Can Agile Enterprise Architecture be Implemented Successfully in Distributed Agile Development? Empirical Findings

Yehia Ibrahim Alzoubi [1]* and Asif Qumer Gill [2]

[1] American University of the Middle East; yehia.alzoubi@aum.edu.kw

[2] University of Technology Sydney; asif.gill@uts.edu.au

* Correspondence: yehia.alzoubi@aum.edu.kw

Abstract

A potential solution to the high failure rate in distributed agile development and enhance the success of projects is through implementing agile enterprise architecture, though the success is still to be established. The present paper empirically investigates the gap, by defining the role and commitment of implementing agile enterprise architecture on distributed agile development. The data were collected by interviewing 12 key team members and observing four team meetings over 2 months and analyzing using thematic analysis. The present study suggests that implementing agile enterprise architecture is possible in distributed agile development and may have a positive impact on project success. However, many questions demand further investigation.

Keywords: AEA implementation, AEA management, AEA principles, AEA role, case study

---

1. Introduction

The geographically distributed agile development (GDAD) has been adopted by many agile software development companies due to many management-related advantages (Alzoubi, Gill et al. 2016; Hoda, Salleh et al. 2017; Shameem, Kumar et al. 2020; Turnu, Melis et al. 2006). GDAD

involves more than one team distributed in different physical locations or different time zones (Alzoubi, Gill et al. 2016; Ammad, Janjua et al. 2019; Pillai, Pundir et al. 2012). However, by the virtue of the same, GDAD is confronted with a slew of issues such as communication, coordination, commitment, time zone related issues, and cultural differences issues among GDAD teams, which result in the high failure rate (Alzoubi, Gill et al. 2016; Ammad, Janjua et al. 2019; Britto, Smite et al. 2019; McCarthy, O'Raghallaigh et al. 2019; Sharp and Robinson 2008; Smite et al. 2019; Zaitsev, Gal et al. 2020). Collocated (i.e., does not adopt GDAD) team members continuously meet and communicate (face-to-face and informally) project requirements (technical and business) (Amin and Horowitz 2008; Lee and Chen 2019; Ribeiro and Fernandes 2010; Shameem, Kumar et al. 2018). This decreases the need for documentation and helps in meeting these requirements. However, in the GDAD environment, the chance of meetings and communication is very low due to the above challenges (Alzoubi and Gill 2021; Nundlall and Nagowah 2021). Despite the teams' autonomy recommended for agile development, the GDAD team should be aligned and collaborated (Horlach, Drechsler et al. 2020; Stadler, Vallon et al. 2019; Yadav 2016). Therefore, organizational mechanisms are required to manage dependencies between teams and services (Acharya 2019; Chikhale and Mansouri 2015; Horlach, Drechsler et al. 2020). The absence of such mechanisms may lead to a high failure rate through unjustifiable decisions, redundant, inconsistent, conflicting solutions (Dingsøyr, Moe et al. 2018; Horlach, Drechsler et al. 2020).

One of the potential solutions that may decrease the effect of the above challenges and provide such a mechanism was by implementing the agile or adaptive enterprise architecture (AEA) shared view or AEA management (Alzoubi and Gill 2020; Horlach, Drechsler et al. 2020; Korhonen, Lapalme et al. 2016). This view may represent a common ground that keeps all distributed teams

on track by providing essential information related to different aspects of business architecture, strategy view, technology architecture view, and solution architecture view (Alzoubi and Gill 2020). Consequently, the objectives will be well understood by GDAD members, overall mission, and vision of the business and project (Alzoubi and Gill 2020), and thus all the teams focus on the common goals as an integrated system and avoiding duplication of work, inconsistency, and redundancy (Britto, Smite et al. 2019; Korhonen, Lapalme et al. 2016). Hence, the AEA view may increase the success rate of GDAD projects (Alzoubi and Gill 2020; Canat, Català et al. 2018; Korhonen, Lapalme et al. 2016).

Most of the previous studies about AEA have investigated the possibility of implementing AEA in agile development (e.g., (Canat, Català et al. 2018; Uludag, Nägele et al. 2019)) due to the conflicts between agile development business teams and technical teams (Hanschke, Ernsting et al. 2015). While the business team appreciates the value and role of AEA, technical teams assume that AEA contradicts the soul of agile development which is a team and communication-oriented rather than a heavy documentation-oriented approach (Uludag, Kleehaus et al. 2019). However, unlike traditional enterprise architecture that follows the top-down approach and focuses on long term objectives, AEA principles and artifacts should follow the spirit of the agile development and be created based on the all-team sharing concept in order to achieve the commitment and expected value, especially from distributed technical teams (Gill, Henderson-Sellers et al. 2018; Hanschke, Ernsting et al. 2015). In other words, the suitability and value of implementing AEA in GDAD are still not clear.

Traditional enterprise architecture procedures are unable to govern agile teams with moderate effort. Enterprise architecture activities should not just focus on enforcement, but also on influencing people by educating, legitimizing, and socializing them (Gill 2015). As a result, an

influence-centric perspective of enterprise architecture is required to complement the enforcement-centric approach, which transfers decision-making power to agile teams and therefore mitigates agile teams' resistance to enterprise architecture governance initiatives (Uludag, Nägele et al. 2019). Even though the number of firms employing agile techniques is growing, there is a scarcity of academic study on how to build architecture principles in an agile way. Accordingly, empirical studies about the successful implementation of AEA in GDAD are scarce (Uludag, Kleehaus et al. 2019). Therefore, the present study fills the gap and empirically investigates how a successful GDAD organization creates the AEA principles collaboratively between all stakeholders, implement AEA, and use it among GDAD teams. The following research questions are addressed by this paper:

RQ1: How is AEA implemented in GDAD?

