IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 11, November 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 11, November 2024| DOI: 10.15680/IJIRCCE.2024.1211022

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

A Case Study on Cloud Computing-Based Agile Software Development

Kotresh Naik D, Suhas K C, Madhu T S, Manoj R

Assistant Professor, Department of Computer Engineering, Channabasaveshwara Institute of Technology, Gubbi, Karnataka, India

Assistant Professor, Department of Computer Engineering, Channabasaveshwara Institute of Technology, Gubbi, Karnataka, India

UG Student, Department of Computer Engineering, Channabasaveshwara Institute of Technology, Gubbi, Karnataka, India

UG Student, Department of Computer Engineering, Channabasaveshwara Institute of Technology, Gubbi, Karnataka, India

ABSTRACT: Self-organizing teams, adaptable planning, a collaborative environment for client and team communication, short development cycles, ongoing design changes, and client feedback are all factors that contribute to the success of agile software development. Through its services, cloud computing facilitates scalability, lowers costs, and improves communication. This study evaluates a generic framework that combined Agile Development and Cloud Computing (ADCC) that was developed in a previous study. The framework is implemented using the Malaysia Research and Education Network (MyRen) cloud. To assess the framework, a case study is carried out. The ADCC framework is explained to the participants prior to the case study.

KEYWORDS: cloud-based agile tools, cloud computing, case studies, and agile development.

I. INTRODUCTION

The environment and methodology used during software development are crucial to its success. Software development procedures have changed over time to satisfy the ever-shifting demands of customers and industry.

Another factor contributing to the growth of software development processes is the adoption of new technologies, such as the emergence of big data, cloud computing, and IOTs. Software development is redirected by these stressors towards parallel processing, iterative development cycles, task prioritization [1]–[3], increased productivity, improving quality while reducing costs [4]–[6], and working together to complete tasks more quickly [7], [8].

Agile software development teams are productive, have a unique way of thinking, and absorb knowledge from more seasoned colleagues. Twelve agile principles capture the goal of agile software development [10]. Customer satisfaction, in-person communication, intimate user engagement, accommodating requirements changes, iterative design refinement, maintaining a simple and well-designed product, and frequent software delivery are the main characteristics of agile concepts. Despite the fact that agile principles highlight the value of agility in software development, their recommendations are not frequently followed because of shifting market demands and scattered development environments. Scalability, transparency [1], in-person communication [11], expert availability [3], seamless development control, the capacity to construct applications from remote locations [8], [9], and resource management [1], [7] are the difficulties encountered in agile development. Innovative ideas must be tested in a setting that adapts to changing demands. The cost of development rises when resources are made available for testing novel concepts.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Agile development is well-known for its short development cycles, ability to accommodate changes at any point during the process, frequent product releases, and robust user interaction throughout the development cycle. Cloud computing plays a key role in putting these agile development elements into practice. A paradigm for Agile Development in Cloud Computing environments (ADCC) was presented in our earlier paper [20]. The dearth of evaluation studies in the field of cloud computing and agile development is highlighted by a thorough literature study. As a result, this study uses a case study to assess the ADCC framework.

II. RELATEDWORK

A Systematic Literature Review (SLR) analysis was carried out to draw attention to the problem backdrop. Following a search of all available research materials, the primary papers were arranged as illustrated in Fig. 1 using the Wieringa Maiden classification.

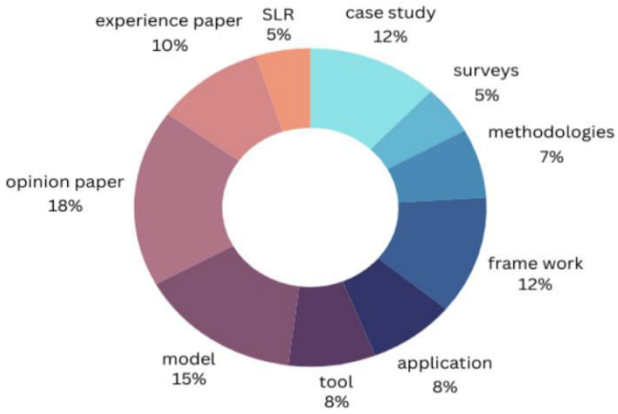

Figure 1 shows the distribution of research contributions from main studies that a study has adopted.

