International Journal of Multidisciplinary Research and Growth Evaluation.

Agile software development: Impacts and challenges within distributed teams

Khalid KA Abdullah [1*], M Rizwan Qureshi [2]
[1-2] Department of Information Technology, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

* Corresponding Author: **Khalid KA Abdullah**

Article Info

ISSN (online): 2582-7138
Volume: 04
Issue: 01
January-February 2023
Received: 19-01-2023;
Accepted: 10-02-2023
Page No: 572-579

Abstract
In agile software development, rapid tasks, which are called the story and collaboration between team members are a necessary requirement. To achieve this, teams need to adhere steps such as active meetings, face-to-face meetings, continuous meetings and cooperation between the work team. However, Separation through geographical distance, time zones, and culture has an influence on the ability of members of a distributed team to work with each other as one team in distributed software development. We determined that trust among members of a distributed team is critical for bridging spatial, behavioral, and cultural barriers so that they can collaborate as one team. In this paper, we discuss the obstacles and challenges of using Agile in a distributed team's environment and recommend a proposed solution of four steps and procedures to solve these issues. The validation of the proposed solution is done by using a survey and the results indicate that communication efficiency has positive effects and found encouraging effects on functionality, quality, and on-budget completion.

DOI: https://doi.org/10.54660/.IJMRGE.2023.4.1.572-579

Keywords: Agile software development, Distributed teams, Global software engineering, Software engineering

1. Introduction

Until recently, the teams working on the projects were located in a unified and common workplace and one of the reasons for the success was their presence under one roof. Where all team members are able to communicate and interact in a strong way to achieve the best quality in software development. But now, with the rapid development in the field of software development and the urgent need for renewable technologies, working within geographically distributed teams has become an important requirement. This has become a prominent feature of working in multiple geographical locations to make the most of the experiences of the various members of the distributed work teams. Also, one of the advantages of distributed teams was maximizing the profit of companies by reducing development costs, delivering tasks faster, having a larger base of employees, and finally increasing customer loyalty by providing quick feedback to them at all times. Here the agile software methodology came and appeared strongly because its unique concept in project management provides a more flexible and more efficient way to deliver products and projects and integrate them into the market faster. The agile software approach works with a clear concept by dividing the software life cycle into smaller story cycles, each dedicated to managing specific software functions in order to be developed in a focused manner and then successfully delivered. Also, one of the most important advantages of this methodology is its flexibility in receiving any new updates and requirements during the project implementation and design phase, which provides greater flexibility in dealing with the project owner [1]. But with all these advantages, there must be some flaws and drawbacks, which, with their neglect, may lead to the failure of projects. Among the most important of these failings are the multiple cultures of the team members, the difference in languages among them, and the difference in the organizational culture in dealing, which causes major problems in communicating between them. Also, the different working hours in the different

time zones of the members of the work represent another obstacle in this methodology, which leads to fewer meetings due to the lack of a suitable time for everyone [2].

Also, all these different factors create some difficulty in building trust among the members of the distributed team, and the organizational culture needs more effort in order to put the team within a unified framework of work. Therefore, practices must be developed that solve these various problems and increase the strength of bonding and communication between the work team in order to reduce the defects in the use of this methodology within the distributed teams.

The remainder of this paper is structured as follows: Section 2 discusses related work. Section 3 defines the problem described in this paper. The proposed solution was depicted in Section 4. Section 5 validates the proposed solution. Finally, section 6 describes the paper's Conclusion.

2. Related Works

Distributed agile software development is considered one of the best models that address the concerns of software application markets, which have ever-evolving requirements due to their large user base that must be reached within a short time to market [3]. Furthermore, such agile tasks necessitate a task allocation procedure, which has been examined in the aspect of distributed agile software development. To facilitate the quality evaluation of task-member assignments during and after the project completion, a two-phase task allocation methodology is developed.

Shrivastava and Rathod [4] Explain how software companies for cost and time benefits have adopted distributed agile development (DAD) approach. However, this comes with some issues, and causes significant challenges considering the contradicting nature of the agile methodology, and distributed development. "That challenges have been explored, but there has not been any comprehensive work on risks and challenges when these approaches are combined in practice. Results of implementing the partial risk management framework in three multinational IT consultancy companies shows the number of risk factors identified in each project eliminated".

