A STUDY OF AGILE SOFTWARE DEVELOPMENT IN CLOUD ENVIRONMENTS USING SCRUM AND KANBAN PRACTICES

Ashwani Kumar[1] and Geetanjali Amarawat [2]

[1,2] Department of IT & Computer Science, Swaminarayan University, Kalol, Gujarat, India,

ABSTRACT

This Agile software development thrives on self-organizing teams, adaptable planning, a collaborative atmosphere fostering communication among clients and team members, brief development cycles, ongoing design enhancements, and the continual delivery and feedback from clients. Cloud computing reduces costs, facilitates scalability, and improves communication through its services. This study evaluates a generic framework that integrates Agile Development and Cloud Computing (ADCC), as presented in a prior study. The Malaysia Research and Education Network (MyRen) cloud is employed to execute the framework. A study is performed to assess the framework. Prior to executing the study, the participants are instructed on the ADCC framework. The case study results indicate that the efficacy of agile methodologies is enhanced by the implementation of the ADCC framework. The enhancement is evaluated in relation to local and distributed agile development settings.

Keywords

Cloud Computing, AGILE Software Development, Scrum, Kanban

1. INTRODUCTION

Software development requires the capacity to adapt, and a plan-driven approach isn't as effective as agile software development. One manifestation of this adaptability is the ability to alter requirements at any stage of the software development life cycle [1]. One component of agile approach is a self-organizing development team. This model encourages developers and clients to work together closely [2]. Several Agile approaches, when applied to projects, can help achieve this objective. More effective formal and informal communication is expected to result from the adoption of agile methodologies during software development [3]. Open office spaces, daily meetings, product backlogs, sprint planning, iteration retrospectives, and iteration planning meetings are a few strategies that influence communication [4,5] Iteration reviews are yet another illustration. Practices that are likely to effect team communication were one of the main topics of research attention for Pikkarainen and colleagues [6,7,8]. Unfortunately, there is a dearth of information on how different approaches impact the overall budget and time required to finish the project. Only a certain amount of supporting evidence for this effect. The importance of studying the various methods and processes used in Agile projects is highlighted by all of this. This study aims to deconstruct Agile software development projects by analyzing their many approaches and how they interact with one another to get different results. With the Agile methodology that has dominated software development for the past two decades, the Scrum framework has emerged as the most popular approach to the Agile methodology. There is a huge majority of software development teams who utilize a project management technique that can be defined as some kind of "Agile." This is despite the fact that the extremely fragmented nature of the business makes it

Bibhu Dash et al: ICFCSAI-2025
pp. 105-114, 2026. IJCI – 2026 DOI:10.5121/ijci.2026.150111

challenging to obtain meaningful figures. In the most current assessment on the state of agile, which was published on the 15th, it was discovered that 94% of the firms that the 1382 international respondents worked for had implemented Agile. Sixty-six percent of those that utilized a Scrum framework to implement a variant of the Agile Over the last 20 years, software development has been heavily influenced by the Agile technique. The Scrum framework, under this methodology, has become the de facto standard. Most software development teams use a method of project management that might be characterized as "Agile." This is so even though it is difficult to get useful figures due to the highly dispersed nature of the industry. In the most current assessment on the state of agile, which was published on the 15th, it was discovered that 94% of the firms that the 1382 international respondents worked for had implemented Agile. Sixty-six percent of those that utilized a Scrum framework to implement a variant of the Agile methodology at the team level were among the majority of those who did so

