

Enterprise architecture contribution in distributed agile software development

Yehia Ibrahim Alzoubi¹  | Alok Mishra² 

¹College of Business Administration, American University of the Middle East, Egaila, Kuwait

²Faculty of Engineering, Norwegian University of Science and Technology, Trondheim, Norway

Correspondence

Alok Mishra, Faculty of Engineering, Norwegian University of Science and Technology, Trondheim, Norway.
Email: alok.mishra@ntnu.no

Abstract

Geographically distributed agile development may experience a high failure rate due to communication issues, which has a negative influence on project performance. One suggested solution in the literature is to enhance both communication and project performance by implementing agile enterprise architecture. However, the empirical evidence that supports this claim is still scarce. To address this gap, this study empirically explores the role of agile enterprise architecture as an artifact in distributed agile development. The findings of an in-depth qualitative case study from a dispersed agile development organization that involves teams distributed over three locations are used in this work. Over 2 months, data was gathered by interviewing 12 key members of the team and watching three Sprint sessions of agile software development. Text analysis qualitative approach was used to analyze the data. The findings imply that agile enterprise architecture has a positive effect on distributed agile software development communication, quality, and functionality. Agile enterprise architecture may also support on-time completion, but a trade-off with on-budget may be necessary. These findings provide valuable insights, frameworks, and best practices that support organizations in achieving greater agility, collaboration, and success in their distributed software development initiatives. As this is one of the first studies to look at the influence of agile enterprise architecture on distributed agile software development communication and performance, further research is needed to confirm and expand on the conclusions of this study.

KEYWORDS

agile, communication, enterprise architecture, geographically distributed agile development

1 | INTRODUCTION

Due to several benefits of integrating agile software development and geographically distributed software development teams such as faster time, closer to customers, and a low cost of development, most agile software development organizations have adopted what has been called “Geographically Distributed Agile Development” (GDAD).^{1–4}

GDAD refers to the collaboration of numerous dispersed teams on a single project, either inside the same country or beyond borders.⁵

Despite the above-mentioned advantages, GDAD has several problems, particularly in terms of communication. Communication challenges include any obstacle that decreases communication effectiveness and efficiency^{3,6} such as different locations, cultures, languages, personal skills, and regulations.^{7–10} Poor GDAD communication may

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial](https://creativecommons.org/licenses/by-nc/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.

© 2022 The Authors. *Systems Engineering* published by Wiley Periodicals LLC.

result in poor GDAD performance such as longer time and more cost.¹¹ Some solutions have been provided to enhance communication in this context, such as using new communication tools, exchanging visits between distributed sites, forcing all teams to speak the same language such as English, enhancing collaborative culture among distributed teams, and assigning customer representatives such as Product Owner.^{12–14} However, GDAD communication is still challenging compared to co-located software development teams that can achieve high communication efficiency and effectiveness using informal face-to-face communication.^{15,16}

One of the approaches that have been recently adopted by some researchers and professionals is using agile enterprise architecture in this context. Agile enterprise architecture view that represents a shared view of the evolving software architecture (where just-in-time details emerge and are shared incrementally—a dynamic picture of the agile enterprise architecture) to keep all team members informed about the architecture and relevant details, to facilitate and enhance GDAD communication.^{17–22} Agile enterprise architecture is significant because it provides information about all aspects of an organization in terms of different views such as solution architecture and strategy views.²³ For instance, the strategic view of the agile enterprise architecture will facilitate the teams in GDAD to achieve a clearer knowledge of the overall vision and goals at the enterprise and local team level for guiding their actions.²⁴ This will ensure that teams are not working in isolation. This is one of the objectives of agile enterprise architecture—to bring the isolated parts of a GDAD organization on the same page for effective communication. Moreover, several studies reported that implementing agile enterprise architecture in GDAD can keep all team members on track by providing them with the above-mentioned views that guide their local solution architecture design.^{19,25} As a result, implementing agile enterprise architecture may help distributed teams/team members to work toward common goals, while reducing the waste caused by redundancy, inconsistency, and duplication of work, and thus provide higher GDAD performance.^{26,27} In other words, implementing agile enterprise architecture may help enhance GDAD communication and GDAD performance. Therefore, there is interest in studying agile enterprise architecture in the GDAD context. However, the empirical evidence that supports the above claims is scarce.^{28,29}

Hence, the primary goal of this research is to address this gap by empirically investigating the impact of agile enterprise architecture as an “artifact” on GDAD performance and communication. The novelty of this research lies in its examination of the incorporation of agile enterprise architecture within GDAD. To the author’s knowledge, there are limited studies in this specific domain. The scarcity of literature becomes even more pronounced when examining the ramifications of agile enterprise architecture on both GDAD communication and performance. Accordingly, this research delves into the influence of agile enterprise architecture on GDAD’s communication and performance. While Alzoubi and Gill’s work²⁴ investigated this influence through a survey, our study takes a distinct approach by conducting a comprehensive case study of a prominent GDAD organization with a prolonged history of practicing agile enterprise architecture. This approach yields

valuable insights into practical experiences that have not been thoroughly explored previously. The main contributions of the study are as follows.

- Some managers have been inspired to implement agile enterprise architecture in the GDAD environment to enhance communication and project performance, although empirical support is scarce.³⁰ This study follows recent calls to study the impact of agile enterprise architecture on GDAD and extends the work of Alzoubi and Gill²⁴ by providing more insights and revealing hidden patterns of implementing agile enterprise architecture in GDAD. Hence, this study pays efforts to provide empirical evidence about implementing agile enterprise architecture in GDAD.
- This research provides a novel approach that may enhance GDAD performance and communication. Agile enterprise architecture is the blueprint and the common artifact that may keep GDAD team members on track while the development process iterates.
- Implementing agile enterprise architecture in GDAD may cut many extra expenses resulting from many visits (exchange) between distributed software development teams to enhance communication and better understand customer and business requirements, which provides a plausible solution to GDAD organizations.
- This study opens the black box of implementing agile enterprise architecture in GDAD and provides logical reasons for the rejection it faces by agile team members. It reveals the misunderstanding among GDAD members about what the agile enterprise architecture is and how it can be applied. In other words, there is a big debate between business teams and technical teams such as agile software developers on who creates the enterprise architecture principles and the value of implementing agile enterprise architecture in GDAD. While architecture is the responsibility of the strategic team, the technical development team usually is not involved in creating architectural principles or even does not know what enterprise architecture is. Therefore, it is critical to involve GDAD team members in creating the architecture and educate them about its value and contribution to enhancing performance and communication.
- In many large organizations, a multitude of information systems are deployed to tackle specific challenges within distinct functions, often lacking a cohesive strategy for integration into the broader organizational framework. While Enterprise Resource Planning (ERP) systems are suggested as a remedy for this integration gap, they often encounter issues due to their broad scope and notorious implementation complexities. The paper sets out to investigate the significance of enterprise architecture in agile development and the feasibility of employing agile methodologies in its creation. By doing so, it seeks to offer insights and solutions to the issues outlined, which can be a meaningful contribution to addressing the integration and agility challenges faced by large organizations.

A qualitative research approach was used in this study to achieve a better thought of the research phenomenon at hand.^{31,32} An in-depth exploratory case study was used to investigate the impact of agile enterprise architecture on GDAD communication and GDAD

performance. This paper is organized as follows. In Section 2, the conceptual background and literature related to the research problem are presented. In Section 3, the research method is discussed. In Section 4, the findings of this study are discussed. In Section 5, the findings summary, the study implications, and the study limitations are discussed. In Section 6, this paper is concluded.

2 | BACKGROUND AND RELATED WORK

2.1 | Conceptual background and theoretical foundation

A conceptual framework provides the structure of major constructs and their relationships.³³ It also helps in identifying the most important constructs for collecting and analyzing data.³³ This paper is based on the GDAD communication model.^{2,34} This model has three variables; agile enterprise architecture, GDAD communication, and GDAD performance. Agile enterprise architecture may positively affect GDAD communication and performance. Moreover, communication amongst teams has a favorable impact on GDAD performance.