RQ2: What is the role of AEA in GDAD?

The first question is to understand how AEA concepts are developed and disseminated in big, effective GDAD companies. The second question examines the effect of AEA on project success. We utilized an in-depth exploratory case study to answer these research issues. This qualitative approach provides a better understanding of the research phenomena (Yin 2009). Our hypothetical company was a large (i.e., 39-person) software development firm with its headquarters in Sydney and two other teams in India and China. Over two months, we collected data through observation of the primary team in Sydney and semi-structured interviews. Interviews were conducted with 12 important individuals in the three different locations. The data collected were then analyzed using thematic analysis (Mayring 2010).

The present investigation contributes to the successful implementation of AEA in the GDAD environment, how AEA may be implemented, and what are the difficulties faced during

implementation. AE architect and architecture owner in distributed teams worked collaboratively with the business team to create the AEA principles. The findings of this paper show that AEA can be successfully implemented in GDAD. Moreover, the paper provides empirical evidence of the positive impact of AEA on delivering successful GDAD projects.

The remainder of this work is structured as follows. Section 2 presents the background and related work. Section 3 discusses the research method used in this paper to collect and analyze the data. The findings of this paper are discussed in Section 4. The research implications, as well as limitations, are discussed in Section 5. Section 6 concludes this paper.

2. Background and Related Work

2.1. Agile Enterprise Architecture

Traditionally, EA is defined as the logic that organizes business operations and IT assets, representing the company's current model's integration and standardization needs (Ross, Weill et al. 2006). It follows the top-down approach and centralized process by enforcing predefined architectural standards involving modeling, planning, and control (Hauder, Roth et al. 2014). This centric-control management does not work in the autonomous agile environment and it is against the agile technical teams' temporary drives to satisfy the business requirements (Kulak and Li 2017; Uludag, Nägele et al. 2019). This type of management It is possible, according to this type of management, that this will lead to a lack of attention to long-term architectural enhancements (Buchan, Talukder et al. 2019; Dingsøyr, Moe et al. 2018).

Agile approaches depend on emergent architecture without direct structure, which is effective in the collocated agile team structure (Ali Babar, Ihme et al. 2009). Accordingly, AEA is defined as a blueprint that specifies the entire behavioral, architectural, technical, sociological, and facility components of an enterprise's operational environment that share similar aims and concepts, as

well as the capacity to be flexible, responsive, quick, learn, and lean (Gill 2013). According to the Agile Manifesto (2001), during software development, an adaptable quick plan is more effective and can respond flexibly to changes in requirements than the up-front detailed project plan with heavy documentation. It's critical to comprehend both the system functionality and the system architecture that will be employed (Highsmith 2009). Highsmith (2009) explains that agility should strike a balance between flexibility, thorough documentation, structure, and some upfront planning. However, in GDAD, a certain level of architectural planning is required to ensure that distributed teams are aligning on the business goals (Batra, Xia et al. 2010; Leffingwell 2007). The enormous architecture must be split down into smaller sections in GDAD projects so that they may be constructed and evaluated progressively (Highsmith 2009; Sauer 2006). Moreover, while agile development promotes a self-organized team, not like collocated teams, GDAD may require a level of control in order to keep all distributed teams on track, especially, if the GDAD project includes many distributed teams (Dreesen, Diegmann et al. 2020; Highsmith 2009). AEA in GDAD provides governance requirements, aligning individual project strategies, and guiding agile technical teams through technical and business roadmaps (Uludag, Kleehaus et al. 2019). However, AEA management is not a centric-control approach, like traditional EA. Instead, it’s a teamwork approach that should involve all business, architecture, and development or technical teams. In recent years, several studies have investigated how AEA is created and shared among all teams, or what is known by many authors as AEA management (e.g., (Hauder, Roth et al. 2014; Horlach, Drechsler et al. 2020; Uludag, Nägele et al. 2019)). Several studies have emphasized following the AEA management practices when creating AEA principles. These studies, as well as the studies that investigated the impact of AEA on GDAD, are discussed here.

2.2. Suitability of AEA in GDAD

Uludag, Nägele et al. (2019) investigated the possibility and suitability of implementing architecture principles in GDAD following a mixed approach of data collection using both interviews and survey methods. Their findings suggest a positive impact of AEA on the success of GDAD, though with caution that the traditional top-down approach (i.e., forcing technical members such as developers to use AEA) used in traditional enterprise architecture will not be working in GDAD; rather the approach applied should be bottom-up (i.e., including all members in creating architecture principles) in order to get the highest commitment, which reflects positively on GDAD project success. Moreover, to study the applicability of combining agile development and enterprise architecture, Canat, Català et al. (2018) interviewed agile development practitioners in five organizations and their findings suggest that enterprise architecture can be applied to agile software development (Canat, Català et al. 2018). Another study investigated the effect of AEA on GDAD performance and communication by survey technique (Alzoubi and Gill 2020) and reported that AEA has a positive relationship between GDAD communication and AEA. Also, the results showed a positive impact of AEA on most performance aspects (i.e., quality, functionality, and on-budget completion), while there was no significant relationship between on-time completion and AEA (Alzoubi and Gill 2020).