Younas *et al*. [5] came up with a significant solution for the increased cost of agile software development scalability. They came to take advantage of cloud computing to reduce the cost of software development by providing better infrastructure and architecture for pay-as-you-go software only. And that is by activating the ADCC framework. They conclude that agile development is an optimal solution within the cloud computing environment.

Hess *et al*. [6] discussed Requirements complexity in software projects can challenge the cognitive capabilities of stakeholders involved in these projects. Moreover, this challenge is often difficult and time-consuming for project stakeholders to use and analyze requirements documents. Motivated by the pervasive use of agile development approaches and existing challenges faced by requirements engineering in agile projects, the research aimed at investigating whether agile teams could also benefit from the empirical results regarding information needs gained in previous research.

Nowadays the business world is characterized by complexity. Accordingly, service providers are facing the challenge and obstacle to reduce time to market while delivering innovative products that customer love. Agile software development (ASD) promises benefits such as on-time delivery and customer satisfaction [7]. The study investigates what approaches exist to involve stakeholders in the process, which methodologies are commonly used to present the user perspective, and how requirements management is being carried out.

In [8] The communication between onshore and offshore sites in distributed teams is addressed as one of the primary challenges that face the software requirements process. Reducing repetition time at the offshore site requires more efficient communication with the onshore site. By integrating industrial practice and academic techniques, an approach was suggested and encouraged to bridge the gap between industries and research in distributed agile development. A useful and informative software tool that facilitates the management of requirement changes in distributed agile development accompanies the approach. The supporting tool is tested and the results are promising.

Fernandez-Diego *et al*. [9]. Discussed the need for effective models to calculate effort in the project planning process. Agile Software Development (ASD) has often been presented as a kind of successful alternative to traditional approaches and methods of trying to predict. Extensive planning, codified processes and rigorous reuse are the key features for the efficient development of software. Therefore, afford a review to estimate effort in agile software development through an updated systematic literature review (SLR).

In software development projects, the scrum framework, a collection of agile principles and practices for self-organizing cross-functional teams, is used [10]. The study investigates the extent to which key scrum principles and tools, due to their potential positive influence on team dynamics and efficiency levels, can contribute to and assist in task management and coordination in research procedures. The lessons learned from this case study indicate that the scrum framework may not be the optimal agile approach for distributed research management and need to reconsider its suitability as the best agile approach for distributed team management.

In agile software development, team and organizational features of agile development projects are where much of the existing agile development material is concentrated. However, less emphasis has been paid to the coordinating techniques utilized in agile initiatives. Zaitsev *et al*. [11]. Conducted a case study of an agile development project to address and discover agile software development problems involving customer and vendor organizations. The study improved awareness of the role of harmonization artifacts in agile projects gives perceptions into the artifacts' informative potential, and identifies some of the pitfalls that might occur from misunderstanding object use.

As the agile method grows in popularity, so too does the demand for agile adoption guidance. Several agile maturity models (AMMs) have been presented in the past few years to provide guidance in agile adoption. [12] Investigate the order of agile practice introduction mentioned in AMMs, the order of introducing agile practices in the industry, and the similarities and differences between them.

Choras *et al*. [13] intend to report on practical experience through the use of metrics involved in the software development process in order to assist Small and Medium Enterprises (SMEs) in the development of software using an Agile methodology. They followed the principles of action

research in a small Polish software development company. Using a pilot case, we developed and carried out a study protocol tailored to the needs of the company. Moreover, as the result, a catalog of agile development process metrics is practically validated.

Agile approaches were designed for use in single or small development teams and projects. However, these methods and techniques may also be used in Large-Scale Agile Development (LSAD) teams and projects because of their usefulness in the same field. Abrar *et al*. [14] identified from a management perspective that there are a total of 21 motivators for the large-scale adoption of agile approaches. These 21 motivators assessed the discovered motivators based on numerous variables such as continents and digital libraries.

Gandomani *et al*. [15] study attempted to discover answers to ambiguities and problems concerning the role of agile project management, the role of the project manager, and associated challenges by conducting a Systematic Literature Review. The results indicate that in software teams without a project manager, pre-defined roles in Agile methodologies are frequently accountable for project management functions.