An enterprise may refer to an entity that comprises collaborations or associations that are consistently involved in economic operations.³⁵ Architecture, on the other hand, may refer to the underlying notions or qualities of a system in its surroundings as embodied in its components, interconnections, and evolutionary principles.³⁵ In general, enterprise architecture is characterized as the organizational logic for business operations and IT activities.³⁶ The Federation of Enterprise Architecture Professional Organizations defines enterprise architecture as “a well-defined practice for conducting enterprise analysis, design, planning, and implementation, using a holistic approach at all times, for the successful development and execution of strategy.”³⁷ (p. 1) On the other hand, software architecture can be defined as the design choices concerning the system's holistic structure and its operational characteristics.³⁸

The Agile Manifesto³⁹ states that the responsive tight planning that develops throughout software development gives you greater ability to react to needs changes and is more successful than a formal project plan with a lot of documents upfront.⁴⁰ Highsmith⁴¹ states that agility strikes a balance between structure and versatility. The enterprise architecture of huge projects must be broken down into smaller components that can be constructed and evaluated one at a time.⁴² The development process is made simpler by architecture description, which is represented in the form of drawings or text that illustrate the primary components of the project.⁴² While traditional enterprise architecture focuses on heavy up-front detailed documentation, agile enterprise architecture is a responsive and people-oriented approach. Hence, agile enterprise architecture may be defined as “The process of collaboratively defining, sharing, and nurturing common business and technical visions and guidance within an organization to maximize long-term value.”⁴³ (Slide. 19) While conducting interviews and translating the findings, this definition of agile enterprise architecture was employed.

Communication effectiveness and efficiency are two components of the process of communication in GDAD, identified by the model. Providing information to another person who interprets it as intended with little disturbance and confusion is characterized as communication effectiveness.^{44,45} On the other hand, communication efficiency is described as the ability to transmit a message to a recipient with good quality while using the least amount of time, money, resources, and effort possible.^{45,46} The definitions of these two aspects of GDAD communication will be used to interpret the findings of this paper.

GDAD performance includes four aspects: on-budget completion, on-time completion, quality, and functionality.^{16,47} On-budget and on-time completions imply finishing the project within the pre-defined cost and duration, respectively.^{48,49} Functionality refers to meeting user and functional requirements for the developed software.⁴⁹ Software quality is described as providing a GDAD project that addresses a specific concern and improves the way its customers do their tasks.³⁴

GDAD communication faces many challenges that reduce communication effectiveness and efficiency.^{50–52} To enhance GDAD communication; strategies, tools, agile practices, and agile enterprise architecture artifacts can be adopted. The higher the GDAD communication efficiency and effectiveness, the more successful the project delivered in the environment will be. In this paper, agile enterprise architecture artifacts will be studied as boundary objects that may enhance GDAD communication. Thus, this paper is based on the “boundary object” concept that is derived from sociological literature.⁵³ Boundary objects refer to physical artifacts or abstracts such as maps, design drawings, and contracts that are used to facilitate coordination and knowledge sharing between parties or teams by providing different interfaces while allowing diversity in interpretation.^{53,54} This is because boundary objects can provide a common basis for a conversation about problems and solutions while mediating interactions between communicating teams.⁵⁵ Therefore, boundary objects that use agile enterprise architecture artifacts such as sketches, assembly drawings, and mockups may enhance GDAD communication by enabling teams/team members to correct their activities to stay on track.⁵⁶ However, to work as boundary objects, agile enterprise architecture artifacts and the description of the enterprise must be used by different teams/ team members to cross the knowledge boundaries between them.⁵⁶

2.2 | Related work

This section provides an overview of the existing literature related to the utilization of enterprise architecture in GDAD. The literature can be divided into three primary categories: the application of enterprise architecture in GDAD, and the effects of implementing agile enterprise architecture on GDAD communication and performance.

2.2.1 | Utilizing enterprise architecture in GDAD

Canat et al.⁵⁷ studied the applicability of combining enterprise architecture and agile software development by interviewing professionals

such as software developers and software architects in five companies. The authors found that agile software development and enterprise architecture can be combined. They also reported that there was a clear communication problem among architects, different teams, and project owners. Uludag et al.²⁹ studied the use of enterprise architecture principles in GDAD through a mixed data collection approach that is a case study and survey methods. The authors reported that agile enterprise architecture is an important success factor of GDAD. However, the traditional enterprise architecture (top-down) approach which enforces the use of architecture principles among agile stakeholders by connecting strategic considerations only in GDAD is ineffective. Rather, the agile enterprise architecture bottom-up approach which allows all agile team members to participate in the creation of principles and guidelines of enterprise architecture may make the results of enterprise architecture governance efforts more transparent. This approach may also increase the acceptance level of the agile team towards the enterprise architecture, which is reflected positively in the GDAD project's success.

In a study by Kaddoumi and Watfa,⁵⁸ a total of 156 individuals involved in enterprise architecture, possessing practical experience in the field, were surveyed with the aim of constructing a foundational framework that was intended to identify the sources of inspiration, facilitators, and barriers associated with agile enterprise architecture, using the guiding principles of the Agile Manifesto as a basis. The findings unequivocally illustrate the positive perception enterprise architects hold regarding the integration of agile methodologies into the process of enterprise design. The organization's size, its reliance on its enterprise architecture, and its awareness of agile practices significantly influence this favorable perception.⁵⁸ The outcomes also highlight that the willingness of both the business and IT sectors to embrace change stands out as the principal driving force behind the adoption of agile approaches in enterprise design.^{58,59}

In a study by Van Wessel et al.,⁶⁰ The authors demonstrated the synergy between enterprise design and agile development through an examination of five case studies encompassing IT-based projects within both corporate and governmental contexts. Their recommendation to organizations was to integrate specific elements from both approaches. This integration could be achieved by sharing development models, promoting the reuse of resources, and ensuring the uniformity of operational methodologies across different agile teams. Such an approach allows organizations to retain their agility while ensuring the alignment of actions with the broader organizational goals and interests.⁶⁰

2.2.2 | Impact of agile enterprise architecture on GDAD communication

Because it is not possible in GDAD, face-to-face communication may not always be the ideal option and may even act as an inhibitor.⁶¹ Moreover, due to insufficient description of a software component, the absence of proper enterprise architecture creates a communication obstacle through misinterpretation or a needless stream of

communication.⁶² By combining an agile enterprise architecture view with a centralized repository that is accessible to all distributed members using the available connection equipment, storyboards can be seen and updated instantaneously.²² This view might be utilized as a communication medium with a shared lexicon to promote mutual comprehension and communication efficacy.²² Cataldo et al.⁶³ presented a socio-technical alignment technique to assess the influence of task dependency congruence on software developer communication. Communication was shown to be more effective and efficient when congruence was significant.⁶³

2.2.3 | Impact of agile enterprise architecture on GDAD performance

Agile enterprise architecture, on the other hand, has been mentioned in the literature as a way to improve GDAD performance.⁶⁴ Agile enterprise architecture may also offer the foundation for architecture rules, which may increase implementation consistency and minimize the number of mistakes.^{22,25} The agile enterprise architecture perspective may provide a task allocation strategy, enhance GDAD team member interaction, and ensure continued integration across distributed teams.⁶⁵ The agile enterprise architecture view can be useful in establishing project functionality, for example.⁶⁶ Using the agile enterprise architecture view in conjunction with training may help developers enhance their knowledge and productivity.⁶⁷ The adoption of an architectural view to address team disagreements over system design helps speed up the development process while also ensuring that functionality and quality objectives are met.⁶² This might allow enterprises to provide software that meets set cost, time, and quality criteria.⁶⁸ Bass et al.⁶⁹ went even further, claiming that, thankfully, quality expectations about a system may be made merely based on an examination of its architecture.

Alzoubi and Gill²⁴ studied the relationships between agile enterprise architecture, GDAD communication, and performance using a survey method. Their findings showed a positive impact of agile enterprise architecture on GDAD communication efficiency and effectiveness as well as a positive impact on software quality, functionality, and on-budget completion. This research extends the work of Alzoubi and Gill²⁴ and investigates the relationships' insights between agile enterprise architecture, GDAD communication, and performance using an in-depth exploratory case study.

3 | RESEARCH METHOD

This paper is a part of a larger study, which investigates the agile enterprise architecture practice, artifact, and communication issues in the context of GDAD (see^{23,24,34,70}). This paper reports on the part of the study that focuses on the agile enterprise architecture as an artifact of GDAD communication and performance using a qualitative case study approach. This study used the same case study and interview material as in Alzoubi and Gill⁷⁰ and Alzoubi and Gill.²³ In their study, Alzoubi

and Gill,⁷⁰ discovered the main GDAD communication challenges. In addition, the analysis by Alzoubi and Gill²³ was done to investigate the extent to which the surveyed organization utilized the agile enterprise architecture in its development. This study, on the other hand, focuses on how agile enterprise architecture affected GDAD communication and performance in the surveyed organization. As a result, the research questions covered in this study are part of other questions asked during interviews. The following are the questions that were asked during the interview sessions.

- **RQ1:** What impact does agile enterprise architecture have on communication among the pensioner banking system (PBS) project's distributed teams?
- **RQ2:** What impact does agile enterprise architecture have on the functionality, on-time completion, quality, and on-budget completion of PBS projects?