Alzoubi, Gill et al. (2015) introduced the AEA Framework explaining how GDAD teams can use the AEA repository or view to enhance communication and project success in GDAD. This repository has AEA artifacts, available online for all distributed teams acting as a common ground or a boundary among distributed teams (Alzoubi, Gill et al. 2015). It provides all members with their needs related to a project such as customer and business requirements, time frame, quality and functionality requirements, and so on (Alzoubi and Gill 2020; Kornstadt and Sauer 2007).

2.3. Principles of Implementing AEA in GDAD

Horlach, Drechsler et al. (2020) proposed six principles to enable architectural thinking in agile organizations based on exploratory case analysis. Applying these six principles may change the AEA management. These principles are: create an architect around the business ecosystem, facilitate value-oriented architecture assistance, continuously external views, facilitate long-term directions for constant architecting, enable local stakeholders to make timely architectural decisions, and make architecture visible to all stakeholders.

Uludag, Kleehaus et al. (2019) reported the expectations about implementing AEA in large agile development. The authors found that AEA management is guided by threefold: (1) communication between AEA owner and other agile or technical teams is driven by solution architectures which increases the misunderstanding and information loss, (2) interviewees request more support from AEA owner in the future, and (3) the role of AEA is getting bigger with multiteam included which add more pressure on AEA to meet architectural issues from different teams.

To satisfy AEA supervision, agile and lean approaches were used in Bente, Bombosch et al. (2012) who introduced a collaborative AEA that includes the way of how to initiate AEA, how to enable the collaboration between different stakeholders, and how to manage AEA against some issues such as enforcing rules and dealing with the quick changes in requirements. Hauder, Roth et al. (2014) introduced an AEA management function. This function consists of three phases: motivation, collection of information, and analyzing the information and feedback collected about AEA. In the first phase, AEA value should be promoted among all stakeholders included in the GDAD project; business team and development (technical) teams. In the second phase, information about AEA principles should be collected from all stakeholders and formed into stakeholders-specific models and concepts and then communicate these models with all

stakeholders to get their feedback. This phase required heavy communication among AEA owners or teams and different stakeholders. In the third phase, the AEA principles may be refined or adjusted based on the analysis of the feedback from phase 2.

This paper takes the view provided by the above authors that AEA in GDAD should be achieved by a collaborative approach between all agile stakeholders. That is, AEA principles should be created following the AEA management process. Accordingly, this paper follows the AEA management practices introduced by Hauder, Roth et al. (2014) and Uludag, Kleehaus et al. (2019) to evaluate the implementation process in the case under study. This was important to judge the claim that this company successfully followed AEA in GDAD.

This paper looks into a big GDAD organization that, according to participants, completed numerous successful projects. This study adds to the body of knowledge by examining how such organizations create and employ AEA successfully. It also looks into the favorable influence of AEA on the success of the GDAD project. This is critical for GDAD businesses since many GDAD projects cannot properly integrate AEA or employ AEA in the most beneficial method for their GDAD project, and many GDAD projects continue to fail.

3. Research Design

3.1. Case Study Research

To investigate qualitatively, a case study was employed that is an empirical study of a real-life phenomenon (case) that focuses on observing, reconstructing, and evaluation (Yin 2009) with the understanding that a single case study helps in gaining an insight into the composite cases and interpersonal interactions, as well as investigating and evaluating a contextual phenomenon (Pedrycz 2006; Yin 2009). The subjects were 12 interviewees. The organization was purposefully selected because it implements AEA in its GDAD approach. The interviews were conducted from

June 2018 to July 2018. The participants in this study were chosen based on industry ties established by the researchers.

3.2. Research Context

3.2.1. The case organization and the teams

A large-size Sydney-based organization specialized in delivering new applications having a team of 39 members was chosen for the present investigation. To preserve anonymity, the name of the organization and interviewees was coded (i.e., SUNC and the participants from "A", "B", up to "L"). The demography of the interviewees is given in Table 1. The members are segregated into 3 teams: The first 18-member core team (i.e., on-site team) or the main team located in Sydney has the program manager, integration manager, enterprise architect, tester leader, business architect, Scrum master leader, product owner, developers, reviewers, and analysts. Sydney team is responsible for the assignments like collecting and assigning tasks, configuration, and integration. The second team, located in China has 16 members comprising of Scrum Master, solution architect, and developers. The third team located in India has 5 members consisting of a solution architect, a tester leader, and 3 testers. The China and India teams are identified as "off-shore" teams.

In July 2016, SUNC initiated a project named "pensioner banking system" (PBS) and the participants described it as a big project spread over 2 years span. SUNC adopted an agile approach (i.e., Scrum method) in 2010 and GDAD since 2015.