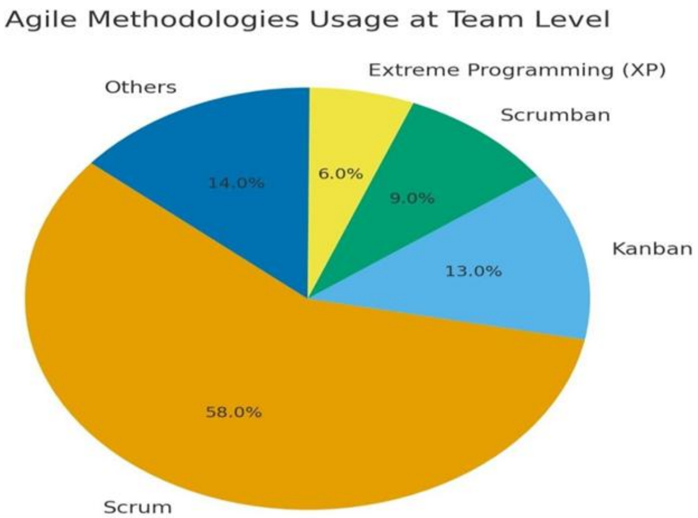

Figure: methodology at team level.

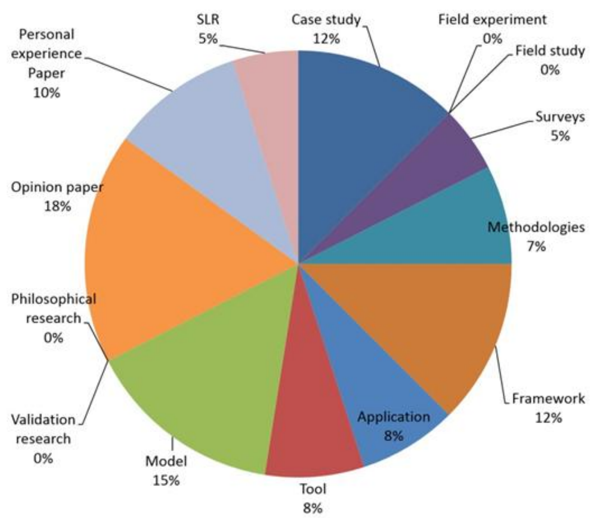

Figure: Distribution of primary studies research contributions adopted by a study

In 2021, the integration of Agile and Scrum emerged as the most frequently utilized software development methodology. The importance of the Scrum framework in contemporary software development is shown by the fact that an additional fifteen percent of respondents adhered to variations closely related to Scrum. This highlights the importance of the Scrum framework with greater clarity. Nine percent of respondents indicated that their team employed ScrumBan, a hybrid approach that amalgamates elements of both Scrum and Kanban. An extra 6% pertains to Scrum/XP, which integrates elements from both Scrum and Extreme Programming (XP). If you could chose only one way to understand the software development process and project management, the Scrum framework would be your choice. Consequently, Scrum serves as an exemplary starting point for exploring the theory underlying Agile software development, together with the concrete procedures and actions delineated by the many frameworks encompassed within Agile methodologies.

2. LITERATURE REVIEW

In accordance with this manifesto, a total of twelve (12) core principles of agile development were also proposed. The principles of agile software development provide more context and support for why agility is so important when creating software. However, due to the terms' implementation in a distributed environment and the market's fast changing requirements, the agile standards are not rigorously enforced.

Scalability, transparency, face-to-face contact, availability of experts, seamless control of development, capacity to design applications from remote places, and resource management are some of the obstacles that need to be solved. It is now feasible to improve software development processes by fusing cloud computing environments and agile approaches with the help of Advanced Techniques. Through time-boxed iterations and cross-functional cooperation, agile techniques like Scrum, Kanban, and Feature Driven Development (FDD) enable iterative, incremental software delivery.

According to research by Prabu Ravichandran (2024) [1], Analysis on Agile Software Development Using Cloud Computing based on Agile Methodology and Scrum Framework. *ISSN: 2321-8169,2024.* The authors examined the adoption of agile software development approaches in cloud computing environments. The empirical investigation showed that ADCC framework is evaluated through the use of a case study. The goals of this case study are to (1) describe the challenges of agile software development and (2) demonstrate the use of the ADCC framework to these problems According to research by BT, Shreya and Cholli, Nagaraj G., Agile Methodologies with Cloud Computing (ADCC), 2024[2] The authors investigated the implementation of agile software development methodologies in cloud computing contexts. The empirical study shown that the ADCC framework is assessed using a case study. The Agile Development and Cloud Computing (ADCC) has resulted in a drastic change in software development procedures. Agile development methodologies prioritize iterative development, continuous feedback, and client requirements.