According to Fig. 1, approaches (7%), frameworks (12%), an application created (8%), tools (8%), and the suggested model (15%) are the main contributors to primary studies. Studies that contribute to the research type include survey papers (5%), and case studies (12%). The results of the SLR demonstrate that the field study received no input. Applications and field experiments make up 8%. A small number of studies concentrate on philosophical and validation research. Evaluation study using real-world case studies and experience reports is required in this field. Because of the quick changes in company and market behavior, development is happening at a very fast rate. The requirements for agile development tools are necessary to meet market demand [9]. Cloud computing services facilitate a wide range of software development management tools, including sprint management, velocity measurements, budgeting, analytics, time tracking and reporting, resource management, Kanban boards, and customer feedback. This aids agile development. These tools, which are mainly used for agile project management, include Jira, CloudForge, and Agilent. Bitbucket, GitHub [11], [18], [19], Code Space, Google Code, Source Forge, and Un fuddle are a few examples of source code repositories.

Source codes and their versions can be managed with the use of code repositories. The Testin Fra tool is used for network infrastructure testing in addition to software testing. Additionally, Jmeterand Cassandra tools for performance testing, JUnit for unit testing, and the Lynis tool for security testing.

III. EXPERIMENTAL SETUP AND RESEARCH METHODS

This section provides a summary of the teams involved in the experiment, the set of requirements used in the experiment, the evaluation criteria, and the research methodology and experimental setup for assessing the ADCC framework. The conceptual perspective of the ADCC framework and its implementation processes are explained by the research methodology. A distinct collection of potential ADCC environments is produced by the ADCC framework's

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

implementation. In this study, one of these potential environments is assessed. Additionally, the methodology describes the pilot test project that was employed to assess the ADCC environment. Here are the specifics:

A. FRAMEWORK OF ADCC EXECUTION

The ADCC framework [20] makes use of current methods and resources to create cloud computing environments for agile development that encourage the reusability of preexisting solutions. The ADCC framework's conceptual architecture is explained in detail in Fig. 2. Cloud services for managing agile development activities are provided by the cloud provider. Fig. 2 shows the workflow for several roles, including developers, testers, and users. The roles and responsibilities of developers, testers, project managers, and users are represented by the red, green, and blue connecting lines.

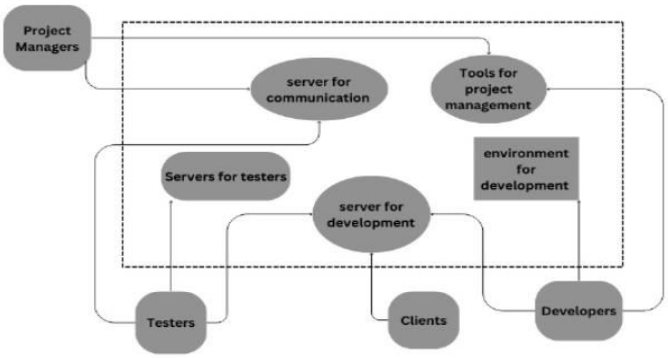

Figure 2 shows the framework's conceptual design.

The second element is the choice of cloud platforms, which are represented by the blue lines in Figure 2 and include development, testing, and deployment servers. As shown by the green lines in Figure 2, the third section of the ADCC architecture is made up of communication and collaboration tools like Google Drive, Skype, and email. The fourth section is dedicated to code management tools like GitHub. The ADCC framework's execution processes are described in Fig. 3 below. The ADCC framework can be implemented in four steps, as shown in Fig 3. Choosing agile tools is the first step, and it is determined by the development team's features and agile approach. The second phase involves choosing the cloud computing platform, which is determined by the type, size, security, and budget of the project. Step three involves choosing the code repository for versioning and code management in an agile distributed setting. The choice of communication tools is completed in step 4.