Table 1. Participant demographics

Participant	Role	Location	Experience (Years)	Interview Method

PA	Program manager	Sydney	10	Face-to-face
PB	Enterprise architect	Sydney	9	Face-to-face
PC	Product Owner	Sydney	7	Face-to-face
PD	Scrum Master leader	Sydney	8	Face-to-face
PE	Integration manager	Sydney	11	Face-to-face
PF	Tester leader	Sydney	6	Face-to-face
PG	Iteration manager	Sydney	8	Face-to-face
PH	Developer	Sydney	4	Face-to-face
PI	Solution architect	India	7	Skype
PJ	Tester	India	6	Skype
PK	Scrum Master	China	4	Skype
PL	Technology architect	China	5	Skype

3.2.2. The PBS Process and Team roles

Sprint information is frequently communicated on the walls in Sydney's site in the form of artifacts such as Sprint goals, drawings of the project plan, completed tasks, and so on. The three teams use “JIRA” - a Scrum-supporting tool created by the Atlassian Company that enables time tracking capabilities, and real-time performance reports in GDAD. In addition, the code is shared and fine-tuned amongst the teams via an internet repository. Various stakeholders, including the enterprise architect, program manager, delivery manager, and client, identify and evaluate business needs. Stakeholders used to attend the end of the Sprint for review through electronic tools. The Sprint retrospective is facilitated by the Sydney team.

The enterprise architect, located in Sydney, shares the blueprint of the whole project at the onset, with the architect’s team distributed in China and India. The enterprise architect also reviews the business requirements with the architect’s team after each Sprint planning and after testing the

new release. The team of architects (i.e., enterprise architect, solution architects, business architect, technology architect, and infrastructure architect) has separate communication tools used in SUNC to discuss architecture principles, progress, conflicts, and so on.

3.3. Data Collection and Analysis

Semi-structured interviews were used to gather data lasting from 50 to 70 minutes. During each interview, we employed both note-taking and voice-recording techniques. Permission to record audio was granted at the start of each interview. A case study approach was employed to maximize research dependability and eliminate any bias between the researcher and the respondents (Yin 2009). At the beginning of the interview, each interviewee was asked to describe his/her role and responsibilities in the PBS project. The interviewees were then asked the following. The pre-defined questions were used aiming to uncover how AEA is defined, shared, and used among distributed teams:

- How the AEA is defined in the context of the PBS project?
- How the principles of AEA are defined and created in the PBS project?
- What are the perceptions of using AEA views that may affect the success of PBS?

Moreover, observation of the team members about the work technique was also recorded while conducting interviews in Sydney's location and field notes were collected. A total of four meetings were observed for the main team located in Sydney: two Sprint planning meetings (around 80 minutes each) and two stand-up meetings (around 40 minutes each).

A thematic analysis that identifies, analyzes, and reports themes (patterns) within data, was conducted across the 12 interviews and observation notes (Mayring 2010). We identified and compared important comments within and across participants. There were several rounds of contrasting and comparing interview remarks to achieve the key insights and shared patterns. The

data were synthesized and coded first against the predefined proposition's elements (e.g., defining AEA, Sharing AEA, AEA commitment), based on the definitions and agile manifesto principles of AEA. The codes resulted in relabeling, dropping, or merging, and subsequently, the codes were translated into themes (Mayring 2010).

4. Findings

This paper used the AEA management practices of Hauder, Roth et al. (2014) and Uludag, Kleehaus et al. (2019) to evaluate the level of collaboration and agile practices embedded in enterprise architecture in GDAD firms. Accordingly, the focus of this paper is on the collaboration of creating and sharing AEA between the business team (e.g., top management, project manager, program manager), AEA team (AEA architect, solution architect, or AEA owner), and technical teams (e.g., developers, testers, integration team). Therefore, we answer the RQ1 and RQ2 by identifying AEA definition, AEA communication, AEA models, AEA availability, the involvement of all teams in creating AEA principles, AEA support to technical teams, feedback of technical teams to the AEA team, and the impact of AEA on GDAD project success. Table 2. Summarizes the findings of this study.

Table 2. Key findings from interviews

Research Question	Findings
RQ1: AEA management practices	• Agile development concepts and abilities should be embedded in AEA. (PB, PD, PE, PL)
	• Communicating AEA is a hard task and required big efforts. (PA, PE, PG)
	• AEA principles can be represented in diagrams, business capability maps, communication models, data models, and interface models. (PB, PI, PL)

	• AEA principles are made available in a common repository to all distributed teams. (PB, PI, PL)
	• All stakeholders should be involved in creating AEA. (PA, PB, PC, PG, PI)
	• The role of the architecture owner is important in creating and sharing AEA with the other agile team members. (PI, PJ, PK, PL)
	• The feedback was based on the regular meetings with all architecture teams. (PB, PI, PL)
RQ2: AEA impact on GDAD	• By sharing project goals and scope, AEA will help keep GDAD teams on track, allowing projects to be completed on time, on budget, and with high quality. (PG, PL)
	• AEA can estimate results, and architects can assist in exchanging AEA objects and resolving disputes between GDAD teams. (PB, PG)
	• An AEA view saved in a repository and accessible by all team members can help to boost communication quality. (PG, PL)

4.1. RQ1 – AEA Management Practices

Although the initial AEA is constructed and solidified in the main team at Sydney (community of practice), the SUNC strategy seeks to include all teams in the implementation of architectural principles and guidelines. Distributed teams are given more trust and responsibility with this strategy. Figure 1 shows the process of developing and sharing AEA principles in SUNC. The community discusses architectural issues in general as well as principles and guidelines. This process adds to the generic architecture principles process by considering the agile teams' extra viewpoint. To keep the decision-making ability effective, one representative for each team should join the respective community (i.e., architecture owner).