In this research, a single case study technique was adopted,⁷¹ which allows researchers to deeply explore and analyze a single entity or phenomenon. This approach is particularly useful in this study context because the aim is to gain an in-depth understanding of the intricacies and complexities of a specific case.⁷¹ The single case study also helps to identify patterns, relationships, or variables that may warrant further investigation in future studies. This approach allows for a deep exploration of the contextual factors that influence the case.⁷¹ A case study should contain research questions, hypotheses, rational ties of the data to the hypotheses, and a unit of analysis, according to.⁷¹ Implementing agile enterprise architecture may improve GDAD performance and communication, according to the rationale tying the data from this article to the literature. A team member in a firm that uses agile enterprise architecture in GDAD is the unit of analysis. Evidence of the relationships between agile enterprise architecture and GDAD performance and communication is the main aspect of interpreting the case-based data.

3.1 | Research context

The data were gathered from a multinational corporation with its headquarters (18 employees) in Sydney, Australia. The firm (SUNC) specializes in the development of novel apps. SUNC has been performing GDAD and using the Scrum approach for almost 7 years. In its initiatives, SUNC used distributed software development teams of software testers and developers. This paper only studied the project area which we called the "pensioner banking system" (PBS) from June 2018 to July 2018. This initiative was initiated late in 2016 by SUNC. SUNC has two dispersed teams: one in China, which has 16 members and includes a scrum master, solution architect, and developers, and another in India, which has five individuals and includes a test lead, solution architect, and three testers. Enterprise software architect, business architect, program manager, delivery manager, integration manager, product owner, iteration manager, product owner, scrum master leader, analysts, reviewers, and software developers are among

TABLE 1 Participant demographics.

Participant	Role	Experience (years)	Interview method	Location
PA	Program manager	10	Face-to-face	Sydney
PB	Enterprise architect	9	Face-to-face	Sydney
PC	Product owner	7	Face-to-face	Sydney
PD	Scrum master leader	8	Face-to-face	Sydney
PE	Integration manager	11	Face-to-face	Sydney
PF	Tester leader	6	Face-to-face	Sydney
PG	Iteration manager	8	Face-to-face	Sydney
PH	Software developer	4	Face-to-face	Sydney
PI	Solution architect	7	Skype	India
PJ	Software tester	6	Skype	India
PK	Scrum master	4	Skype	China
PL	Technology architect	5	Skype	China

the most senior members of the Sydney team. The majority of the job is completed in Sydney. This comprises gathering software requirements, allocating tasks, integrating with other systems, and configuring the system. According to SUNC regulation, all new hires must undergo 1 month of training in Sydney to become familiar with the system and the technical jargon. Table 1 shows the characteristics of all study participants. All participants were labeled alphabetically for the sake of confidentiality.

3.2 | Data collection

Semi-structured interviews were used to acquire the majority of the data. We initiated the interview process by sending emails to all participants and securing their consent to proceed with the interviews. The majority of the interviews were carried out in person within the primary team setting, with eight interviews held at our Sydney location. The remaining interviews were facilitated via Skype. Throughout the interviews, the communication medium employed was English. Each interview was between 50 and 70 min long. Supplementary data were gathered by documenting observations made over 2 months at the primary team site in Sydney, including activities like sprint planning and routine face-to-face meetings. Additionally, notes were taken on various artifacts used, such as charts and sketches, to enrich the dataset. Each interview commenced with introductory questions, typically inquiring about the interviewee's role and responsibilities within the PBS project. Following this, we proceeded to present the above two research questions.

3.3 | Roles and development process

Three teams manage the release of a product every four Sprints (12 weeks). At the start of each Sprint, all teams must attend the Sprint

planning meeting. The meeting is normally held around 2 p.m. Sydney time to compensate for the time difference between dispersed teams, which is 3 h for China and 5 h for India. The three dispersed teams used the JIRA tool, an Atlassian Company application⁷² that supports Scrum methods, time-tracking features, and real-time performance metrics. JIRA is a widely used project management and issue-tracking software developed by Atlassian. Originally designed to facilitate bug tracking and issue management, JIRA has evolved into a versatile tool used for various project management purposes, including agile software development.⁷² SUNC also includes an online repository where teams may share code and modifications.

The Sydney executives were in charge of all planning, instruction preparation, and team responsibilities. Although the original agile enterprise architecture was designed and developed by the core team in Sydney, the SUNC technique is adopted to include distributed teams in the implementation of architecture rules and standards. Distributed teams are given more trust and responsibility under this method. Please see Alzoubi and Gill²³ for further information on developing and disseminating agile enterprise architecture principles. The enterprise software architect, based in Sydney, shares the agile enterprise architecture view with the software architects based in China and India at the start of the project. After each Sprint planning and after testing the new release, he discusses the business needs with the architect's team. He makes regular trips to each location to check that the standards are completed. Enterprise software architect, business architects, solution architects, infrastructure architects, and technology architects all have their private groups in the SUNC JIRA tools. This group is used by the architect's team to discuss architectural concepts, disagreements, progress, and so forth.

3.4 | Data analysis

Twelve interviews and observation notes were subjected to a theme analysis.³³ To obtain a holistic picture, significant ideas, patterns, and all relevant remarks from all participants were recognized and compared, and multiple contrasting and comparing of comments were undertaken.³³ The instructions established by Miles and Huberman³³ were applied to evaluate and code the acquired data, which involve four stages: data collection, data reduction, data presentation, and conclusion drawing. Following data collection, the data should go through a preparation procedure to ensure that they are suitable for analysis. As a result, all audio recordings, as well as notes taken during observations, were converted into Word documents. Initial coding is necessary for converting data into a usable form in the data reduction stage. This involves removing personal identifiers. Data visualization is sorting, condensing, and combining data into more legible representations like charts and tables. Each interview document was investigated to identify the elements related to a theme or category and then mapped into diagrams and tables. The final stage is drawing the conclusion, where tables and charts were created based on predetermined themes. Data coding, classification, and primary theme identification were all mapped to the three constructs established by²⁴ (i.e., agile

TABLE 2 Summary of agile enterprise architecture usage patterns.

Findings	
Agile enterprise architecture usage patterns	<ul style="list-style-type: none"> ■ The architecture owner plays a critical role in ensuring that agile enterprise architecture is satisfied and that developer conflicts are resolved (PA, PE) ■ Portfolio support and organizational culture may play a role in adhering to agile enterprise architecture standards and delivering high-quality projects (PA, PB) ■ Using some form of documentation of agile enterprise architecture enables tracking Sprint activities between software architecture and execution (PB, PG)

enterprise architecture, GDAD performance, and GDAD communication). The final stage seeks to make the acquired data into a cohesive and meaningful shape.³³

4 | RESEARCH FINDINGS

In the following subsections, the effect of agile enterprise architecture on GDAD performance and GDAD communication effectiveness and efficiency are discussed. Figure 1 presents a summary of this paper's findings. All dispersed teams will have access to a single repository including agile enterprise architecture views, questions, responses, software architectures, and so on, which they may use as a reference point when creating.

4.1 | Agile enterprise architecture usage patterns

Table 2 presents the summary of the key patterns of agile enterprise architecture. It is important to note that the case study explored in this research had been using the agile enterprise architecture for a few years at the time of the interviews. Although the concept and use of agile enterprise architecture were apparent to the core team in Sydney, few of the distributed team members (e.g., technical or development members) were unaware of the distinction. Accordingly, at the beginning of each interview, all participants were asked to define agile enterprise architecture in order to ensure that everyone was on the same page.

Even though the initial agile enterprise architecture was created and established by the main team in Sydney, the SUNC approach attempts to engage every team in the creation of agile enterprise architecture guidelines and principles. With this approach, distributed teams are given additional confidence and responsibility.⁷³ This approach utilizes the generic architecture principles process by taking into account the unique insight of each team. There are several views or understandings of agile enterprise architecture concepts.⁷⁴ Therefore, it was necessary to uncover how various participants define agile enterprise

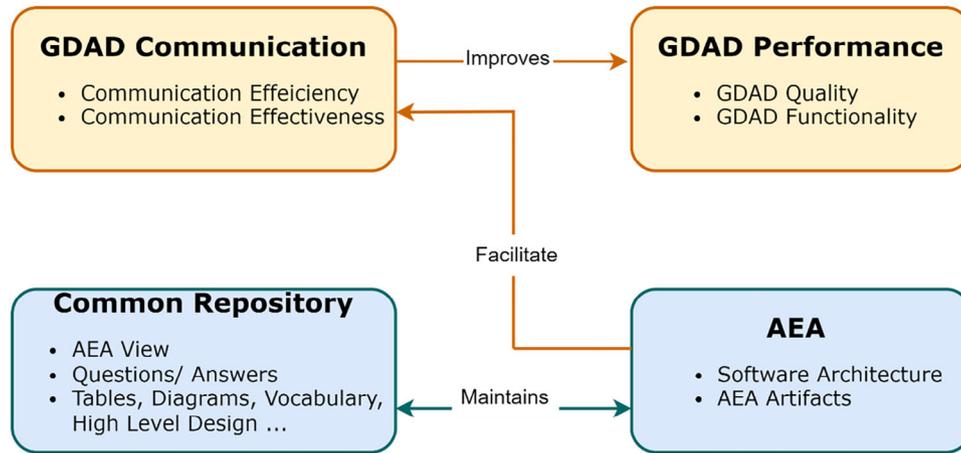


FIGURE 1 Impact of agile enterprise architecture (AEA) on GDAD.

architecture to understand and analyze the agile enterprise architecture in the context of the SUNC case study. Agile enterprise architecture should have flexible agile development iterative and incremental abilities and principles. PL reported that: “agile enterprise architecture focuses on the flexibility of change... iterative, incremental approach.”