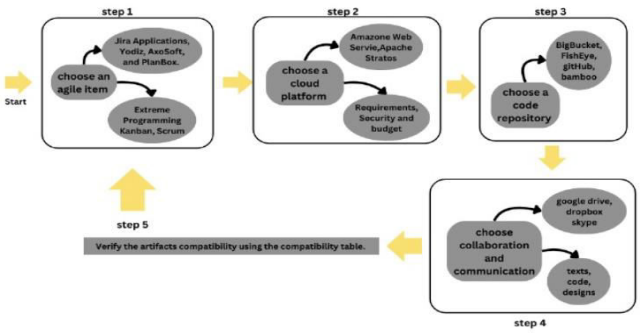

Figure 3. The process of putting the framework's conceptual design into practice

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

B. SUMMARY OF THE HOSPITAL MANAGEMENT MODE

In order to replace the current manual, paper-based approach, the Hospital Management approach (HMS) was created. To efficiently use resources while maintaining data integrity, the HMS system creates a variety of reports. Every piece of information is handled effectively and economically. Hospital administration will be automated by HMS, increasing productivity and reducing errors. Its objectives are data standardization, data consolidation, data integrity, and inconsistency reduction.

C. OVERVIEW OF THE TEAM

This case study has four teams who take part in four distinct scenarios. As seen in Figure 4, the scenarios relate to the choice of environment (like simple agile or ADCC) and mode of environment (like distributed or local). The case study involves two development teams made up of four students from a Bachelor of Computer Science class enrolled in Software Engineering II, a course offered in the fourth year of the program at the Government College University Faisalabad (GCUF), Pakistan, computer science department. A female student and three male students with a CGPA higher than 3.0 out of 4.0 make up each squad.

Teams from GCUF and UTM operate independently and locally in scenario 1, implementing a straightforward agile methodology. In scenario 2, the ADCC environment is used locally by the GCUF and UTM teams to complete the case study. Two individuals from GCUF and two from UTM make up each team in scenarios three and four, where the mode is a distributed environment. In scenario 4, the ADCC environment is used, whereas in scenario 3, standard agile development is used.

D. SUMMARY OF THE CASE STUDY'S PILOT TEST PROJECT

The teams in this case study create a software program called the "Hospital Management System" (HMS). The HMS application is being completed using the Scrum approach. There are six releases of the finished application. By carrying out agile tasks including vision definition, backlog ranking, release planning meetings, and final release, the project is divided into user stories. Two iterations of the HMS application were created using various teams and work settings.

IV. A CASE STUDY TO APPROVE THE ADCC FRAMEWORK

Table 1. Teams' performance with ADCC and agile development in a local or on-premisesetting.

	number of days					
	Average days		Team 1-UTM		Team 2-GCUF	
Phases of Development	use of ADCC	use of agile development	use of ADCC	use of agile development	use of ADCC	use of agile development
Deployment and Testing	6.5	8	6	7.5	7	8.5
Elicitation of requirements	1.5	2	1	1.5	2	2.5
Design ,coding and planning	33	36	30	33	36	39

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Table 2. performance of the team using ADCC and agile development in a distributed context.

	number of days					
	Average days		Team 1-UTM		Team 2-GCUF	
Phases of Development	use of ADCC	use of agile development	use of ADCC	use of agile development	use of ADCC	use of agile development
Deployment and Testing	6.5	9	7	10	6	8
Elicitation of requirements	1.5	3	1.5	3	1.5	3
Design ,coding and planning	33	40	31	37	35	43