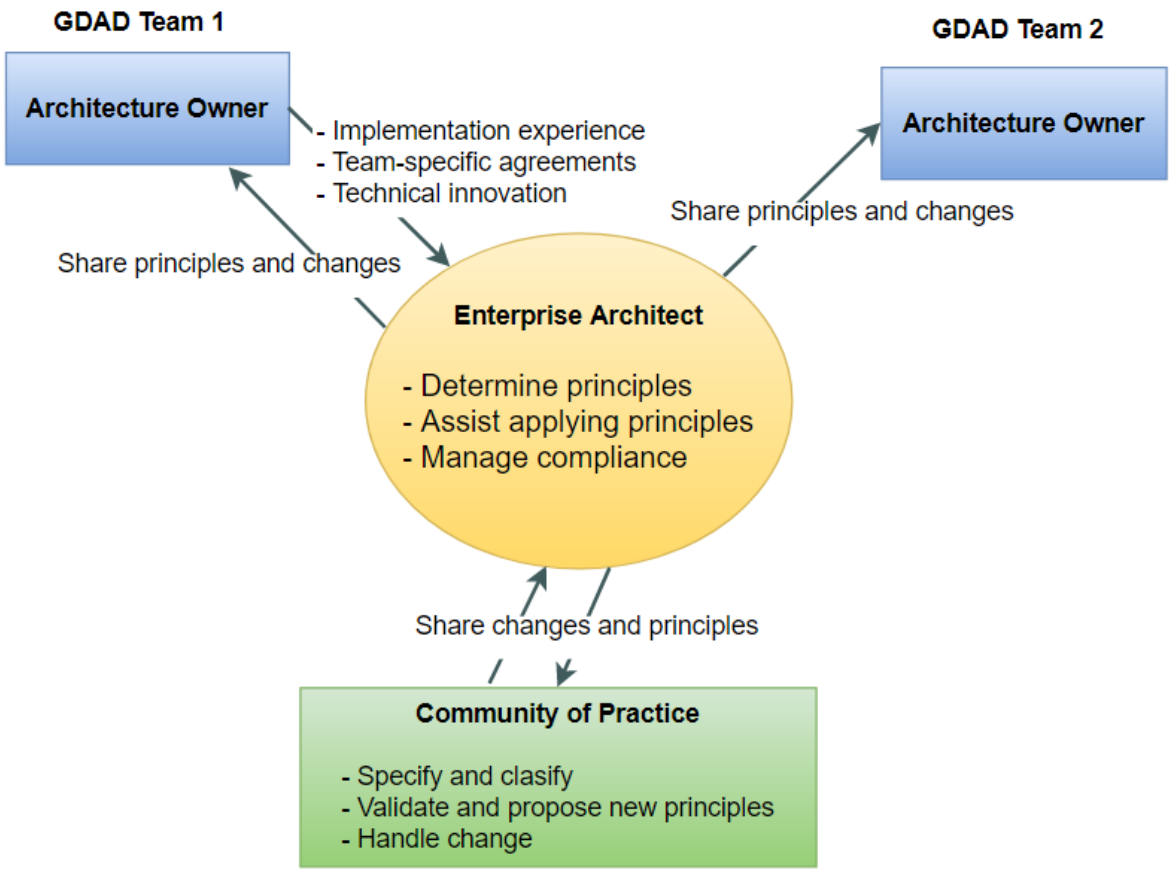

Figure 1. AEA creation and sharing process

The principles are defined by the community of practice and their fulfillment must be measurable. Where these principles are too vague or unspecific to direct practical execution, or where their fulfillment requirements cannot be specified, this is required. Architectural principles and criteria are applied by technical staff following the architecture owner's approval. Enterprise architect provides an overview of guidelines and manages the compliance as well as share the technical recommendation with the community of practice and the AEA principles with the technical teams.

4.1.1. AEA Definition

Various GDAD firms use various definitions for AEA. There are a variety of EA frameworks accessible in the literature, but there is no consensus on which framework should be utilized or

regarded as the best option for the business (Abdallah, Abran et al. 2021). As a result, before an organization chooses a framework, all relevant frameworks should be assessed against a set of criteria (attributes), and the best framework should be chosen based on the results (Abdallah, Abran et al. 2021).

Therefore, in order to see the level of agility applied to the AEA at SUNC, it was important to reveal how different interviewees define AEA. There was an agreement among all interviewees that agile development skills and concepts should be embedded in AEA. They described AEA as a method for capturing and reflecting intended change, as well as the extent to which that transition has been achieved (PB, PL). Moreover, they said that in the same way as agile methodologies focus on flexibility in response to change, AEA should do the same (PD, PE). The aim of AEA, according to PB, is: "*Focus on the flexibility of change. There is an iterative, incremental approach to building your EA, and that teams are key stakeholders in the refinement of the Product Backlog.*" AEA should "*respond to changing requirements*", as described by PL.