Software development projects developed by a single agile team working in the same location have proven less commonplace in the latest years, and numerous agile software development environments have become geographically dispersed. Šmite *et al*. [16] attempt to understand if cultural barriers exist in distributed projects where Indian engineers cooperate with more empowered Swedish management, and if so, how to overcome them. As a result, twelve cultural obstacles were discovered, six of which were classified as roadblocks to agile software development approaches, and a report was developed detailing how these obstacles manifested themselves in five DevOps teams.

Over the years, global software development (GSD) has taken over the co-located software development model because of increased quality and decreased cost. Organizations are increasingly adopting new methodologies for global software development, including agile in the GSD industry, which has both benefits and drawbacks.

However, software development teams fail to address situational demands, resulting in the discovery of incompatible assumptions and architecture-level rework delays in the development process. In [17] they undertake a systematic literature review (SLR) to determine the situational aspects that software development teams must consider before developing a software product.

Table 1: Limitations and restrictions of the related work

Title Of The Paper	Limitations
"A Quantitative Framework for Task Allocation in Distributed Agile Software Development" [3]	The suggested structure for models necessitates a pool of tasks and team members per location. The effectiveness of the strategy relies on suitable pooling. The solution space is constrained by the defined pool's boundaries.
"A risk management framework for distributed agile projects" [4]	Due to time constraints and financial restrictions, the majority of the gathered data was from Indian firms. This may influence the amount of generalization and, as a result, the validity and reliability.
"Agile Software Development Using Cloud Computing: A Case Study" [5]	The applications of the study did not share the same requirements as the case study, which is a bias. Due to these constraints, the current study cannot be compared to previous studies due to the absence of requirements utilized in the case of studies for application development.
"Agile Requirements Engineering: A systematic literature review" [7]	By Selection criteria which picked. They concentrated on articles published in English. As a result, relevant research authored in languages other than English may have been excluded due to the exclusion criteria utilized.
"An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review" [9]	The main limitation of this work (concerns the possible biases introduced in the selection process. they attempted to avoid this bias by defining the search strategy for primary studies in accordance with Kitchenham's guidelines for performing SLRs in Software Engineering)
"Adapting the scrum framework for agile project management in science: case study of a distributed research initiative" [10]	The limitations were (in terms of a shared understanding and coherent application of the scrum framework, when compared to similar experiences in the use of agile methods in research projects).
"Coordination artifacts in Agile Software Development" [11]	The focus of the study was constrained to the harmonization between a client and a developer because this is one of the most mutual relationships in Agile software development that necessitates effective organization
"Measuring and Improving Agile Processes in a Small-Size Software Development Company" [13]	"They were not able to get a random sample of participants in the pilot project. In addition, they defined an evaluation protocol in advance, which included a specific description of the planned procedure and the order of using the materials."
"Overcoming cultural barriers to being agile in distributed teams" [16]	"Generalizability of the conclusions drawn from results are, of course, limited to the studied context. The empirical results concentrating on behavioral differences and scenarios may be appropriate and interesting for other Scandinavian businesses doing business in India. However, this does not apply to relationships between other countries or Scandinavians working with offshoring destinations other than India."
"Towards Taxonomical-Based Situational Model to Improve the Quality of Agile Distributed Teams" [17]	Among reactions gathered through online questionnaires, only 18 were complete. This might be described as a limited sample. Accordingly, to organize the examined situational obstacles into five situational aspects, an informal technique was used. This might affect the validity of the situational-barrier categorization process.

3. Problem Definition

In distributed agile software development, we need a lot to cooperate with many teams, some of which are in the same city, some in the same country, and others from outside the country, and this has a positive impact on the tasks distributed among these teams, such as reducing the burden and sometimes the cost and time required to complete the project. However, this distributed team works may have a negative

impact on the project, as communication is not good between them due to the difference in time. Language, and even some cultural and civilizational considerations between different countries, and many studies on the effectiveness and quality of work within the distributed teams found that outcomes were very brief and limited, as we find in the study [9], which was one of its limitations that it was focused on (papers written in the English language. Therefore, there might be relevant studies written in languages other than English, which were excluded because of the applied exclusion criteria due to the language difference between the teams and the difficulty of obtaining papers in other languages that were translated clearly and properly).