The role of the architecture is to link and integrate relevant processes, and organizational elements, including technologies, information, procedures, and infrastructure. PB reported that: “architecture is all about linking.” Thus, agile enterprise architecture spans numerous systems and functional units across the organization. Furthermore, agile enterprise architecture enables specific business units to develop securely by striking the proper steadiness between business requirements and IT productivity. So, agile enterprise architecture ensures that its IT strategy is aligned with the demands of the business. PG stated that agile enterprise architecture is: “...very much alignment.” Agile enterprise architecture is also a map that retains all troops on the same page.²³ Artifacts created by the agile enterprise architecture will be always accessible and reflect the complete picture that needs to be communicated to GDAD teams for execution.⁷⁵ Team members can be assisted by agile enterprise architecture throughout the development process. PE reported that: “create the complete image, making sure it moves in the right direction.”

Agile enterprise architecture artifacts (e.g., maps, diagrams, and tables) are designed and promoted with simple terminology so that all team members can understand them. All teams have access to the architecture artifacts through a central repository. For instance, the Sydney team established the core of agile enterprise architecture and shared it with GDAD teams to create a shared view and solid foundation for execution. Accordingly, to be easily understood by GDAD teams, the agile enterprise architecture common view should be produced and communicated with simple terminology.⁵⁷ All teams should be given standards, checklists, and examples. PG explained: “the view is communicated among all architects, and artifacts are housed in an online repository so anybody who wants it can go look at it. The architecture owner or solution architect will go through the view... with the rest of the team and explain them.”

Further, it has also been noticed that agile enterprise architecture needs to be communicated to senior management to support the decision-making process.⁷⁶ According to PB, “We have a discussion group for all architects. We assist our executives (CIOs) by providing them with sufficient information to make informed decisions.” Agile enterprise architecture is a strategic discipline. Strategic planning combined with an agile mentality may create a noticeable shift that changes the game.⁷⁷ This happens if and only if the strategic agile enterprise architecture and tactical delivery GDAD teams are constantly aligned. PA reported that: “planning of agile portfolio ... defines the strategy on how much to invest.” Moreover, while agile development values working software products over detailed documentation, this does not ignore the value of documentation.³⁹ PB reported that “enterprise architecture and solution architecture are both documented.” Well, this seems to suggest it is not so much about the architecture documentation as such, rather, it is about its availability and access to relevant GDAD team members.

4.2 | Communication perspectives

Table 3 presents the summary of the key findings of the relationship between agile enterprise architecture and GDAD communication effectiveness and efficiency. The two aspects of GDAD communication (i.e., effectiveness and efficiency), according to Alzoubi and Gill,⁷ are discussed in the following sections.

4.2.1 | Communication efficiency perspective

The Sydney location has an open office environment. This enables more casual dialogue, which enhances the efficiency of communication since an open environment allows the software development team to have a more efficient daily conversation about the project’s progress.⁷⁸ PD mentioned: “We have an open channel with the project leads all the way... It is much faster than solving any issue.” It is clear to note here

TABLE 3 Relationship between agile enterprise architecture and GDAD communication.

Relationship	Findings
Effect of agile enterprise architecture on GDAD communication effectiveness and efficiency	<ul style="list-style-type: none"> ■ The agile enterprise architecture view may decrease the problems of interactions in the GDAD environment, which may make communication faster (PI) ■ Agile enterprise architecture view that is available and accessible by all teams may enhance communication efficiency (PB, PG) ■ Agile enterprise architecture artifacts may be used as a common language and may help in delivering a clear and understandable message (PB, PG) ■ The architecture owner's role may enhance the effectiveness of GDAD communication (PL) ■ Using showcases may enhance GDAD communication effectiveness (PK)

Abbreviation: GDAD, geographically distributed agile development.

that achieving the same level of communication efficiency in GDAD is difficult when compared to co-located teams.⁷⁹ This is owing to the numerous obstacles that GDAD communication faces, such as time disparities, various languages, and diverse cultures across distributed teams.⁷

The agile enterprise architecture view can be represented by various artifacts such as charts, diagrams, or tables. Solution architecture is a part of the agile enterprise architecture view. This view was updated and refined before being stored in the repository during PBS project iterations, which may enhance communication efficiency with GDAD teams. PG reported: "Solution architects share enterprise architecture artifacts with their teams. They help a lot to speed up the conversation with off-shore members." Furthermore, the agile enterprise architecture view represented a reference for answers since it held all members' questions and related discussions, as was reported by some participants. Accordingly, communication efficiency can be facilitated among GDAD teams. PE reported: "It is helpful because it gives them the directions...they know what we talking about...they do not ask many questions." Moreover, the Sydney team used the agile enterprise architecture view to check if the progress was going according to the plan. This may increase the efficiency of horizontal and vertical communication among GDAD teams and with business teams. PG reported: "We divide the enterprise architecture view in the releases. I use visual methods... I think this increases the speed of conversation among development teams and with business people."

4.2.2 | Communication effectiveness perspective

Lack of effective communication can lead to issues including a lack of collaboration between remote teams and a lack of understanding of customers' needs. PG mentioned that: "...information should

TABLE 4 Relationship between agile enterprise architecture and GDAD performance.

Relationship	Findings
Effect of agile enterprise architecture on GDAD performance	<ul style="list-style-type: none"> ■ Agile enterprise architecture may keep GDAD teams on track by sharing the goals and scope of the project, which may help in delivering high quality and functionality, on-budget, and on-time projects (PG, PL) ■ Agile enterprise architecture may provide the performance estimations and architects help in sharing agile enterprise architecture artifacts and solving the conflicts among GDAD teams (PB, PG) ■ Distributed providers should agree before any project on delivering it on-time (PA, PI) ■ A tradeoff between functionality and quality vs. on-budget and on-time completion (PA, PC, PI)

Abbreviation: GDAD, geographically distributed agile development.

be delivered and understood properly to deliver the correct product." Moreover, communication effectiveness is better and easier to accomplish within co-located software development teams than among GDAD teams due to less informal conversation and various GDAD communication issues.²⁴ As discussed earlier, agile enterprise architecture artifacts can be used as a blueprint and as a common language for communication between GDAD teams. Further, the production of new agile enterprise architecture artifacts can be guided by existing agile enterprise architecture artifacts. Thus, the availability and access to agile enterprise architecture artifacts, among GDAD teams, were considered a key driver of the effectiveness of conversations. PB explained: "We provide our iteration updates... to all of our dispersed locations, which is a useful tool for keeping everyone informed about the dialogue."

The agile enterprise architecture's view was reported to ensure that members are clear about what they want to talk about, resulting in a clearer message. PB reported: "The agile enterprise architecture view is beneficial since it advises them of the steps, we take to complete our project. They have a lot of questions, and if agile enterprise architecture view isn't there, it will make answering them a hard job." Moreover, the agile enterprise architecture was reported as an important tool for minimizing the frequency of communication among GDAD teams in SUNC. The architecture owner may be able to assist in improving GDAD communication effectiveness while also reducing the frequency of communication.²³ PL reported that: "for good communication with technical teams, having an architect or architecture expert for each team is beneficial."

4.3 | GDAD performance perspective

Table 4 presents the relationship between agile enterprise architecture and GDAD performance. Some participants, such as PA, refer to functionality as the project's scope: "My major aim is to get the team to concentrate on the breadth of work that has to be completed." Different organizations take different ways of determining the performance

elements of their projects. All participants agree that during Sprint planning meetings, all stakeholders are present, all tasks are specified and allocated to various teams, all priorities are established, and all estimations for each task are provided. Time and cost, on the other hand, are difficult to predict. "We usually attempt to form up the cost upfront in agile projects, because the vendor cost is typically more than the estimated," PA stated. Organizations need to be able to demonstrate quantitative value delivery for agile enterprise architecture to succeed. To retain momentum and support from technical teams and business teams, agile enterprise architecture must consistently produce demonstrable value. "Stakeholders must realize the benefits of our business activities in order to recognize and support our plan," PB said. It isn't only hanging work or placing stuff on the wall. Stakeholders must recognize the benefits of doing so. PG said: "Teams benefit from EA, but whether or not they deliver successful projects is determined by their ability to engage, make wise decisions, experiment, and operate as a team."