4.1.2. AEA Communication

Infrastructure architects, solution architects, technology architects, business architects, and other architects may be employed by big GDAD enterprises. Architecture owner or architect (i.e., architecture representative such as enterprise architect or architect) role was important in the PBS project (PA, PE). In certain situations, the function of the AEA owner is not considered significant (Cammin, Heilig et al. 2021). The core of AEA was congealed by the main team (Sydney) in the initial iteration and shared with all distributed architecture owners (PB). The interviews at the Sydney location justified this practice due to the fact that in some projects, new distributed teams or members may join the project (PA). This way; distributed developers could start upon a solid architecture (PG). However; off-shore (China and India) teams did not like following AEA (PE).

This maybe was due to the AEA constraints and requirements, according to PE who explained: *"They think that they are not going to deliver until they meet all these requirements. It scares them."*

Communicating EA was reported to be a hard task and required big efforts. This was reported by PC as: *''The main worry of using AEA is communicating it."* Enterprise architect may have to visit teams in different continents in order to discuss and express the AEA perspectives. PB expressed this as: *"Whenever any team needs clarity in some stage of the project, I travel and meet with the architecture owner of that team and discuss the state."*

4.1.3. AEA Model (View)

Architects or architecture owners should offer technical teams with diagrams, business capability maps, and other supplementary information such as applications, technologies, and infrastructure components as key architectural models (Uludag, Nägele et al. 2019). AEA view was developed and promoted using plain, straightforward language that was easy to comprehend by all team members (PB). This view included different models (e.g., diagrams, business capability maps, communication models, data models, interface models) (PB, PI, PL). The online team repository offered all teams examples, checklists, and common standards (PE, PG). JIRA tool was very helpful to share the AEA view among distributed members. JIRA tool along with the central team repository provided all teams with EA view. PE explained: "*It is better to be more general when talking to off-shore teams and don't go for that much detail."* The apparent utility of AEA models to major stakeholders has been reported, such as the creation of good information that necessitates a great focus on the proposed stakeholders (PA).

4.1.4. AEA Availability

The following are the requirements of given AEA models from technical agile teams: availability, binding force, quality, and relevance that provides value, applicability, and degree of detail (Uludağ, Nägele et al. 2021). Following the dissemination and communication of the AEA viewpoint to both business and technical architects, the AEA view was made available on the online team repository, which can be accessed anytime and anywhere by all teams (e.g., development teams, testers, analysts) (PB, PC). Then it was the role of the architect (e.g., architecture owner) in each team to communicate that view with his/her team members (PI, PL). PI explained: *"AEA artifacts are hosted in an online repository at the start of the project after all teams have been informed of the AEA view. Anyone who requires it may go look at it, which provides guidance to everyone."*

4.1.5. Involvement of All Stakeholders in Creating AEA Principles

Rather than guiding too much and in detail, the AEA should promote the incorporation of bottom-up innovations in big forms. Rather than directing a project on a too technical level, AEA should be supported to fit inside the larger picture and be connected with the company objectives. To encourage cooperation and prevent overwhelming stakeholders, Gill preferred evolutionary issue solutions over rigidly blueprinting the future (Gill 2015). SUNC claimed that their AEA view was cross-team development, which expands the definition of conventional EA, which has a technical emphasis, to include a broader view of EA (PA, PB, PC). PC stated: *"there is a need to change from the dominating blueprint-focused EA to a relational-focused AEA."* As the project progressed, AEA elements (view) were reported to be integrated into daily updates that all participants were aware of and shared on developing solution design (PG, PI). PI reported: *"Architects cannot accomplish EA aims by themselves. This typically requires an interpersonal tactic based on mutual understanding and trust and friendship."* Some interviewees claimed that

SUNC follows a “team-product” AEA, which is in contrast to the conventional plan-focused EA where EA is not considered in governance policies and objectives (PB, PE). *“We don’t follow a hierarchical approach in our organization. It is a federated approach”* reported PB.

According to some respondents, the continual integration of the technical and business teams might offer a significant shift in SUNC via strategy development with agile concepts in creating AEA principles (PA, PC). PA reported: “*Planning of an agile portfolio in our company defines the decision on how much to invest is driven by the value that will be added.”*

Through architectural coordination circles, technical teams must be actively participating in AEA creation procedures, facilitating the construction of the to-be AEA (Gill 2014). For agile teams, cooperation between enterprise architects and architectural owners was rated as extremely beneficial by fostering a common understanding of the architecture (PI, PL), guidelines (PB), information, and advice were some of the major benefits listed (PE). Shorter feedback cycles and frequent exchanges were also reported as advantages, leading to the constant improvement of existing guidelines (PD). PI reported that *“People who put it into practice may come up with more practical or superior answers than the architect.”* Moreover, PK reported that *“AEA benefit from close contact with teams because it allows them to better appreciate the issues of developers.”* The involvement of GDAD teams improves their acceptance and motivation, as was mentioned by participants (PD, PH). PH reported that *“the participation of agile teams is critical for acceptance and admittance.”*

4.1.6. AEA Supports to Technical Teams

AEA supports functions and also acts as a facilitator for cross-project collaboration, not just to create a shared vision, but also to share lessons learned from previous projects (Cammin, Heilig et al. 2021). An enterprise architect's general responsibility, for instance, was to convey and

express the AEA's point of view (PB, PI, PL). PI reported: *"The role of a solution architect is to describe what should be done to various stakeholders who are unfamiliar with the architect and persuade them that following the architect would result in better advantages."* The enterprise architect at the Sydney location and the solution architects of the two off-shore teams provided the technical teams with the AEA principles as well as the explanation and discussion of these principles when needed (PI).