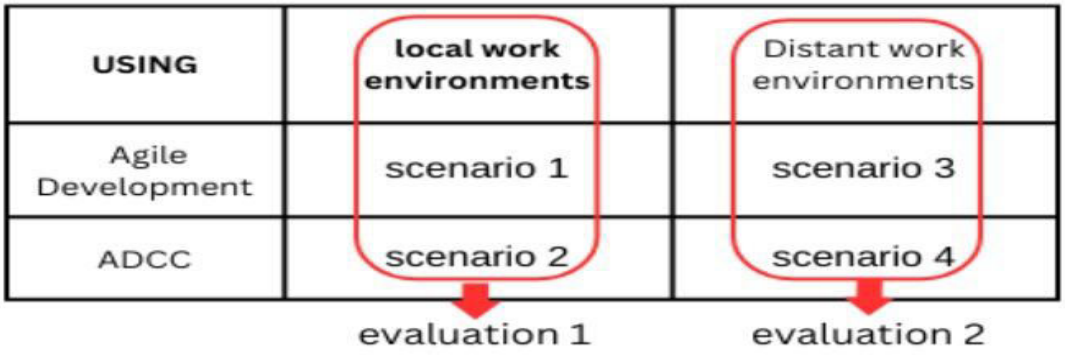

Figure 4: Scenarios for case studies.

Figure 4 illustrates the development environment (both local and distant work environments) as well as ADCC and without utilizing ADCC. Section III-C provides a description of each team taking part in the four scenarios. In every one of the scenarios, a Hospital Management System application (HMS) is created. The HMS description may be found in Section III-B. The application has been finished and released six times.In the initial assessment, the effectiveness of software development using the ADCC and simple agile environments in On-premises and local are contrasted. Both scenarios 1 and 2 are utilised.in the initial assessment. In the second assessment, the effectiveness of software development using basic agile techniques. A comparison is made between ADCC environments in different places.

A. SETTING UP APPLICATION DEVELOPMENT WITH SIMPLE AGILE METHODS

The group used PHP and MySQL to create a web-based application. They set up the setup after installing the NetBeans IDE and configuring it for the Apache HTTP server, PHP engine, and MySQL database server. Every team member set up every piece of software on their workstations. The group collaborated to create stages for resolving the issues in each iteration. How many days it takes to finish the three stages of software development—requirement elicitation, planning, design, and coding, and testing and deployment—is how the assessment is calculated.

B. ADCC Framework-Based Application Development Setup

The ADCC framework, which is run by the Malaysia Research and Education Network (MyRen) cloud, is used to create the identical application in the second and fourth scenarios. The ADCC framework's actual configuration employs the Cloud Agility tool to oversee project operations. In this experiment, the programming environment is Eclipse for PHP. GitHub is used for versioning and code management. Both email and Skype are used for team communication.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

The prototype is created, tested, and revised by the developers while the application is being developed. Cloud virtual machines are used to carry out all of the tasks in concurrently. A development environment is set up on the Selenium test server. Environment as outlined in Section III-A above. After the first iteration is finished, it is distributed on the deployment server, and the product owner (teacher) and the end user, the user can access it, verify it, and add further specifications. If necessary. Every iteration is finished in the same manner and the module is merged after all releases are finished. The instructor receives the finished solution. The outcomes in the form of the overall completion time for each step for Every deliverable is taken into account.

V. RESULTS OF CASE STUDY

Assessment 1 and Assessment 2 are the two subparts of the case study. Evaluation 1 compares the outcomes of scenario 1 with those of scenario 2. Three stages in the development of software are contrasted in scenarios 1 and 2, including the use of basic agile environment and working locally or on-premises with the ADCC environment. Performance is measured in the following phases: i) requirement gathering, ii) design and coding planning, and iii) deployment and testing. The situation involves utilizing only utilizing the ADCC framework and agile software development. The total number of days is used to measure the amount of workutilized to complete the artefact. It is determined how average each team is across the three stages of development. Figure 5 illustrates the variation in the number of days required to complete agile development tasks in the two scenarios. Figure 5 illustrates how ADCC reduces development time in a local setting when compared to a basic agile setup.