We also see in the study [5], which mentioned that the "current study does not compare with the existing studies due to unavailability of requirements used in case studies for application development". From here, we see that the study of the requirements was facing difficulty because some of the requirements were not sufficiently available. so the comparison and measurement of the extent of improvement in the distributed agile software development were poor because of the different case studies, which is also It is difficult to reach them uniformly within the different teams in the project.

In addition, the distributed teams in distributed agile software development suffer from the time and cost limits allocated to the project as well. As mentioned in [4], the information was limited to a narrow scope within the state of India because of what was mentioned in the study. The majority of data was acquired from Indian firms due to time and budget restrictions. This would have an influence on the level of generalization and, as a result, the validity and reliability.

Another heavy burden in distributed agile software development is Different geographic regions and the same thing is repeated in one of the solutions presented in the framework developed in [3] where the limitations in the tasks were limited by the time and spatial limits set by the scope of the tasks within the project.

However, in contrast to the negatives, problems and challenges in working within distributed teams in distributed agile software development, it is not always this bad influence. Challenges and effects in this field are great but distributed teams, on the other hand, continue to be one of the most essential approaches in distributed agile software development. However, the distributed tasks have clarity of requirements, having a unified language for dealing between teams and an appropriate time are the most important requirements for working with successful teams within the agile distributed software development.

4. The Proposed Solution

Many benefits and paybacks were offered by distributed teams. However, if improperly handled, managing distributed teams could be a challenging task. Knowing this, there should be a solution to these problems and the suggested solution consist of four phases:

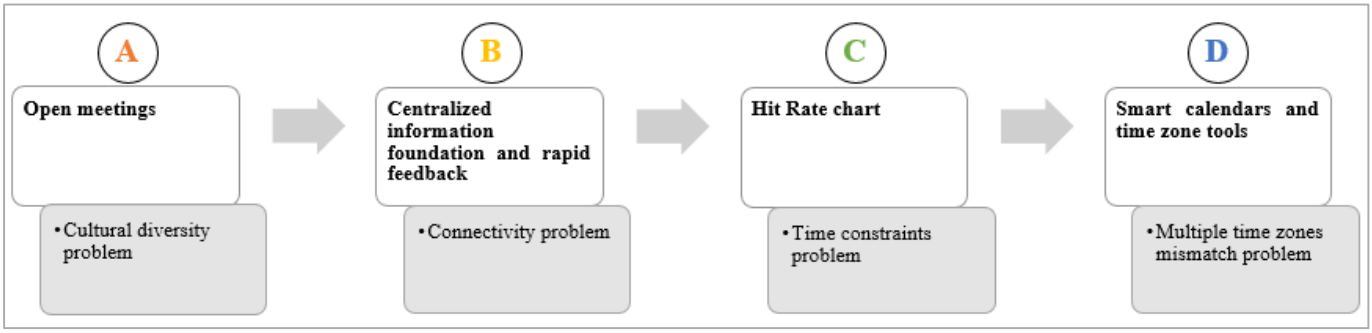

Fig 1: The proposed solution phases to minimize the challenges within distributed teams

A. Open Meetings to Blend Distributed Teams into Smooth Cultural Disparity

While cultural diversity is an important factor to take into account, it also creates complications with standard communication language and culture across the distributed teams. Culture is an intangible factor that affects productivity but is not directly measured. As previously stated in [7], the project's work is hampered by differences in languages, beliefs, and ideas.

We discovered that in routine business meetings, the team's most fluent speakers always take the lead in thinking. It might be tough to make remote employees feel like they are a part of the present community because they do not work in the same physical location every day.

To overcome this obstacle, team leaders must think and persuade people who have low faith in themselves to speak up in meetings. Another option to close these gaps is to have written documentation of the project activity. Team members who are more introverted are also permitted to share their thoughts, and having clear language devoid of ambiguous accents contributes to the efficacy of teamwork.

Another prevalent approach is to hold open meetings that could be conducted remotely. In the first ten minutes of meetings, chat about what is going on outside the workplace. Checking how team members work is a useful technique to establish trust and deeper relationships. People who are hesitant to speak out in a large gathering might use chat to express their questions and opinions in an inconspicuous manner. Executives also learned that when management was not in the same room to coordinate responses, participants reacted differently, leading to increased active engagement in open meetings in the form of give-and-take between directors and shareholders.