Architects should help technical teams by delivering architecture concepts and supporting them in putting them into practice. Moreover, in order to realize AEA, technical teams require direct assistance and advice (Korhonen, Lapalme et al. 2016). The solution architects in the off-shore teams helped their teams during the project in solving the misunderstanding related to the customers and business requirements (PK). That is team members in the off-shore teams used to return to the solution architect, from a business and technical standpoint, to assert or obtain a clear image of the given software. PJ reported: *"Team can connect their representative architect anytime to discuss EA view"*. Moreover, the solution architect helped in solving the conflicts among team members and discuss the updates with the enterprise architect (PJ, PL). PL reported that: "*In face-to-face hang meetings, I discuss and clarify the EA perspective with my team on a frequent basis as things progress."*

4.1.7. AEA Feedback

AEA takes into account feedback from technical teams. The architecture principles and guidelines should be revised in light of the feedback. Generating new architectural models and providing revised or new artifacts are greatly welcomed by technical teams (Uludag, Nägele et al. 2019). AEA is not a project *"that is ever completed"*, according to PI. Some interviewees explained that AEA grows in such a way that the future state becomes the new present state, and

it changes AEA's contents and strategies to reflect this (PB, PE). This was based on the feedback from architecture owners (i.e., solution architect) in the off-shore teams as well as the architecture teams in Sydney (PE).

Prototypes are a well-known method of evaluating the viability of ideas to reduce the failure rate (Gill and Maheshwari 2021). The systems can be replaced after all criteria have been met, and resources can be liberated for future EA initiatives (Ghantous and Gill 2019). This may be especially important in the case of critical parts of the EA, including a vital business process or security components. The feedback was based on the regular meetings with all architecture teams (enterprise architect, solution architects, infrastructure architect, business architect, technology architect); such that the feedback from the off-shore teams provided by the solution architect in each of the off-shore teams. Some interviewees added that it's difficult to transition from a one-time event to an annual event, and from a budget-driven process to an agile process that changes the content of EA's management and regulatory activities anytime strategic activities alter the outcome of architect tasks (PA, PB, PG). PB explained: *"What we initially do is examine the prior state, the current state, the final state, and the gaps. Gaps are used as opportunities."* Moreover, through best practices, case studies, and fact-based research, communication platforms and apps such as Wikies were utilized to track the important problem of the failures and successes in reaching the intended future state (PD, PH). PG explained: *"We keep track of iteration and Sprint activity using our communication tool."*

4.2. RQ2 – AEA Impact on GDAD

AEA may be thought of as a way to improve GDAD performance. AEA provides the framework for architecture principles, which can help with consistency in implementation, minimize the number of mistakes, and utilize the same design and language patterns (Kornstadt

and Sauer 2007). The AEA view may offer a work allocation plan, make it easier for GDAD team members to collaborate, and allow for continuous integration across remote teams (Madison 2010).

As was reported by interviewees, the AEA view was utilized to connect and integrate all operations inside SUNC in an efficient manner (PB, PE). Therefore, the AEA view can refer to the whole enterprise, including technology, information, infrastructure, and processes, as well as a specific domain within the organization (PA). Accordingly, the AEA view spans multiple functions and various systems across the organization (PI). They also added that the individual business units may develop securely because of the AEA view's ability to strike the correct balance between IT productivity and business innovation (PA, PB). In addition, the AEA view aided in verifying that IT strategy was aligned with the demands of the enterprise (PB, PE). PB reported: *"It is extremely well aligned, so anytime the business planning cycle occurs, we input our IT needs and then determine what the company wants from a one-year or five-year plan. Then we determine if we need to do it, whether we need to close the gap, or whether we need to establish competency. This is the EA in a nutshell."*

Moreover, it was reported that the AEA view helped to keep all teams on the same page by referring to the AEA view which could be viewed at any moment and displayed the entire picture (PI, PL). The architect's role was important during the development of iterations (PI). PE had uncovered this as: *"EA creates the overall picture and ensures that it is moving in the right direction. We can go back to the EA or the architects if anything really goes wrong and seek their assistance."* The enterprise architect estimated the cost, time, and resources that were available (PA, PB). The solution architects shared these estimations with their teams (PI, PL). This function was characterized by PB as: *"At the portfolio level, the architects' team also supplies performance predictions."*

Cataldo, Herbsleb et al. (2008) developed a socio-technical congruence technique to assess the influence of task dependency alignment on developer coordination. Coordination was shown to be more effective when the alignment was high. SUNC claimed that AEA should be viewed from a wider socio-political perspective (PE). The consumer was the major emphasis of agile development as part of this social structure (PA, PC). PA uncovered this as: *"The AEA should specify how GDAD expects to communicate with their consumers, the methods that will be employed, and how the company will interact with consumers during the development cycle."* Moreover, the AEA view was claimed to disclose quantitative value to retain support and commitment from members of the team and upper management in order to achieve success (PB, PF). PF stated: *"Stakeholders must recognize the importance of our business effort in order to appreciate and support our strategy."*