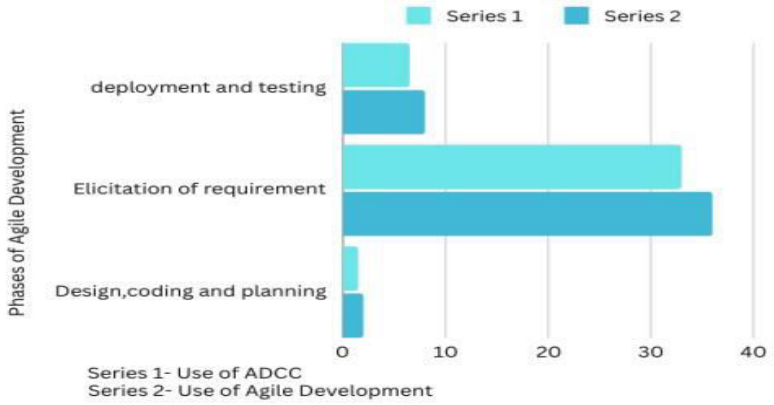

Figure 5 shows a comparison of agile stages in a local development environment with and without the use of ADCC.

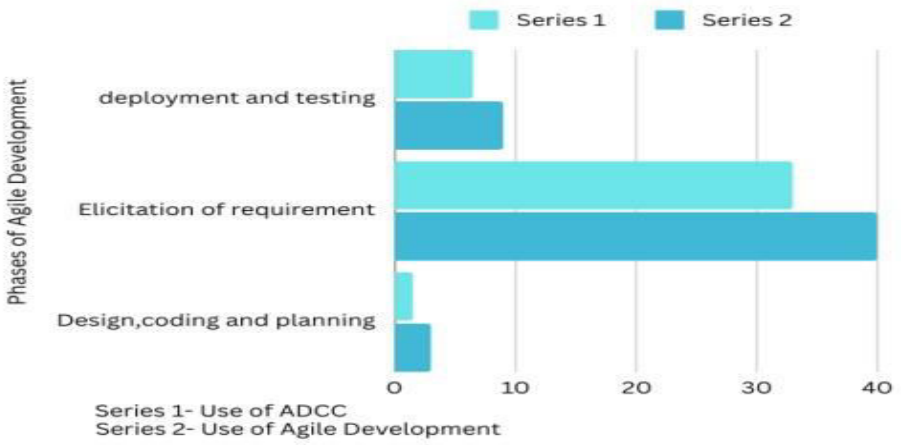

Figure 6 Agile phases in a distributed development environment with and without ADCC are shown.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

To reduce the impact of teams' competency, Table 2 computes the average of two composite teams. As demonstrated in Figure 6, the ADCC environment outperforms a basic agile environment in terms of performance. The ADCC environment has improved because teams and users may communicate without any delays and because hardware and computer resources are available without requiring patch, update, or installation setups. Delays-free testing and deployment are offered by the ADCC environment. A basic agile software development process in a dispersed setting may be hampered by a communication breakdown between development teams and other stakeholders, which may ultimately impact the requirement gathering procedure. However, the ADCC structure facilitates communication and teamwork, which aids in the phases of requirement gathering and planning. As a result, the elicitation process utilizing solely agile software development takes three days, while the ADCC framework, as seen in Fig. 6, takes one and a half days.

Agile approaches confirm requirements through user involvement until the project's completion; nevertheless, as Fig. 7 illustrates, communication and cooperation issues cause the requirement elicitation process to be delayed and task completion times to increase. It might be challenging for team members and clients to share design, concepts, code, and prototypes in a dispersed environment. Delays in collaboration and communication result in longer design and code completion times, as illustrated in Fig. 7. Additionally, sharing prototypes and development progress through client deliverables and test reports encounters challenges in a remote setting and eventually lengthens the time needed to complete a work. The delay during software development phases rises in a distributed environment. The increased time in various agile development phases as a result of the dispersed environment is depicted in Figure 7.

VI. CONCLUSION

A case study is used to assess the ADCC framework. This case study aims to outline the challenges encountered during agile software development and demonstrate how the ADCC framework is used to address these challenges. The team has resources during an agile software development process, but unlike in actual software development, these are not connected. It gets expanded if specialized resources are set up, and additional network personnel are needed to maintain the system. Conversely, the ADCC architecture (cloud computing) offers an integrated environment. Every developer and team member has the impression that they are working in an on-premise setting. Every system functions similarly to every other system.