B. Centralized Information Foundation and Rapid Feedback to Improve Connectivity in Teams

In distributed teams, business culture and employee relationships suffer. It is difficult to share information and immerse workers in the environment because they do not have any interaction with one another. As noted problem in [5], we may discover that connectivity with teams is inadequate due to the dearth of clear requirements employed

in software development case studies.

Building a centralized information foundation is one of the suggested potential solutions. Which will keep all necessary tools, legal documentation, and workflows on hand to avoid remote workers from starting extended conversations and waiting for assistance and guidance. Rapid feedback is another feasible option. We can use surveys and feedback forms to determine how the team is engaged and motivated. Furthermore, it is imperative to complete recommended tasks after obtaining feedback so that improves team involvement.

C. Hit Rate Chart to Overcome Time Constraints

Distributed teams suffer from a lack of time, time constraints, and difficulties in completing tasks on time, as indicated in [4]. As a result of these constraints, there is a lack of achieving the desired results.

A Deliverable Hit Rate chart is a tool that can help with this problem. The chart illustrates the progress of completed tasks across time in comparison to a target. It presents a graphical representation of whether the task completion frequency is on schedule to deliver the software program on time. This is a simplified form of an agile burn-down chart. It is simpler to use since it minimizes the need for task estimation. It uses data to construct a time-averaged trend of deliveries.

D. Smart Calendars and Time Zone Tool to Match Due to Multiple Time Zones

Challenge found in [3] is Time mismatch due to multiple time zones within geographically distributed teams typically having people in different time zones. This implies a time correspondence for effective meetings, especially for scattered teams far from each other in several time zones. Furthermore, there is also a psychological effect of the time zone mismatch, which affects team members' comfort ability and the whole project's efficiency negatively.

This problem might be solved by utilizing services like Google Calendar and Outlook, which allow each member to schedule meetings automatically. The Every Time Zone is an additional useful tool in this instance, which assistance the calendar and meeting hours by determining each team member's availability. Always scheduling a few minutes of informal discussion at the first minutes of the meeting is feasible to relax and establish confidence before the official half of the meetings to cover the gap in times mismatch and comfort the distributed teams.

5. Validation

Validation is an important procedure in scientific research for determining the effectiveness of the suggested solution, models, or approaches, as well as evaluating the contents of research papers. The validation of the suggested solution is done in this paper using a survey method. The objective in utilizing such a method is to get the desired tester sample scope and collect the data out of multiple geographical locations disregarding time scale or expense restrictions. The survey is being utilized as a research technique and tool to put the suggested solution to the test.

Several disadvantages are linked with adopting a survey approach, such as the fact that a substantial percentage of contributors did not fill out the survey correctly and completely, and the majority the of given information supplied is ambiguous. The results are based on a sample of forty participant's size.

A bar chart has been used to illustrate the outcomes and results of the survey. The obtained sample size includes forty professionals working in multiple firms from various geographic areas. Participants in the sample work in a variety of occupations, including project manager, creative director, front-end developer, back-end developer, and designer. For the questionnaires, we developed an online google form, and we spread the survey via numerous social media channels such as "WhatsApp", "Telegram", "Twitter", and "Facebook". The contributors answered the electronic survey, which was divided into five numbers, using a "Likert" scale. Number 1 indicates strongly disagree, number 2 indicates disagree, number 3 indicates nominal, number 4 indicates agree, and number 5 indicates strongly agree. The survey has 22 questions separated into four main goals:

- **Goal 1:** Determine the importance of holding Open meetings to blend distributed teams into smooth cultural disparity.
- **Goal 2:** To improve connectivity in teams by using a centralized information foundation and rapid feedback.
- **Goal 3:** Determine the usability of the Hit Rate chart to overcome time constraints.
- **Goal 4:** Minimize the unmatched multiple time zones by using Smart calendars and the time zone tool.

A. Accumulative Outcomes of Questionnaire for Goal 1

The detailed accumulative outcomes for goal 1 are presented in table 2. Figure 2 illustrates that 12.08% of contributors are "strongly agreed" that the suggested solution is appropriate and effective, while 45.83% are "agreed", 20.83% are in a neutral state, 15.83% have "disagreed", and 5.42% of the contributors "strongly disagreed".