It was clear from the participants' feedback that it is hard to achieve the designated levels of performance aspects in the GDAD environment. This was reported by PA: "those aspects are constantly at odds; therefore, time is typically pitted against cost, and scope is pitted against both time and expense. This is more difficult in agile development than in plan-driven development because vendors do not operate agile. With distributed agile development, the problem becomes much more difficult." Even though quality should be supplied as agreed with consumers, several participants stated that quality-related development issues were not unusual. Software quality is adversely correlated with on-time and on-budget completion, as well as the possibility of late and over-budget projects to attain excellent quality. Nevertheless, to fulfill deadlines, a quality trade-off is frequently accepted. PC said: "I believe that on-time and on-budget are linked to the quality and that improving quality will come at the cost of more time and money." Furthermore, as PE pointed out, the trade-off may be apparent even between on-budget and on-time completion. PE reported: "I believe there is always some sort of trade-off between these factors (time, budget, and quality). So, you won't be able to have all of them right in any one project." PH added: "It will be finished on budget and on time, but the quality may be compromised."

Requirements and revisions may be discussed and shared rapidly when all members are in the same open office environment.⁸⁰ However, in a GDAD situation, the costs of getting people together to talk about problems and find solutions may quickly build up.⁸¹ This may result in delayed or costly projects, producing anxiety among team members and leaders.⁸² Hence, agile enterprise architecture views assure that all GDAD teams will be on the same page and can code without falling off course, even if they don't reach high levels of communication effectiveness and efficiency. Consequently, they will be able to meet deadlines and costs while maintaining quality and functionality (to the best of their ability). PG explained: "Plans and estimates are commonly used to establish functionality and quality, so everyone is working toward the same goal." However, according to PL, on-time completion is very hard to achieve although other performance aspects can be enhanced. PL explained: "On-time completion has never happened in the seven years I've worked in (some start-ups, some large

corporations). Agile enterprise architecture can help you meet deadlines by reducing the likelihood of missing them." Although the purpose of agile enterprise architecture is to connect different parts of an organization. Given actual time and resource constraints, agile enterprise architecture must focus on deliverables that are both feasible and relevant. As a result, the agile enterprise architecture should change as the software development process progresses (i.e., agile enterprise architecture's objective and scope also evolve). Customer and business requirements, market changes, team member habits, and information and technology changes may all contribute to this evolution. PE said: "The business has a significant influence on the scope and technology parts."

Agile enterprise architecture may also be used to assess the contribution of IT and project portfolios to business objectives. The role of agile enterprise architecture, on the other hand, isn't to single out or compel teams to use certain resources or apps; rather, it is to offer teams a variety of alternatives and estimates for completing a project. PB said: "...enterprise architect assists in offering some alternatives for the teams based on our estimation." Further, it was noticed that estimates may vary somewhat during delivery, depending on the risks. In addition to estimates, the role of the architecture owner was critical in resolving disagreements between various architectural design choices, which was a regular issue, particularly in GDAD teams. PI reported: "Conflict in design is unavoidable. A smart architect should be able to resolve these issues on time."

5 | DISCUSSION

Traditional enterprise architecture is usually described as being too slow, heavily documentation-oriented, and inflexible. To address these drawbacks of traditional enterprise architecture, agile enterprise architecture was promoted. However, the empirical evidence of the use of agile enterprise architecture in GDAD is relatively uncommon. This work represents an effort to empirically find out how agile enterprise architecture affects GDAD symptoms. An in-depth case study was used to accomplish this, utilizing a qualitative technique that includes 12 interviews, in addition to the notes that were collected during teamwork observation. The purpose of this study is to address two research questions: how does agile enterprise architecture influence GDAD communication (RQ1) and how does agile enterprise architecture influence GDAD performance (RQ2)? This paper extends the work of Alzoubi and Gill²⁴ and explores insights into the relationships between agile enterprise architecture with GDAD performance and GDAD communication. The following sections discuss the implications as well as the limitations of this study.

5.1 | Summary of the findings

Several insights have been identified in this research. These insights can be summarized in four pillars: Agile enterprise architecture artifacts, in GDAD, can work as boundary objects, can enhance communication effectiveness and efficiency, enhance functionality and quality, and preserve the agility essence.

5.1.1 | Artifacts as boundary objects

Shared understanding is critical for the success of GDAD. This is where the concept of “boundary objects” comes into play. In the context of agile enterprise architecture, artifacts such as architectural diagrams, documentation, and models can act as these “boundary objects.” These objects act as bridges between different perspectives, knowledge domains, and cultures. These artifacts can serve as a common ground or shared reference point for team members with diverse backgrounds, roles, and expertise, as it was reported by PE: “create the complete image, making sure it moves in the right direction.” Hence, agile enterprise architecture artifacts can establish a common language or visual representation that helps team members from various backgrounds and roles understand and discuss complex architectural concepts. Moreover, they help align the team’s understanding of the architecture, ensuring that everyone is on the same page regarding the system’s structure and design. Furthermore, by providing a visual or documented framework, these artifacts promote collaboration. Team members can work together more effectively, even if they are geographically dispersed. Additionally, when agile teams need to make architectural decisions, these artifacts serve as a basis for discussions and evaluations, ensuring that decisions are well-informed and supported by a shared understanding, as it was reported by PB: “...assist our executives (CIOs) by providing them with sufficient information to make informed decisions.”

5.1.2 | Enhanced communication

Agile enterprise architecture artifacts serve as a vital tool for improving communication efficiency and effectiveness in GDAD, as it was reported by PG: “... I think this increases the speed of conversation among development teams and with business people.” They provide a shared understanding, reduce misunderstandings, and facilitate collaboration between distributed members. This contributes to a smoother and more productive development process. Agile enterprise architecture artifacts encourage collaboration among distributed team members. Team members can refer to these artifacts during meetings, planning sessions, and discussions, allowing them to collaborate more productively, even when they are geographically dispersed. Also, clear communication facilitated by these artifacts minimizes the likelihood of misunderstandings that could lead to rework. Rework is a common concern in software development, and efficient communication helps prevent unnecessary revisions and saves time and resources.

5.1.3 | Enhanced GDAD performance

Agile enterprise architecture artifacts improve GDAD communication efficiency and effectiveness and enhance GDAD functionality and quality, as it was reported by PB: “...is a useful tool for keeping everyone informed about the dialogue.” These artifacts serve as a record of architectural decisions and discussions. This documentation ensures that important information is not lost or forgotten. It provides a historical context that team members can refer to when needed, which

is particularly valuable in a distributed setting where real-time communication may be limited. Moreover, when issues or challenges arise during development, having Agile enterprise architecture artifacts readily available allows team members to diagnose problems more efficiently. They can trace the impact of potential changes or identify the source of issues, which streamlines the troubleshooting process. Additionally, the artifacts offer clarity by presenting complex architectural concepts in a visual and organized manner. This clarity aids in conveying ideas and information more effectively. Team members can quickly grasp the architecture’s key components, relationships, and design principles, resulting in more precise and efficient communication. Furthermore, the presence of these artifacts fosters transparency within the development process. All team members, regardless of their location, have access to the same information, reducing information silos and ensuring that everyone is on the same page.

5.1.4 | Preserving agility

The adoption of agile enterprise architecture doesn’t compromise the agility of GDAD processes and tools. Instead, it adds structure and organization to enhance collaboration and communication.^{83,84} It’s crucial to dissect how GDAD and agile enterprise architecture may coexist without harming one another in order to comprehend this claim more fully:

- **Flexibility and Adaptability:** Agile enterprise architecture is intended to be agile and responsive when properly applied. It supports adaptability and iterative development, as it was reported by PL: “Agile enterprise architecture focuses on the flexibility of change... iterative, incremental approach.” This is consistent with the fundamental tenets of software development agility. It implies that teams may adapt architecture and take into account new needs without compromising agility. This is complemented by agile enterprise architecture, which offers a structured yet flexible foundation. It guarantees that architectural choices and adjustments are made progressively and in accordance with the changing project needs.
- Agile development places a focus on adapting quickly to change and user input, but it doesn’t eliminate the necessity for some degree of architectural structure. Agile enterprise architecture maintains agility while balancing the need for architectural direction.²³
- Metrics and evaluation mechanisms may be incorporated into agile enterprise architecture to evaluate the effects of architectural changes on the development process agility. This strategy guarantees that agile enterprise architecture principles and practices are compatible.

