SourceTree	✓	✓	✓	✓

people over processes. This forces teams to collaborate and discuss blockers and solutions. A distributed team may become separated and distant if not participating in an agile methodology and many methods in agile can be used to diminish this problem. It takes much to utilize distributed teams to improve the agile methodology. The use of distributed teams can separate a good team from a great and highly proficient team.

For the successful implementation of agile project management on distributed teams, few key points such as proficient leadership, periodic face-to-face meetings, frequent communication, minimization of time and cultural differences, decoupling of distributed systems etc. must be satisfied. There are certain practices and principles that ensure that these key points are satisfied, and communication differences are reduced. Some of these practices include establishing strategic goals, choosing the right leaders and mentors, arranging periodic in-person meetings, assigning proxies to synchronize the communications among the teams using automation and integration techniques to make sure that the teams are on the right path.

In the globalized world, above results suggest that distributed teams not only work well with agile methodology, but also it is improved through the listed practices of the distributed teams. These practices in the agile-cloud model ensure individuals who have worked extensively in distributed teams will be able adopt agile methodology much more effectively than collocated team members.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a holistic approach for distributed agile software system development. Tools and practices have been carefully picked and improved to support the successful distributed agile development in the cloud framework. The proposed model shows that distributed teams can indeed be incorporated in an agile software development environment in the cloud environment. However, there needs to be some amendments to this model to ensure that the distributed teams work properly and yield a more efficient software product in the agile-cloud model.

From the software engineering perspective, it is necessary to develop or adopt a useful agile software development process that can incorporate and deal with all the issues and challenges pertaining to cloud computing. This requires a model that allows for progression through the various stages of the development process for projects of different domains and review the findings and results. Focusing more on agile-cloud integration is an effective way to find suitable application development procedures. We attempt to provide this by establishing several connections between pillars of agile methodology and cloud computing that do not exist in the literature.

We believe that effective agile-cloud integration reduces the chance of system errors and unsatisfied stakeholder expectation. Although we have some promising preliminary results with the proposed approach, this is definitely the beginning of more general agile-cloud studies. We think this study implies a new model of agile-cloud integration and there should be an increase for research that is performed over already existing models. This will help us to develop unique, robust, and creative models in order to have an overall successful development of the software. As future enhancement, we plan to carry out a detailed case study to test the application of our proposed model and explore the strengths and the weaknesses of the new model which will design and build an agile-cloud integrated system with distributed and real time characteristics.

COMPETING INTERESTS

Author has declared that no competing interests exist.

REFERENCES

1. Moe NB, Cruzes DS, Dybå T, Engebretsen E. Coaching a global agile virtual team. IEEE 10th International Conference. 2015; 235-240.
2. Kalem S, Donko D, Boskovic D. Agile methods for cloud computing. Agile Methods for Cloud Computing, 36th International Convention on Information and Communication Technology, Electronics and Microelectronics; 2013.
3. Younas M, Ghani I, Jawawi DNA, Khan MM. A framework for agile in cloud computing environment. Journal of Internet Computing and Services. 2016;17(5):67-74.
4. Bender L, Walia GS, Fagerholm F, Pagels M, Nygard KE, Munch J. Measurement of the nontechnical skills of software professionals: An empirical investigation. In SEKE. 2014;478-483.
5. Paasivaara M, Behm B, Lessenius C, Hallikainen M. Large-scale agile transformation at Ericsson: A case study. Emperical Software Engineering. 2018;5: 2550-2596.
6. Lacher LL, Walia GS, Fagerholm F, Pagels MK, Nygard KE, Munch J. A behavior marker tool for measurement of the non-technical skills of software professionals: An empirical investigation. In SEKE. 2015; 409-414.
7. Omar M, Katuk N, Abdullah LS, Hashim NL, Romli R. Assessing personality types preferences amongst software developers: A case of Malaysia. ARPN Journal of Engineering and Applied Sciences. 2015; 10(3).
8. Ebert C, Kuhrmann M, Prikladnicki R. Global software engineering: Evolution and trends. Global Software Engineering (ICGSE) 2016 IEEE 11th International Conference on. 2016;144-153.
9. Brede Moe N, Cruzes DS, Dybå T, Storebrand EE. Coaching a global agile team. IEEE 10th International Conference on Global Software Engineering; 2015.
10. Papatheocharous E, Belk M, Nyfjord J, Germanakos J, Samaras G. Personalized continuous software engineering. In Proceedings of the 1st International Workshop on Rapid Continuous Software Engineering, ACM. 2014;57-62.
11. Gray A, Jackson Stamouli I, Tsang SL. Forming successful extreme programming teams. In Agile Conference, 2006, IEEE. 2006;10.
12. Colomo-Palacios R, Gonzalez-Carrasco I, Lopez-Cuadrado JL, Garcia-Crespo A. Resyster: A hybrid recommender system for scrum team roles based on fuzzy an drough sets. International Journal of Applied Mathematics and Computer Science. 2012;22(4):801-816.
13. Da Silva FG, Franca ACC, Suassuna M, De Sousa Mariz LM, Rossiley I, De Miranda RC, Gouveia TB, Monteiro CV, Lucena E, Cardozo ES, et al. Team building criteria in software projects: A mix-method replicated study. Information and Software Technology. 2013;55(7):1316-1340.
14. Matturro G, Fontan C, Raschetti F. Soft skills in scrum teams, a survey of the most valued to have by product owners and scrum masters. In SEKE. 2015;42-45.
15. Melo CDO, Cruzes DS, Kon F, Conradi R. Interpretative case studies on agile team productivity and management. Information

and Software Technology. 2013;55(2):412-427.

16. Kettunen P. Directing high-performing software teams: Proposal of a capability-based assessment instrument approach. In International Conference on Software Quality, Springer. 2014;229-243.
17. Wood S, Michaelides G, Thomson C. Successful extreme programming: Fidelity to the methodology or good team working? Information and Software Technology. 2013;55(4):660-672.
18. Melo C, Cruzes DS, Kon F, Conradi F. Agile team perceptions of productivity factors. In Agile Conference (AGILE), 2011, IEEE. 2011;57-66.
19. Chow T, Coa DB. A survey study of critical success factors in agile software projects. Journal of Systems and Software. 2008; 81(6):961-971.
20. Ryan S, O'connor RV. Development of a team measure for tacit knowledge in software development teams. Journal of Systems and Software. 2009;82(2):229-240.
21. Celar S, Turic M, Vickovic L. Method for personal capability assessment in agile teams using personal points. In Telecommunications Forum Telfor (TELFOR), 2014 22nd, IEEE. 2014;1134-1137.
22. Lous P, Tell P, Michelsen C, Dittrich Y, Kuhlmann M, Ebdrup A. Virtual by design: How a work environment can support agile distributed software development. In ICGSE'18: 13th IEEE/ACM International Conference on Global Software Engineering, Gothenburg, Sweden. ACM, New York, NY, USA; 2018.
23. Ebert C. Managing software products in a global context. In Proceedings of ICGSE Int. Conference on Global Software Engineering, Association for Computing Machinery, USA; 2018.
24. Domino M, Hevner A, Webb Collins R. Applying agile software development processes to global virtual teams: A study of communication modalities. In ICGSE'18: College of Business Administration, University of South Florida; 2018.
25. Zadok E. Kurma: Secure geo-distributed multi-cloud storage gateways. The 12th ACM International Conference. 2019;109-120.
DOI: 10.1145/3319647.3325830

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle3.com/review-history/49500