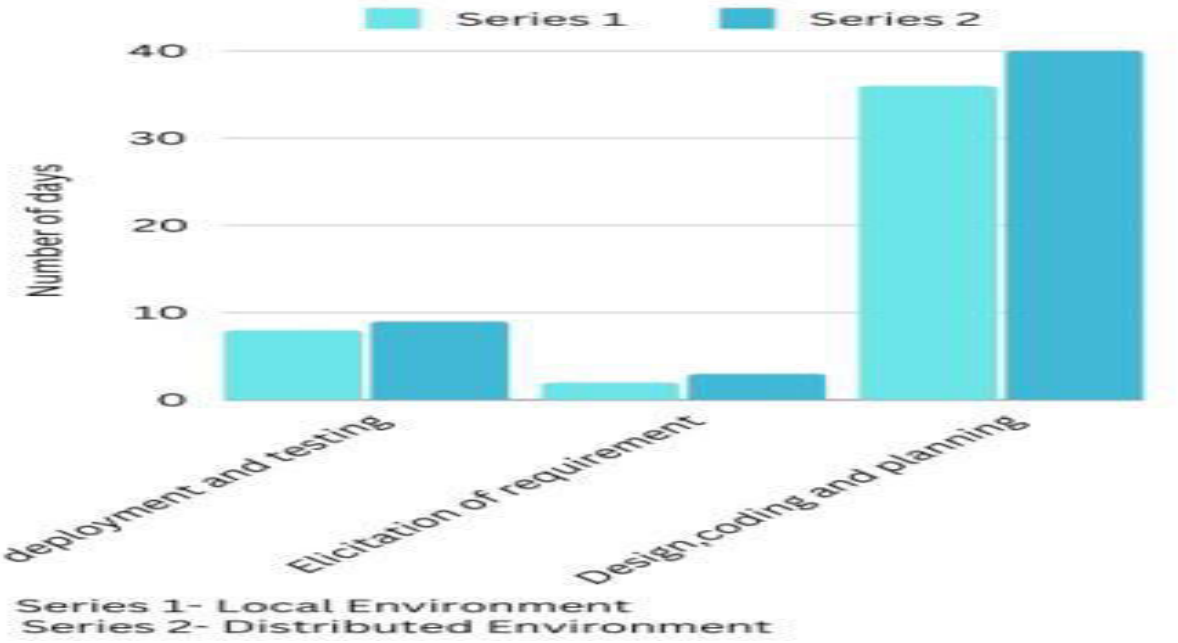

Figure 7: The influence of dispersed and local environments on agile development.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Although the ADCC framework is a general solution for agile development in cloud computing settings, it can be used to various software engineering approaches in the future by choosing different environments and tool sets.

VII. VALIDITY THREATS

The framework offers a variety of options for platform and tool selection. The validity of the case study is threatened by the following factors. When the same project is created by many teams using and without the ADCC framework, performance bias could arise. Different teams may have varying levels of competency. Teams are informed about the ADCC's experimental setting in the case study, therefore attitudes during environmental education may affect the outcomes. Communication between team members from the same class or university is uncontrollable in a distributed setting. Additionally, there is bias because the requirements employed in the case study were not shared by both apps. Due to these restrictions, the requirements utilized in case studies for application development are not available, hence the current study cannot be compared to previous research.

ACKNOWLEDGMENT

The Government College University Faisalabad (GCUF), Pakistan, provided case study participants, and the Malaysia Research and Education Network (MyRen) cloud provided resources for this study. The authors also acknowledge University Tecnology Malaysia (UTM) for Transdisciplinary Research Grant 06G23.