5.2 | Implications for research

This paper has several implications for research. The findings reveal an important gap that urges more research, namely, how GDAD communication and GDAD performance can be enhanced using agile enterprise architecture in practice. Researchers are expected to pay

more attention to how much agile artifacts can be used in GDAD, and what new approaches or strategies to use when adopting agile enterprise architecture in GDAD. Therefore, there is a need for more empirical research in adopting agile enterprise architecture in the GDAD environment.

An agile enterprise architecture view can provide phrases and notions that all GDAD teams can use as a shared language. All solutions that may be utilized in refactoring to achieve a basic design are included in the agile enterprise architecture view. This view may help to improve GDAD communication by allowing for simplicity, which is necessary for improving communication effectiveness and efficiency. Moreover, the lack of adequate architecture reduces GDAD team members' understanding. Even though the agile enterprise architecture view is not thorough, project planning employing an agile enterprise architecture view could have a favorable influence on the GDAD project.²² Therefore, the agile enterprise architecture view may represent a boundary object, which can provide a common basis for a conversation about problems and solutions while mediating interactions between communicating teams.⁵⁶ Hence, the agile enterprise architecture view could have a positive impact on GDAD communication effectiveness and efficiency by providing a blueprint and a common language that keeps distributed teams on the same page during development iterations. These findings provide empirical evidence that agile enterprise architecture can mitigate the effects of communication challenges by facilitating and enhancing GDAD communication. These findings are aligned with the findings of Alzoubi and Gill.²⁴ In future research endeavors, there is a compelling opportunity to examine how specific domains or viewpoints within agile enterprise architecture—such as technological or business perspectives—influence communication dynamics within the context of GDAD. These insights can inform best practices and guidelines for optimizing communication and collaboration in the context of agile enterprise architecture adoption.

Moreover, this paper's findings reveal a positive impact of agile enterprise architecture on GDAD performance. This is consistent with previous literature. Agile enterprise architecture offers the basis for architecture principles, reducing the number of mistakes, improving execution integrity, using the same design and language patterns, scoring from the same quality criteria, and using a consistent scoring mechanism.^{22,25,66} However, the findings reveal that the impact of using the agile enterprise architecture view was more recognized and supported on GDAD functionality and quality, which is also aligned with the findings of Alzoubi and Gill.²⁴ However, the impact of the agile enterprise architecture view on on-budget and on-time completion was not clear, which is partially aligned with the findings of Alzoubi and Gill,²⁴ where agile enterprise architecture had no substantial influence on on-time completion. Whereas the business team believes that agile enterprise architecture helps with all elements of performance, technical teams, such as software developers, believe that agile enterprise architecture is more focused on functionality and software quality. This is because, at the start of a project, the GDAD teams, which may include suppliers or contractors, must agree on the costs and delivery date. PE stated: "... they have to produce on time and budget, as agreed upon at the start of the project." Therefore, more studies are needed to val-

idate these results. Future research in this area holds the potential to uncover not only the technical ramifications but also the customer-oriented and team-oriented consequences of applying agile enterprise architecture in GDAD. This holistic perspective can offer a comprehensive understanding of the benefits and challenges associated with the integration of agile enterprise architecture principles across various facets of software development.

On the other hand, implementing agile enterprise architecture in GDAD is still facing difficulties among technical developers. More research is needed on the nature and purpose of agile enterprise architecture, as well as how to use enterprise architecture in agile methods, as it was mentioned by several participants. For example, PJ explained that "... agile enterprise architecture is applied at a strategic level." This is in line with Canat et al.⁵⁷ results, which reported that there are problems among architects, technical teams, and project owners about using and sharing agile enterprise architecture. Given the apparent prevalence of these problems, it underscores the necessity for further research in this domain.²⁹ Future research would aim to delve deeper into these challenges, their underlying causes, and potential solutions. By doing so, future research may address the critical question of how best to mitigate these issues and enhance the successful application of agile enterprise architecture principles within GDAD organizations.

5.3 | Implications for practice

This paper identified several implications for practice. The findings of this paper show that the agile enterprise architecture view may enhance GDAD communication by enabling simplicity and consistency (terminology), which is needed to enhance communication efficiency and effectiveness. This is an interesting implication for practitioners since communication has been a major challenge for GDAD. This paper demonstrates how agile enterprise architecture potentially improves GDAD communication without burdening agile teams, presenting new communication methods or technology, or unnecessarily boosting the number of trips among distant locations. This paper also presents the empirical finding that agile enterprise architecture implementation potentially improves GDAD performance. This underlines the need to use agile enterprise architecture rather than traditional enterprise architecture to generate effective IS projects, as traditional enterprise architecture has been criticized for not providing value. Using the agile enterprise architecture view and coaching in conjunction with training might help developers increase their knowledge and productivity.^{67,85} Because of this understanding, GDAD may be able to develop software that meets its cost, schedule, and quality objectives.⁶⁸ Moreover, the findings imply that documenting agile enterprise architecture concepts (e.g., artifacts, models) is an essential component to guide GDAD teams without being too prescriptive and allow the solution design to emerge during the development sprints. These findings complement prior research (e.g., Refs. [29, 64]).

The application of agile enterprise architecture concepts must be flexible enough to allow team leaders or members to pick a specific action depending on the results of continuing collaborative progress.

This finding is consistent with Uludag et al.²⁹ finding, which recommends that the GDAD firms should enable the technical teams to participate in the development of agile enterprise architecture concepts to ensure project success and team commitment. Therefore, GDAD firms are urged to be more liberal when developing the agile enterprise architecture view by involving both technical and business members. Furthermore, the findings show that the architecture owner plays a critical role in sharing and discussing agile enterprise architecture artifacts with their teams. This role also aids in the detection and resolution of disputes among members of a GDAD team. This aids with meeting pre-determined deadlines and budgets while maintaining a high level of quality and functioning. Therefore, GDAD practitioners are encouraged to focus more on the architecture owner role. To corroborate this claim and provide a better understanding of how different architectural actors interact in the GDAD ecosystem, more empirical research is required.

Some participants expressed the idea that enterprise architecture works independently from software development teams, implying that traditional enterprise architecture and agile enterprise architecture have no differences. PJ explained that “the execution level does not prefer it.” This view seems to ignore the societal implications of agile enterprise architecture and its capacity to adapt in comparison to traditional enterprise architecture. This view, on the other hand, implies that the agile enterprise architecture owner may not be a member of the GDAD team. However, it does not rule out the possibility of agile enterprise architecture artifacts being disseminated or transmitted among remote teams by a member who can act as an agile enterprise architecture owner. Such GDAD organizations may employ hybrid agile enterprise architecture. Alternatively, they could merely stick to the traditional enterprise architecture strategy, which might not produce the intended outcomes in a fluid and flexible agile environment. The GDAD team’s response could be a rejection or failure to “really follow” the traditional enterprise architecture plan.

To sum up, it is critical to emphasize that agile enterprise architecture artifacts can be communicated to GDAD teams through efficient and effective communication planning and approaches for maximizing performance outcomes. When using agile enterprise architecture in GDAD, more emphasis should be paid to disseminating agile enterprise architecture across distributed agile software development teams, highlighting its merits, and instructing team members on how to use it are also very important. Architecture owners should cooperate with other teams throughout development iterations. They should assist and attempt to resolve any misunderstandings between the members. Moreover, the agile enterprise architecture view should be simple and have clear artifacts. Agile enterprise architecture, communication, and GDAD artifacts should be kept and updated in a shared repository that is available to all team members.

5.4 | Limitations

When evaluating the conclusions of this work, like any other empirical research, several limitations must be acknowledged. First, the defini-

tion of agile enterprise architecture for each participant was not clear and in most cases is defined as the traditional EA. This represented a challenge while conducting interviews. Therefore, the definition of agile enterprise architecture was clarified to participants. Furthermore, collecting data through the interview method is subjective by nature. Hence, to avoid and decrease the impact of this limit, an interview protocol was developed, based on the study’s objectives. This protocol was used during the interviews to guide the interview process. In addition, the interview findings were triangulated with the observation of teamwork. Another limitation was related to the duration of the observation. The duration was based on prearrangement with the team project manager at the Sydney location. Future research is encouraged to conduct longitudinal as well as cross-case analysis studies on how agile enterprise architecture may affect GDAD performance and communication. Another limitation of this study can be the bias of the source of data collected, which tends to be more representative of managerial and architectural roles in the Sydney region, as shown in Table 1. Additionally, the study’s focus appears to be weighted more towards participants based in Sydney compared to those in the Chinese and Indian locations. Moreover, the number and selection of the participant process may be considered another limitation. Although all major roles were interviewed, not all team members were interviewed due to their unavailability and time constraints. The availability, necessary skills, and desire to participate in the interview were all factors in the selection process.