According to research by Hashmi, S. I., Baik, J., & Grover, LK(2018)[3], integrating DevOps practices with agile methodologies in the cloud improves software delivery performance. The authors examined the adoption of agile software development approaches in cloud computing environments. The empirical investigation showed that emphasizing collaborative behaviors and ongoing feedback loops was the key to this model's effectiveness. Other research has investigated ways to improve the accuracy of agile approaches in cloud environments by taking extra considerations like security and compliance into account

2.1. Contemporary Agile Frameworks in SaaS

A range of Agile frameworks have gained traction in SaaS development. SAFe supports large-scale SaaS implementations, while Scrum is ideal for compact teams (Leffingwell, 2018). Despite this, top-performing SaaS firms frequently develop hybrid models, combining elements from various frameworks. Spotify's engineering framework, as presented by Kniberg and Ivarsson (2012), pioneered the squad-tribe-guild model, now a SaaS industry staple. This system enables team autonomy while preserving alignment through cultural and operational consistency.

3. Rapid Integration and Deployment Framework

Contemporary SaaS development is deeply rooted in the use of Continuous Integration and Continuous Delivery (CI/CD) pipelines to support fast and dependable software rollouts. In contrast to traditional models—where updates might roll out every few months—SaaS solutions frequently deploy changes several times a day (Fowler, 2013). This demands robust automation and testing frameworks to preserve quality while sustaining a high cadence of releases.

Innovators like Amazon and Netflix have been at the forefront of CI/CD adoption, leveraging advanced strategies such as automated canary releases and blue-green deployments (Humble & Farley, 2010). These methodologies help reduce deployment-related risks and enable swift reversals when problems arise.

The combination of agile with cloud computing is advantageous for the creation of distributed applications, the sharing of data, the prioritization of tasks, transparency, and the construction of infrastructure. Specifically, this is due to the fact that cloud computing is having an increasing influence on the ecosystem of agile software development. When referring to agile methodologies, the phrase "ecosystem" is used to describe a system or a collection of pieces that are interconnected. Examples of ecosystems include the development environment, the interconnectedness of teams, and the entire system. One of the many advantages of cloud computing is the improvement in performance it brings to agile software development, as well as the reduction of expenses and the capacity to scale. These are just some of the advantages. In addition, the agile methodology utilizes social technologies that are hosted in the cloud in order to facilitate the implementation of the communication practice that is a component of the agile methodology. On the other hand, when it comes to academic research, there is not a lot of study that examines such discoveries, shares research gaps, and analyzes relevant concerns in both of these fields together. In their own right, agile and cloud computing contain sufficient amounts of content that has been peer-reviewed, but not both.

Consequently, the focus of this review paper is on the research that has been conducted on both the agile approach and cloud computing.

4. Research Methodology

4.1. Execution of the ADCC framework

The ADCC framework utilizes existing processes and technology to guarantee the creation of environments that are ideal for agile development in cloud computing. These environments foster the reusability of solutions that are already in existence and actively advocate their reuse. In Figure 2, the conceptual architecture of the ADCC framework is broken down in greater detail.