Table 2: Detailed accumulative outcomes for goal 1

Q. No	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	5	22	8	3	2
2	2	3	8	22	5
3	2	5	10	17	6
4	2	4	14	16	4
5	0	2	4	28	6
6	2	2	6	24	6
Total	13	38	50	110	29
Avg.	5.42	15.83	20.83	45.83	12.08

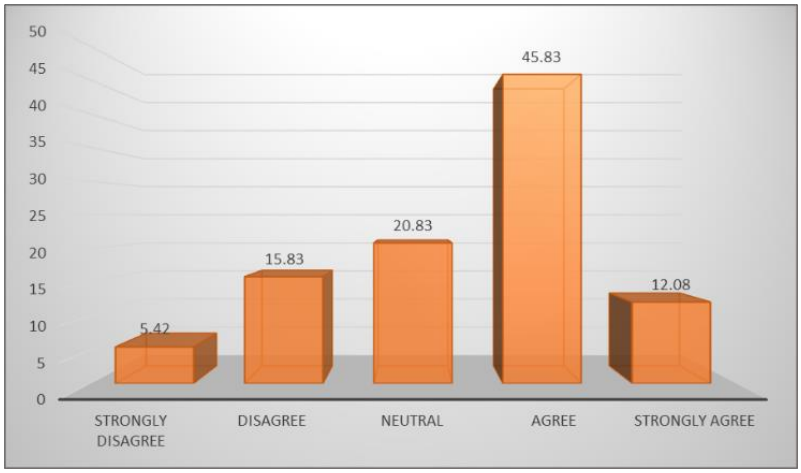

Fig 2: Accumulative outcomes for goal 1 chart

B. Accumulative Outcomes of Questionnaire for Goal 2

The detailed accumulative outcomes for goal 2 are presented in table 3. Figure 3 illustrates that 22% of contributors are "strongly agreed" that the suggested solution is appropriate and effective, while 54% are "agreed", 16% are in a neutral state, 5% have "disagreed", and 3% of the contributors "strongly disagreed".

Table 3: Detailed accumulative outcomes for goal 2

Q. No	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	0	2	6	18	14
2	2	2	6	26	4
3	0	4	8	20	8
4	4	2	10	16	8
5	0	0	2	28	10
Total	6	10	32	108	44
Avg.	3	5	16	54	22

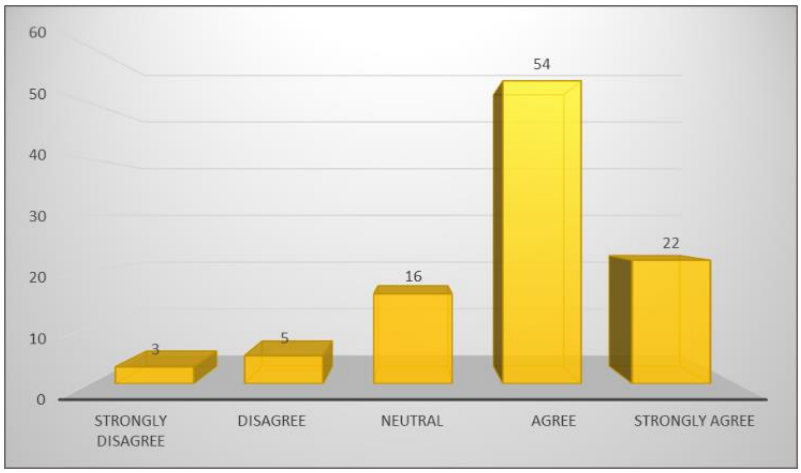

Fig 3: Accumulative outcomes for goal 2 chart

C. Accumulative Outcomes of Questionnaire for Goal 3

The detailed accumulative outcomes for goal 3 are presented in table 4. Figure 4 illustrates that 25% of contributors are "strongly agreed" that the suggested solution is appropriate and effective, while 53% are "agreed", 12% are in a neutral state, 6% have "disagreed", and 4% of the contributors "strongly disagreed".