6 | CONCLUSIONS

GDAD has become an important agile software development approach. However, the communication issue is still the main challenge of GDAD which negatively impacts GDAD performance and results in a high failure rate. Hence, one suggested solution was to implement agile enterprise architecture, in general, which may positively affect GDAD performance and GDAD communication. Through in-depth observation and interviews, this work empirically explored this claim. This paper addressed RQ1 and RQ2 by exploring the impact of agile enterprise architecture on GDAD performance and communication. It also investigated how agile enterprise architecture is developed, shared, and used in large GDAD companies. The findings of this paper provide empirical evidence that the agile enterprise architecture artifacts may represent boundary objects among GDAD teams which potentially enhance GDAD communication, functionality, and software quality. Moreover, the findings reveal that implementing agile enterprise architecture in GDAD is still in its early stages and needs more work from both the industry and academia. While business appreciates the role of agile enterprise architecture, technical agile software development teams are still learning about the value of agile enterprise architecture in GDAD. This is one of the first studies to look at the involvement of agile enterprise architecture in GDAD. Therefore, more research is needed on what and how much agile enterprise architecture artifacts can be used in GDAD, and which new approaches or strategies should be used when implementing agile

enterprise architecture in GDAD. This paper has established the basis for other researchers to study the importance of implementing agile enterprise architecture in GDAD.

ACKNOWLEDGMENTS

It was challenging to find a suitable case and relevant participants for studying a unique combination of agile enterprise architecture and GDAD. We wish to express our heartfelt gratitude to the study participants for their time and substantial contributions to this study.

CONFLICT OF INTEREST STATEMENT

Authors declare no conflict of competing interest.

DATA AVAILABILITY STATEMENT

The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

ORCID

Yehia Ibrahim Alzoubi  <https://orcid.org/0000-0003-4329-4072>

Alok Mishra  <https://orcid.org/0000-0003-1275-2050>

REFERENCES

- Khan AA, Shameem M, Nadeem M, Akbar MA. Agile trends in Chinese global software development industry: fuzzy AHP based conceptual mapping. *Appl Soft Comput*. 2021;102:107090.
- Alzoubi YI, Gill AQ, Moulton B. A measurement model to analyze the effect of agile enterprise architecture on geographically distributed agile development. *J Softw Eng Res Dev*. 2018;6:1-24.
- Shameem M, Kumar RR, Kumar C, Chandra B, Khan AA. Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process. *J Softw Evol Process*. 2018;30:e1979.
- Mishra A, Alzoubi YI. Structured software development versus agile software development: a comparative analysis. *Int J Syst Assur Eng Manag*. 2023;14:1504-1522.
- Dreesen T, Diegmann P, Rosenkranz C. The impact of modes, styles, and congruence of control on agile teams: insights from a multiple case study. Proceedings of the 53rd Hawaii International Conference on System Sciences, Grand Wailea, Hawaii, 2020.
- Clark HH, Brennan SE. Grounding in communication. *Perspect Soc Shar Cog*. 1991;13:127-149.
- Alzoubi YI, Gill AQ, Al-Ani A. Empirical studies of geographically distributed agile development communication challenges: a systematic review. *Inf Manag*. 2016;53:22-37.
- Beecham S, Clear T, Lal R, Noll J. Do scaling agile frameworks address global software development risks? An empirical study. *J Syst Softw*. 2021;171:110823.
- Griffin L. Implementing lean principles in scrum to adapt to remote work in a Covid-19 impacted software team. In: Przybytek A, Miller J, Poth A, Riel A, eds. *Lecture Notes in Business Information Processing*. Springer; 2021:177-184.
- López-González M, Serrano-Gómez L, Miguel-Eguía V, Muñoz-Hernández J, Sánchez-Núñez, M. Comparative analysis of the scrum and PMI methodologies in their application to construction project management. In: Muñoz JLA, Blanco JLY, Capuz-Rizo SF, eds. *Project Management and Engineering Research*. Springer; 2021:17-31.
- Al-Saqqa S, Sawalha S, AbdelNabi H. Agile software development: methodologies and trends. *Int J Interact Mob Technol*. 2020;14:246-270.
- Buchan J, Talukder ANA, Senapathi M. Coordination in distributed agile software development: insights from a COTS-based case study. Proceedings of the Australasian Conference on Information Systems (ACIS 2019), Perth Western Australia, 2019; 942-952.
- Dingsøyr T, Moe NB, Seim EA. Coordinating knowledge work in multiteam programs: findings from a large-scale agile development program. *J Proj Manag*. 2018;49:64-77.
- Stadler M, Vallon R, Pazderka M, Grechenig T. Agile distributed software development in nine central European teams: challenges, benefits, and recommendations. *Int J Comput Sci Inf Technol*. 2019;11.
- Aslam W, Ijaz F. A quantitative framework for task allocation in distributed agile software development. *IEEE Access*. 2018;6:15380-15390.
- Vithana V, Asirvatham D, Johar M. An empirical study on using agile methods in global software development. Proceedings of the 18th International Conference on Advances in ICT for Emerging Regions (ICTer 2018), Colombo, Sri Lanka. 2018; 150-156.
- Ambler S. Agile modeling: Communication on agile software projects. 2014a.
- Avritzer A, Paulish D, Cai Y, Sethi K. Coordination implications of software architecture in a global software development project. *J Syst Softw*. 2010;83:1881-1895. doi:10.1016/j.jss.2010.05.070
- Batra D, Xia W, VanderMeer D, Dutta K. Balancing agile and structured development approaches to successfully manage large distributed software projects: a case study from the cruise line industry. *Commun Assoc Inf Syst*. 2010;27:379-394.
- Islam G, Storer T. A case study of agile software development for safety-critical systems projects. *Reliab Eng Syst Saf*. 2020;200:106954.
- Korhonen JJ, Lapalme J, McDavid D, Gill AQ. Enterprise architecture for the future: towards a reconceptualization of EA. Proceedings of the 18th IEEE Conference on Business Informatics (IEEE CBI 2016), Paris, France, 2016.
- Kornstadt A, Sauer J. Tackling offshore communication challenges with agile architecture-centric development. Proceedings of the Working Conference on Software Architecture (WICSA 2007) Mumbai, India, 2007; 28-31.
- Alzoubi YI, Gill AQ. Can agile enterprise architecture be implemented successfully in distributed agile development? Empirical findings. *Glob J Flex Syst Manag*. 2022;23(2):221-235.
- Alzoubi YI, Gill AQ. An empirical investigation of geographically distributed agile development: the agile enterprise architecture is a communication enabler. *IEEE Access*. 2020;8:80269-80289.
- Madison J. Agile architecture interactions. *IEEE Softw*. 2010;27:41-48.
- Britto R, Smite D, Damm L-O, Börstler J. Performance evolution of newcomers in large-scale distributed software projects: an industrial case study. Proceedings of the 14th International Conference on Global Software Engineering (ICGSE 2019), Montreal, Canada, 2019; 1-11.
- McCarthy S, O'Raghallaigh P, Fitzgerald C, Adam F. Towards a framework for shared understanding and shared commitment in agile distributed ISD project teams. Proceedings of the 27th European Conference on Information Systems (ECIS 2019), Stockholm & Uppsala, Sweden, 2019; Research Papers. 83.
- Gill AQ. Distributed agile development: applying a coverage analysis approach to the evaluation of a communication technology assessment tool. *Int J e-Collab*. 2015;11:57-76.
- Uludag Ö, Nägele S, Hauder M. Establishing architecture guidelines in large-scale agile development through institutional pressures: a single-case study. Proceedings of the 25th Americas Conference on Information Systems (AMCIS 2019), Cancun, Mexico, 2019; 1-10.
- Ambler S. Agile data: Agile enterprise architecture. Web page. 2014. Accessed 20 September 2019. <http://www.agilemodeling.com/essays/communication.htm>
- Gable GG. Integrating case study and survey research methods: an example in information systems. *Eur J Inf Syst*. 1994;3:112-126.