When it comes to managing agile development activities, the cloud provider offers assistance in cloud services administration. Figure 2 is a representation of the workflow used by several roles, including users, testers, and developers. The red, green, and blue connecting lines show that users, testers, project managers, and developers all have separate responsibilities and roles. The red lines in Figure 2 represent one of the initial components of the ADCC framework: agile software project management. The second part is picking a cloud platform, which comprises servers for testing, development, and deployment. The blue lines seen in Figure 2 indicate these cloud platforms. Figure 2 shows the third component of the ADCC architecture, which includes mediums for communication and cooperation such as Google Drive, Skype, and email. The fourth component, on the other hand, is made up of code management tools like GitHub and various other similar platforms. The procedures that are involved in the implementation of the ADCC framework are delineated in the following paragraphs, as seen in Figure2 These are just some of the advantages. In addition, the agile methodology utilizes social technologies that are hosted in the cloud in order to facilitate the implementation of the communication practice that is a component of the agile methodology. On the other hand, when it comes to academic research, there is not a lot of study that examines such discoveries, shares research gaps, and analyzes relevant concerns in both of these fields together. In their own right, agile and cloud computing contain sufficient amounts of content that has been peer-reviewed, but not both. Consequently, the focus of this review paper is on the research that has been conducted on both the agile approach and cloud computing.

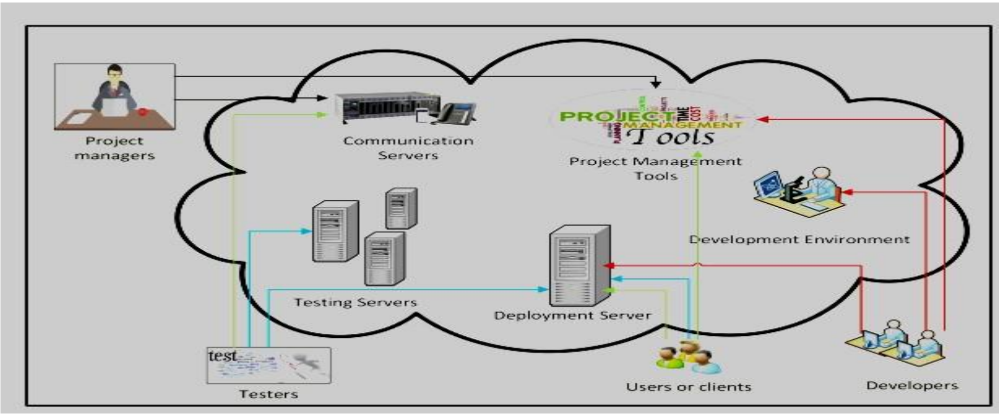

Figure 2. Designing the framework conceptually.

The second part is picking a cloud platform, which comprises servers for testing, development, and deployment. The blue lines seen in Figure 2 indicate these cloud platforms. Figure 2 shows the third component of the ADCC architecture, which includes mediums for communication and cooperation such as Google Drive, Skype, and email. The fourth component, on the other hand, is made up of code management tools like GitHub and various other similar platforms. The procedures that are involved in the implementation of the ADCC framework are delineated in the following paragraphs, as seen in Figure 3.