REFERENCES

[1] A. Tuli, N. Hasteer, M. Sharma, and A. Bansal, ‘‘Empirical investigationof agile software development: Cloud perspective,’’ SIGSOFT Softw. Eng.Notes, vol. 39, no. 4, pp. 1–6, 2014.
[2] F. Almudarra and B. Qureshi, ‘‘Issues in adopting agile developmentprinciples for mobile cloud computing applications,’’ Procedia Comput.Sci., vol. 52, pp. 1133–1140, 2015.
[3] A. Nazir, A. Raana, and M. F. Khan, ‘‘Cloud computing ensemblesagile development methodologies for successful project development,’’Int. J. Mod. Educ. Comput. Sci., vol. 11, pp. 28–35, Nov. 2013.
[4] N. Jain and S. Dubey, ‘‘Agile development methodology with cloud computing,’’ Int. J. Eng. Comput. Sci., vol. 3, no. 4, pp. 5373–5378, 2014.
[5] I. Inayat, S. S. Salim, and Z. M. Kasirun, ‘‘Agile-based software productdevelopment using cloud computing services: Findings from a case study,’’Sci. Int. (Lahore), pp. 1065–1069, 2013.
[6] S. Kalem, D. Donko, and D. Boskovic, ‘‘Agile methods for cloud computing,’’ in Proc. 36th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO), May 2013, pp. 1079–1083.
[7] A. Dumbre, S. P. Senthil, and S. S. Ghag, ‘‘Practising agile softwaredevelopment on the windows azure platform,’’ Infosys, Bengaluru, India,White Paper, 2011.
[8] A. Sever, ‘‘Modeling distributed agile software development utilizingcloud computing: A holistic framework,’’ Current J. Appl. Sci. Technol.,vol. 35, no. 6, pp. 1–12, 2019.
[9] A. Alliance. (Feb. 25, 2001). Agile Manifesto. [Online]. Available:http://www.agilemanifesto.org
[10] M. R. J. Qureshi and I. Sayid, ‘‘Scheme of global scrum managementsoftware,’’ Int. J. Inf. Eng. Electron. Bus., pp. 1–7, Mar. 2015.
[11] S. Singh and I. Chana, ‘‘Introducing agility in cloud based software development through ASD,’’ Int. J. u- e-Service, Sci. Technol., vol. 6, no. 5,pp. 191–202, Oct. 2013
[12] R. Shriver. (2012). Agile Cloud Development: The Future ofSoftware. Accessed: Feb. 20, 2016. [Online]. Available: http://www.virtualizationpractice.com/agile-cloud-development-the-future-ofsoftware-16226/
[13] N. Rathod and A. Surve, ‘‘Test orchestration a framework for continuousintegration and continuous deployment,’’ in Proc. Int. Conf. Pervas. Comput. (ICPC), Jan. 2015, pp. 1–5.
[14] B. Portelli. (2010). The Beauty of Agile in The Cloud. Accessed: Sep. 2017.Available: [Online]. Available: https://www.agileconnection.com/article/beauty-agile-cloud
[15] H. Benfenatki, C. F. D. Silva, A.-N. Benharkat, and P. Ghodous, ‘‘Cloud–based business applications development methodology,’’ in Proc. IEEE23rd Int. WETICE Conf., Jun. 2014, pp. 275–280.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

[16] T. G. P. Mell. (2011). The NIST Definition of Cloud Computing.[Online]. Available: http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[17] S. A. Butt, M. I. Tariq, T. Jamal, A. Ali, J. L. D. Martinez, andE. De-La-Hoz-Franco, ''Predictive variables for agile development merging cloud computing services,'' IEEE Access, vol. 7, pp. 99273–99282,2019.

[18] T. Haig-Smith and M. Tanner, ''Cloud computing as an enabler of agileglobal software development,'' Issues Informing Sci. Inf. Technol., vol. 13,pp. 121–144, Mar. 2016.

[19] W.-T. Tsai, W. Wu, and M. N. Huhns, ''Cloud–based software crowdsourcing,'' IEEE Internet Comput., vol. 18, no. 3, pp. 78–83, May 2014.

[20] M. Younas, I. Ghani, D. N. Jawawi, and M. M. Khan, ''A framework foragile development in cloud computing environment,'' J. Internet Comput.Services, vol. 17, no. 5, pp. 67–74, 2016.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details