32. Venkatesh V, Brown SA, Bala H. Bridging the qualitative-quantitative divide: guidelines for conducting mixed methods research in information systems. *MIS Quarterly*. 2013;37:21-54.
33. Miles MB, Huberman AM. *Qualitative Data Analysis: An Expanded Sourcebook*. Sage Publications; 1994.
34. Alzoubi YI, Gill AQ. An agile enterprise architecture-driven model for geographically distributed agile development. In: Vogel D, Guo X, Linger H, Barry C, Lang M, Schneider C, eds. *Transforming Healthcare Through Information Systems*. Springer International Publishing; 2016:63-77.
35. Gill AQ. Towards the development of an adaptive enterprise service system model. In Proceedings of the 19th Americas Conference on Information Systems (AMCIS 2013), Chicago, Illinois, 2013, 1-9.
36. Ross JW, Weill P, Robertson D. *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution*. 1st ed. Harvard Business Press; 2006.
37. Federation of Enterprise Architecture Professional Organizations. *A Common Perspective on Enterprise Architecture*. White paper. 2018; 1-12. Accessed 13 October 2023. <https://feapo.org/wp-content/uploads/2018/10/Common-Perspectives-on-Enterprise-Architecture-Final-1-copy.pdf>
38. Software Engineering Institute. ND. Software architecture. Web page. 2022. Accessed 13 October 2023. <https://www.sei.cmu.edu/our-work/software-architecture/>
39. AgileManifesto. Manifesto for agile software development. Web page. Accessed 20 September 2020. <http://www.agilemanifesto.org/2001>
40. Stapleton J. *DSDM: The Method in Practice*. Addison-Wesley Longman Publishing Co.; 1997.
41. Highsmith J. *Agile Project Management: Creating Innovative Products*. Pearson Education; 2009.
42. Cockburn A. *Agile Software Development: The Cooperative Game*. Addison-Wesley: Harlow; 2007.
43. Ambler SW. Agile enterprise architecture: Oxymoron or savior? PowerPoint presentation, Accessed 13 October 2023. <https://www.agilealliance.org/wp-content/uploads/2016/01/Ambler-Agile-EA.pdf.2012>
44. Khan AA, Shameem M. Multicriteria decision-making taxonomy for DevOps challenging factors using analytical hierarchy process. *J Softw Evol Process*. 2020;32:e2263.
45. Alzoubi YI, Gill A. An agile enterprise architecture driven model for geographically distributed agile development. In Proceedings of the 24th International Conference on Information System Development (ISD 2015), Harbin, China, 2015.
46. Shameem M, Nadeem M, Zamani AT. Genetic algorithm based probabilistic model for agile project success in global software development. *Appl Soft Comput*. 2023;135:109998.
47. Chow T, Cao D-B. A survey study of critical success factors in agile software projects. *J Syst Softw*. 2008;81:961-971.
48. Aladwani AM. An integrated performance model information systems projects. *J Manag Inf Syst*. 2002;19:185-210.
49. Lee G, Xia W. Toward agile: an integrated analysis of quantitative and qualitative field data. *MIS Quarterly*. 2010;34:87-114.
50. Ammad G, Janjua UI, Madni TM, Cheema MF, Shahid AR. An empirical study to investigate the impact of communication issues in GSD in Pakistan's IT industry. *IEEE Access*. 2019;7:171648-171672.
51. Gill AQ, Bunker D. Towards the development of a cloud-based communication technologies assessment tool: an analysis of practitioners' perspectives. *Vine*. 2013;43:57-77.
52. Sungkur RK, Ramasawmy M. Knowledge4Scrum, a novel knowledge management tool for agile distributed teams. *Vine*. 2014;44:394-419.
53. Karsten H, Lyytinen K, Hurskainen M, Koskelainen T. Crossing boundaries and conscripting participation: representing and integrating knowledge in a paper machinery project. *Eur J Inf Syst*. 2001;10:89-98.
54. Star SL. This is not a boundary object: reflections on the origin of a concept. *Sci Technol Human Values*. 2010;35:601-617.
55. Dodgson M, Gann DM, Salter A. "In case of fire, please use the elevator": simulation technology and organization in fire engineering. *Organ Sci*. 2007;18:849-864.
56. Abraham R. Enterprise architecture artifacts as boundary objects—a framework of properties. Proceedings of the 21st European Conference on Information Systems (ECIS 2013), Paper 120, Utrecht, Netherlands, 2013.
57. Canat M, Català NP, Jourkovski A, Petrov S, Wellme M, Lagerström R. Enterprise architecture and agile development: friends or foes? Proceedings of the 22nd International Enterprise Distributed Object Computing Workshop (EDOCW 2018), Stockholm, Sweden, 2018, 176-183.
58. Kaddoumi T, Watfa M. A foundational framework for agile enterprise architecture. *Int J Lean Six Sigma*. 2022;13:136-155.
59. Sararuch S, Wannapiroon P, Nilsook P. Dimensions of agile enterprise architecture. Proceedings of the Invention, and Innovation Congress: Innovative Electricals and Electronics (RI2C), Bangkok, Thailand. IEEE, 2022, 304-309.
60. van Wessel RM, Kroon P, de Vries HJ. Scaling agile company-wide: the organizational challenge of combining agile scaling frameworks and enterprise architecture in service companies. *IEEE Eng Manage Rev*. 2023. doi:10.1109/EMR.2023.3277128
61. Vidgen R, Wang X. Coevolving systems and the organization of agile software development. *Inf Syst Res*. 2009;20:355-376.
62. Martini A, Pareto L, Bosch J. Communication factors for speed and reuse in large-scale agile software development. Proceedings of the 17th International Software Product Line Conference (ISPLC 2013), Tokyo, Japan, 2013; 42-51.
63. Cataldo M, Herbsleb JD, Carley KM. Socio-technical congruence: a framework for assessing the impact of technical and work dependencies on software development productivity. Proceedings of the 2nd ACM-IEEE international symposium on Empirical software engineering and measurement (ISESEM 2008), Kaiserslautern, Germany, 2008; 2-11.
64. Batra D. Modified agile practices for outsourced software projects. *Commun. ACM*. 2009;52:143-148.
65. Sauer J. Architecture-centric development in globally distributed projects. In: Šmite DNM, Ågerfalk P, eds. *Agility Across Time and Space*. Springer; 2010:321-329.
66. Helquist JH, Deokar A, Meservy T, Kruse J. Dynamic collaboration: participant-driven agile processes for complex tasks. *ACM SIGMIS Database*. 2011;42:95-115.
67. Van Waardenburg G, Van Vliet H. When agile meets the enterprise. *Inf Softw Technol*. 2013;55:2154-2171.
68. Green R, Mazzuchi T, Sarkani S. Understanding the role of synchronous & asynchronous communication in agile software development and its effects on quality. *J Inf Technol*. 2010; 21:8.
69. Bass L, Clements P, Kazman R. *Software Architecture in Practice*. 3rd ed. Addison-Wesley; 2013.
70. Alzoubi YI, Gill AQ. The critical communication challenges between geographically distributed agile development teams: empirical findings. *IEEE Trans Prof Commun*. 2021;64:322-337.
71. Yin RK. *Case Study Research: Design and Methods*. Sage Publication: Thousands Oaks; 2009.
72. Atlassian. What is Jira used for? 2020.
73. Guo H, Li J, Gao S, Smite D. Agile enterprise architecture by leveraging use cases. Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2021), Online Streaming, 2021; 501-509.
74. Bruls WA, van Steenberg M, Foorthuis R, Bos R, Brinkkemper S. Domain architectures as an instrument to refine enterprise architecture. *Commun Assoc Inf Syst*. 2010;27:517-540.
75. Cammin P, Heilig L, Voß S. Assessing requirements for agile enterprise architecture management: a multiple-case study. Proceedings of

- the 54th Hawaii International Conference on System Sciences, Grand Wailea, Maui, Hawaii, 2021; 6007-6016.
76. Gill AQ. *Adaptive Cloud Enterprise Architecture*. World Scientific; 2015.
 77. Ren M, Lyytinen KJ. Building enterprise architecture agility and sustenance with SOA. *Commun Assoc Inf Syst*. 2008;22:74-87.
 78. Hummel M, Rosenkranz C, Holten R. The role of social agile practices for direct and indirect communication in information systems development teams. *Commun Assoc Inf Syst*. 2015;36:273-300.
 79. Brewer EC, Holmes TL. Better communication = better teams: a communication exercise to improve team performance. *IEEE Trans Prof Commun*. 2016;59:288-298.
 80. Simhadri RS, Shameem M. Challenges in requirements gathering for agile software development. Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, Oulu, Finland, 2023; 406-413.
 81. Mishra D, Mishra A. Simplified software inspection process in compliance with international standards. *Comput Stand Interfaces*. 2009;31:763-771.
 82. Mishra A, Abdalhamid S, Mishra D, Ostrovska S. Organizational issues in embracing agile methods: an empirical assessment. *Int J Syst Assur Eng Manag*. 2021;12:1420-1433.
 83. Özkan D, Mishra A. Agile project management tools: a brief comparative view. *Cybern Inf Technol*. 2019;19:17-25.
 84. Yu L, Mishra A. Risk analysis of global software development and proposed solutions. *Automatika*. 2010;51:89-98.
 85. Mishra A, Akman I. Information technology in human resource management: an empirical assessment. *Public Pers Manag*. 2010;39:271-290.

How to cite this article: Alzoubi YI, Mishra A. Enterprise architecture contribution in distributed agile software development. *Systems Engineering*. 2023;1-15. <https://doi.org/10.1002/sys.21739>