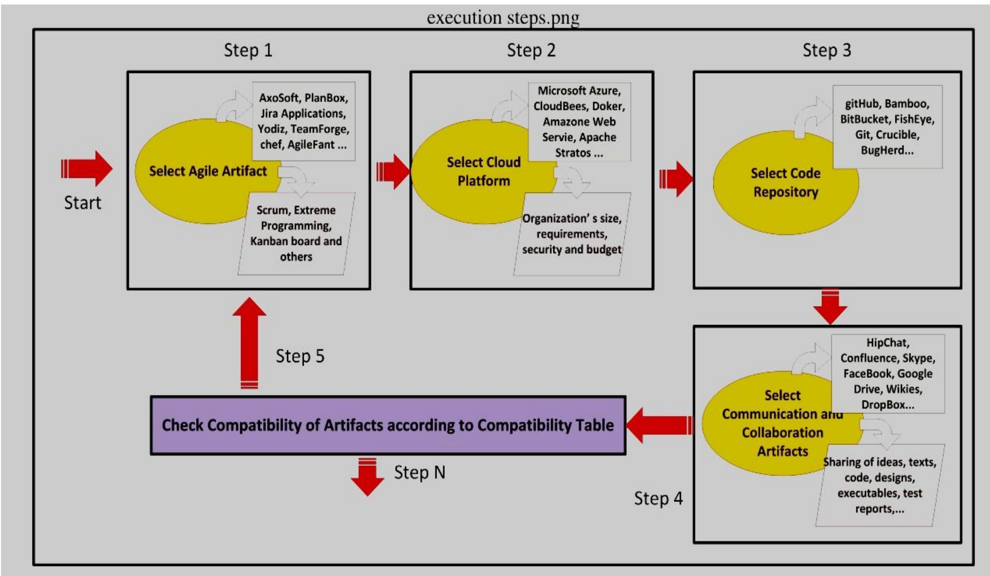

Figure 3. Systematic approach to the implementation of the conceptual design of the framework.

In order to construct an ADCC environment, the research uses the approach that is outlined in Figure 3, the specifics of which are laid forth in the following lines. The ADCC framework can be implemented in four steps, as shown in Figure 3. The first thing that has to be done is to choose agile tools, which are determined by the agile methodology and the features that are utilized by the development team. The selection of the cloud computing platform is the second phase, and it is determined by the nature of the project together with its budget, size, and level of security. When working in an agile distributed setting, the third stage involves selecting the code repository for the purpose of managing code and versioning instances. The choice of communication tools is made at the fourth step of the process. To select all of these tools in just four steps, it is important to check the compatibility of the tools, which is done in the fifth phase of the procedure. This is done in order to ensure that the tools are compatible with one another. Concerns regarding compatibility can be addressed and ultimately resolved by the utilization of the comparability table in Study. In the event that the choice that was made did not achieve the desired results, it is important to repeat steps one through four until all of the compatibility concerns have been resolved.

5. RESULTS AND STUDY

Table1. Agile vs ADCC with Cloud-Native CI/CD Stacks

Phase	Agile Practices (Traditional Agile)	ADCC Practices (Agile + Cloud Computing)	Cloud-Native CI/CD Tools Supporting ADCC
Requirement Elicitation & Planning	Sprint planning, user stories, product backlog in tools like Jira, Trello	Cloud-hosted Agile boards with integrated pipelines for traceability	Azure Boards + Azure DevOps, GitLab Issues & Boards, Jira Cloud + GitHub/GitLab

Design & Coding	Local development, Git version control, peer reviews	Cloud-based collaborative coding environments, instant code sync, cloud IDEs	GitHub Codespaces, AWS Cloud9, GitLab Cloud IDE
Build & Integration (CI)	Local builds, Jenkins servers, manual integration	Fully automated cloud-native CI pipelines with scalable runners	AWS CodeBuild, Google Cloud Build, Azure Pipelines, GitLab CI
Testing (Unit, Integration, Security)	Manual/automated tests, run locally or on-prem test servers	Cloud-based automated test suites, parallel test execution, integrated security scans	Selenium Grid on Cloud, GitHub Actions + SAST/DAST, AWS Device Farm, GitLab Security Tests
Deployment (CD)	Manual deployments, limited automation	Continuous delivery with cloud orchestration, GitOpsdriven rollouts	AWS CodeDeploy, Google Cloud Deploy, Argo CD, Jenkins X, Spinnaker
Monitoring & Feedback	Sprint reviews, burndown charts, velocity tracking	Cloud-native observability, automated feedback loops, integrated metrics	AWS CloudWatch, GCP Cloud Monitoring, Azure Monitor, Prometheus + Grafana
Collaboration & Communication	Standups, retrospectives, Slack/MS Teams	Cloud-native collaboration with real-time dashboards, low-latency remote sync	Slack + Jira Cloud, Microsoft Teams + Azure DevOps, Mattermost + GitLab

Within the context of the local environment, a group of personnel drawn from both GCUF and UTM worked together in a manner that was distinct and independent. Table 2 shows the outcomes for a GCUF group and a UTM group in terms of the total number of days needed to finish the development phases. The participants in both sets were first-year college students.

Table 2. The effectiveness of the team in a distributed setting through the application of agile development and ADCC.