Table 4: Detailed accumulative outcomes for goal 3

Q. No	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	2	4	6	22	6
2	0	2	2	28	8
3	0	2	6	18	14
4	4	2	6	24	4
5	2	2	4	14	18
Total	8	12	24	106	50
Avg.	4	6	12	53	25

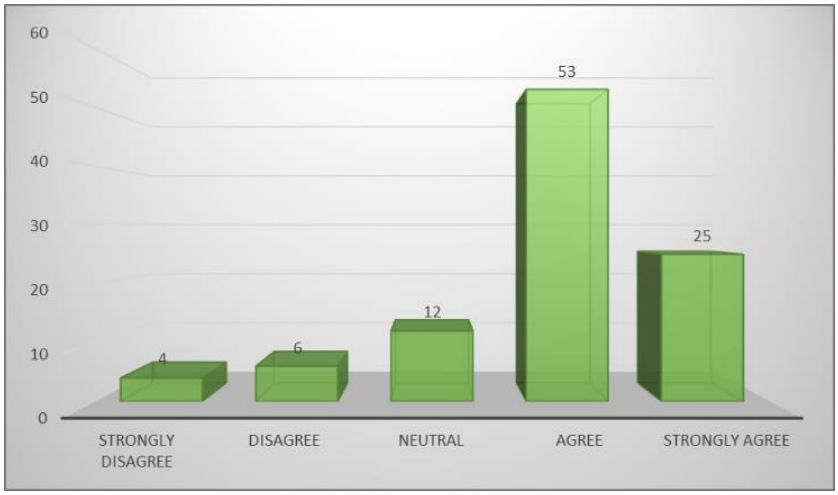

Fig 4: Accumulative outcomes for goal 3 chart

D. Accumulative Outcomes of Questionnaire for Goal 4

The detailed accumulative outcomes for goal 4 are presented in table 5. Figure 5 illustrates that 38.33% of contributors are "strongly agreed" that the suggested solution is appropriate and effective, while 31.67% are "agreed", 15.83% are in a neutral state, 7.50% have "disagreed", and 6.67% of the contributors "strongly disagreed".

Table 5: Detailed accumulative outcomes for goal 4

Q. No	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1	2	0	4	10	24
2	0	2	0	18	20
3	6	4	14	10	6
4	2	2	2	12	22
5	4	6	6	12	12
6	2	4	12	14	8
Total	16	18	38	76	92
Avg.	6.67	7.50	15.83	31.67	38.33

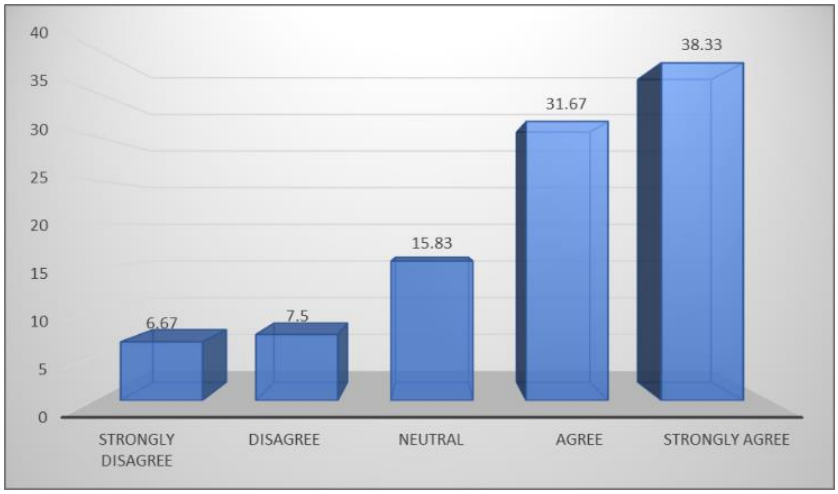

Fig 5: Accumulative outcomes for goal 4 chart

6. Conclusions

Projects managed by distributed teams continue to be challenging, and managing distributed projects under agile software development is far more complicated, However, these challenges are exacerbated when dispersed collaboration collides with cultural boundaries. This paper discusses and discovers obstacles ingrained in the disparities between individual and organizational cultures. The paper's findings outcomes support previous research that suggests that cooperation within distributed teams in general, and organizational and individual cultural obstacles may hamper the effective implementation of agile ways of working in particular. Furthermore, we observed how integrating members of the team into the team's cultural principles is a critical mission, which may be especially prevalent in multinational companies with strong employee turnover. In this paper, an efficient solution consists of four phases proposed to bridge spatial, behavioral, and cultural barriers and limitations so that the distributed teams can collaborate as one team. In addition, some tools have been suggested that contribute to achieving the proposed solution, some of which must be installed on the distributed teams' centralized work

platform, and others work completely online. Such tools have strongly contributed to activating some of the core functions of the suggested solution. Moreover, the findings of this study might be used in future work and implemented in real-world projects that help the agile master in handling agile-based projects and managing global distributed teams.