Development Phases	No. of Days					
	Team 1-UTM-GCUF		Team 2-UTM-GCUF		Average Days	
	Using Agile Development	Using ADCC	Using Agile Development	Using ADCC	Using Agile Development	Using ADCC
Requirement elicitation	3	1.5	3	1.5	3	1.5
Planning, design and coding	37	31	43	35	40	33
Testing and deployment	10	7	8	6	9	6.5

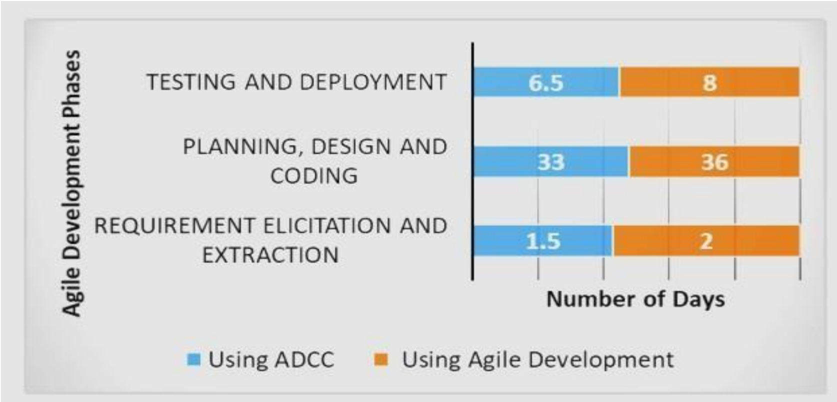

Figure4. Comparative analysis of agile phases in a local development environment, both with and without the utilization of ADCC.

The level of expertise possessed by each team is almost identical for both of the scenarios. In addition, the average of both teams is computed in order to moderate the influence of the competency. Each and every student is capable of doing the tasks of creating, coding, and testing. The role of the product owner and end user can be assumed by the class instructor and his aide in an emergency. Finding the mean is as simple as averaging the results across all teams in relation to the three phases of development. Figure 4 shows that when comparing the two scenarios, the amount of days spent on completing agile development activities is significantly different. When compared to a straightforward agile environment, Figure 6 demonstrates that the use of ADCC results in better utilization of development time in a local setting.

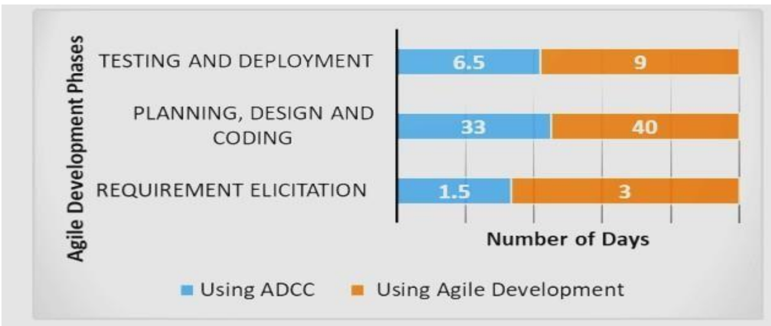

Figure5. Agile development life cycle (LCL) comparisons using and without ADCC methods in a distributed setting.

Table 2 shows the results of averaging two composite teams in an effort to lessen the effect of team competences on the organization's overall performance. Figure 5 shows that compared to a basic agile environment, the ADCC environment yields superior performance. Thanks to the improved communication between teams and users, the ADCC environment has seen a remarkable improvement. Another factor is the efficient provision of hardware and computing resources, which is achieved without any concerns about patch, update, or installation configurations.

6. CONCLUSION

The ADCC framework is assessed by conducting a case study. The objectives of this case study are to (1) detail the obstacles associated with agile software development and (2) illustrate the application of the ADCC framework to address these issues. Although the team in an agile software development process has access to resources, they are not necessarily connected in the same way as they would be in a more traditional software development process. The system the will become expansive if dedicated resources are arranged, requiring employment of additional network professionals to manage it. Conversely, the ADCC framework, which is a form of cloud computing, provides a networked environment. The developers and other team members are under the impression that they are working in an environment that is situated on the premises. Everything functions as if it were a single system. The ADCC framework offers advantages over a straightforward agile development environment when the number of days required to complete software development-related procedures is.

REFERENCES

[1] Dönmez, D.; Grote, G.; Brusoni, S. Routine interdependencies as a source of stability and flexibility. A study of agile software development teams. Inf. Organ. 2016, 26, 63–83. [Google Scholar] [CrossRef]

[2] Prabu Ravichandran, Analysis on Agile Software Development Using Cloud Computing based on Agile Methodology and Scrum Framework International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169,2024

[3] BT, Shreya and Cholli, Nagaraj G., Agile Methodologies with Cloud Computing (ADCC) (August 01, 2024). at SSRN: https://ssrn.com/abstract=4940644 or http:// Available dx.doi.org/10.2139/ssrn.4940644

[4] Ozkan, N.; Gök, M.Ş.; Köse, B.Ö. Towards a better understanding of agile mindset by using principles of agile methods. In Proceedings of the 2020 15th Conference on Computer Science and Information Systems (FedCSIS), Sofia, Bulgaria, 6–9 September 2020; pp. 721 730. [Google Scholar]

[5] Dorairaj, S.; Noble, J.; Malik, P. Effective communication in distributed agile software development Teams. In International Conference on Agile Software Development; Springer: Berlin/Heidelberg, Germany, 2011; pp. 102–116. [Google Scholar]

[6] Baham, C.; Hirschheim, R. Issues, challenges, and a proposed theoretical core of agile software development research. Inf. Syst. J. 2021, 32, 103–129

[7] Pikkarainen, M.; Haikara, J.; Salo, O.; Abrahamsson, P.; Still, J. The impact of agile practices on communication in software development. Empir. Softw. Eng. 2008, 13, 303– 337.

[8] Arcos-Medina, G.; Mauricio, D. Identifying Factors Influencing on Agile Practices for Software Development. J. Inf. Organ. Sci. 2020, 44, 1–31.

[9] Tam, C.; Moura, E.J.D.C.; Oliveira, T.; Varajão, J. The factors influencing the success of on-going agile software development projects. Int. J. Proj. Manag. 2020, 38, 165– 176.

[10] Taylor, R. Interpretation of the Correlation Coefficient: A Basic Review. J. Diagn. Med. Sonogr. 1990, 6, 35–39. [Google Scholar] [CrossRef]

[11] Jeong, H.; Mason, S.P.; Barabasi, A.; Oltvai, Z.N. Lethality and centrality in protein networks. Nature 2001, 411, 41– 42. [Google Scholar] [CrossRef] [PubMed] [Green Version]