References

1. School of Software, University of Technology Sydney, 15 Broadway/Ultimo NSW 2007, Australia. and Y. Ibrahim Alzoubi, Distributed Agile Development Communication: An Agile Architecture Driven Framework, JSW, 2015 681-694. doi: 10.17706/jsw.10.6.681-694.
2. Sul Haq, M Raza, A Zia, MNA Khan. Issues in Global Software Development: A Critical Review, JSEA. 2011; 04(10):590-595. doi: 10.4236/jsea.2011.410069.
3. W Aslam, F Ijaz. A Quantitative Framework for Task Allocation in Distributed Agile Software Development, IEEE Access. 2018; 6:15380-15390. doi: 10.1109/ACCESS.2018.2803685.
4. SV Shrivastava, U Rathod. A risk management framework for distributed agile projects, Information and Software Technology. 2017; 85:1-15. doi: 10.1016/j.infsof.2016.12.005.
5. M Younas, DNA Jawawi, AK Mahmood, MN Ahmad, MU Sarwar, MY Idris. Agile Software Development Using Cloud Computing: A Case Study, IEEE Access. 2020; 8:4475-4484. doi: 10.1109/ACCESS.2019.2962257.
6. Hess P Diebold, N Seyff. Understanding information needs of agile teams to improve requirements communication, Journal of Industrial Information Integration. 2019; 14:3-15. doi: 10.1016/j.jii.2018.04.002.
7. EM Schön, J Thomaschewski, MJ Escalona. Agile Requirements Engineering: A systematic literature review, Computer Standards & Interfaces. 2017-2019; 49:79-91. doi: 10.1016/j.csi.2016.08.011.
8. D Lloyd, R Moawad, M Kadry. A supporting tool for requirements change management in distributed agile development, Future Computing and Informatics Journal. 2017; 2(1):1-9. doi: 10.1016/j.fcij.2017.04.001.
9. M Fernandez-Diego, ER Mendez, F Gonzalez-Ladron-De-Guevara, S Abrahao, E Insfran. An Update on Effort Estimation in Agile Software Development: A Systematic Literature Review, IEEE Access. 2020; 8:166768-166800. doi: 10.1109/ACCESS.2020.3021664.
10. ES Hidalgo. Adapting the scrum framework for agile project management in science: case study of a distributed research initiative, Heliyon. 2019; 5(3):e01447. doi: 10.1016/j.heliyon.2019.e01447.
11. Zaitsev U Gal, B Tan. Coordination artifacts in Agile Software Development, Information and Organization. 2020; 30(2):100288. doi: 10.1016/j.infoandorg.2020.100288.
12. Nurdiani J Börstler, S Fricker, K Petersen, P Chatzipetrou. Understanding the order of agile practice introduction: Comparing agile maturity models and practitioners' experience, Journal of Systems and Software. 2019; 156:1-20. doi: 10.1016/j.jss.2019.05.035.
13. M Choras, *et al*. Measuring and Improving Agile Processes in a Small-Size Software Development Company, IEEE Access. 2020; 8:78452-78466. doi: 10.1109/ACCESS.2020.2990117.
14. MF Abrar, *et al*. Motivators for Large-Scale Agile Adoption From Management Perspective: A Systematic Literature Review, IEEE Access. 2019-2020; 7:22660-22674. doi: 10.1109/ACCESS.2019.2896212.
15. TJ Gandomani, Z Tavakoli, H Zulzalil, HK Farsani, The Role of Project Manager in Agile Software Teams: A Systematic Literature Review, IEEE Access. 2020; 8:117109-117121. Doi: 10.1109/ACCESS.2020.3004450.
16. D Šmite, NB Moe, J Gonzalez-Huerta. Overcoming cultural barriers to being agile in distributed teams, Information and Software Technology. 2021; 138:106612. doi: 10.1016/j.infsof.2021.106612.
17. Sarwar Y Hafeez, S Hussain, S Yang. Towards Taxonomical-Based Situational Model to Improve the Quality of Agile Distributed Teams, IEEE Access. 2020; 8:6812–6826. DOI: 10.1109/ACCESS.2020.2964432.