[12] Liu, D.; Zhai, Z. An Empirical Study of Agile Planning Critical Success Factors. Master's Thesis, Blekinge Institute of Technology, Karlskrona, Sweden, June 2017. [Google Scholar]
[13] Moniruzzaman, A.; Hossain, D.S.A. Comparative Study on Agile software development methodologies. arXiv 2013, arXiv:1307.3356. [Google Scholar]
[14] Cao, L.; Ramesh, B. Agile Requirements Engineering Practices: An Empirical Study. IEEE Softw. 2008, 25, 60– 67. [Google Scholar] [CrossRef]
[15] Ghimire, D.; Charters, S.; Gibbs, S. Scaling agile software development approach in government organization in New Zealand. In Proceedings of the 3rd International Conference on Software Engineering and Information Management, Sydney, NSW, Australia, 12–15 January 2020. [Google Scholar] [CrossRef]
[16] Salameh, H. What, when, why, and how? A comparison between agile project management and traditional project management methods. Int. J. Bus. Manag. Rev. 2014, 2, 52– 74.
[17] Nadella, Geeta Sandeep. Validating the Overall Impact of IS on Educators in U.S. High Schools Using IS-Impact Model – A Quantitative PLS-SEM Approach, DAI-A 85/7(E), Dissertation Abstracts International, Ann Arbor, ISBN 9798381388480, 189, (2023).
[18] Gonaygunta, Hari, Factors Influencing the Adoption of Machine Learning Algorithms to Detect Cyber Threats in the Banking Industry, DAI-A 85/7(E), Dissertation Abstracts International, Ann Arbor, United States, ISBN 9798381387865, 142, 2023.
[19] Ramya Manikyam, J. Todd McDonald, William R. Mahoney, Todd R. Andel, and Samuel H. Russ. 2016.Comparing the effectiveness of commercial obfuscators against MATE attacks. In Proceedings of the 6th Workshop on Software Security, Protection, and Reverse Engineering (SSPREW'16)
[20] R. Manikyam. 2019.Program protection using software based hardware abstraction.Ph.D. Dissertation.University of South Alabama.
[21] GPB GRADXS, N RAO, Behaviour Based Credit Card Fraud Detection Design And Analysis By Using Deep Stacked Autoencoder Based Harris Grey Wolf (Hgw) Method, Scandinavian Journal of Information Systems 35 (1), 1-8.
[22] R Pulimamidi, GP Buddha, Applications of Artificial Intelligence Based Technologies in The Healthcare Industry, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 4513-4519.
[23] R Pulimamidi, GP Buddha, AI-Enabled Health Systems: Transforming Personalized Medicine And Wellness, Tuijin Jishu/Journal of Propulsion Technology 44 (3), 4520-4526.
[24] GP Buddha, SP Kumar, CMR Reddy, Electronic system for authorization and use of cross-linked resource instruments, US Patent App. 17/203,879.
[25] Hari Gonaygunta (2023) Machine Learning Algorithms for Detection of Cyber Threats using Logistic Regression, 10.47893/ijssan.2023.1229.
[26] Hari Gonaygunta, Pawankumar Sharma, (2021) Role of AI in product management automation and effectiveness, https://doi.org/10.2139/ssrn.4637857
[27] Sri Charan Yarlagadda, Role of Artificial Intelligence, Automation, and Machine Learning in Sustainable Plastics Packaging markets: Progress, Trends, and Directions, International Journal on Recent and Innovation Trends in Computing and Communication, Vol:11, Issue 9s, Pages: 818–828, 2023.
[28] Sri Charan Yarlagadda, The Use of Artificial Intelligence and Machine Learning in Creating a Roadmap Towards a Circular Economy for Plastics, International Journal on Recent and Innovation Trends in Computing and Communication, Vol:11, Issue 9s, Pages: 829-836, 2023.
[29] Amol Kulkarni, Amazon Athena Serverless Architecture and Troubleshooting, International Journal of Computer Trends and Technology, Vol, 71, issue, 5, pages 57-61, 2023.
[30] Amazon Redshift Performance Tuning and Optimization,International Journal of Computer Trends and Technology, vol, 71, issue, 2, pages, 40-44, 2023.
[31] B. Nagaraj, A. Kalaivani, S. B. R, S. Akila, H. K. intelligence in STEM Higher Education: A Critical review," International Research Journal of Multidisciplinary Technovation, pp. 10.54392/irjmt2351. 1–19, Aug. 2023, doi:
[32] D. Sivabalaselvamani, K. Nanthini, Bharath Kumar Nagaraj, K. H. Gokul Kannan, K. Hariharan, M. Mallingeshwaran, Healthcare Monitoring and Analysis Using ThingSpeak IoT Platform: Capturing and Analyzing Sensor Data for Enhanced Patient Care, IGI Global eEditorial Discovery, Pages: 25, 2024. DOI: 10.4018/979- 8-3693-1694-8.ch008.
[33] MUHAMMAD YOUNAS,2019 et all, Agile Software Development Using Cloud Computing: A Case Study Sachdev, and S. K. N, "The Emerging Role of Artificia