Agile software development is described as a communication-oriented approach (Alzoubi, Gill et al. 2018). The success of GDAD ventures is directly influenced by the degree of coordination between GDAD teams (Buchan, Talukder et al. 2019; Paul, Drake et al. 2016). Various objects, such as maps, graphs, and tables, may be used to depict an AEA view (PB). The concepts and principles of the AEA can serve as a model and a shared language (PB). This can be regarded as a significant contributor to the coordination among dispersed teams (PB, PG). Some interviewees said they use AEA artifacts for scattered teams to help them communicate more effectively with other distributed members (PG, PI, PL). Participant L reported that *"They just need to look at the maps to keep track. This, I believe, raises the pace at which production teams and business people converse."*

5. Discussion

AEA has been introduced to overcome the shortcomings of traditional EA in the context of agile development. However, the empirical evidence or research about the implementation of AEA is still very rare. The present study evaluates the implementation of AEA in a successful GDAD organization and draws the most essential concepts for AEA in the current context of GDAD. This was done using a qualitative approach through an in-depth case study to answer the two research questions; how AEA is implemented in GDAD (RQ1) and what the role of AEA in GDAD is (RQ2). The contributions to literature and implication for practice, as well as study limitations and future research directions, are explored in the following sub-sections. Figure 2 illustrates the focus and benefits of AEA.

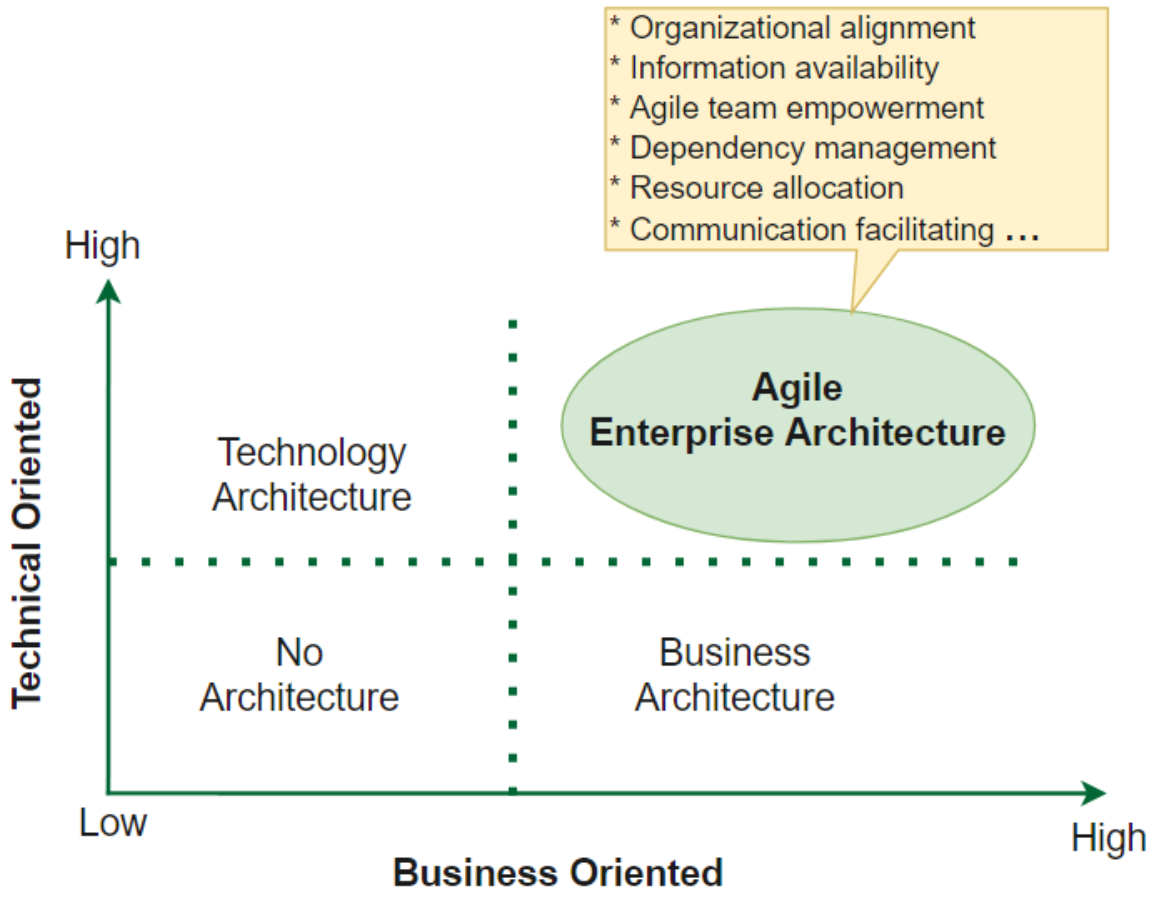

Figure 2. The focus and benefits of agile enterprise architecture

5.1. Contributions to Literature

This study adds to the literature by providing key insights based on empirical research, which indicates that AEA can be successfully implemented in GDAD. This supports the earlier findings of (Hauder, Roth et al. 2014; Medeiros, Santana et al. 2021; Uludag, Kleehaus et al. 2019). However, although the evaluation was conducted based on the AEA management practices provided by (Hauder, Roth et al. 2014; Uludag, Kleehaus et al. 2019), not all practices were provided by these two studies. In this study, it is important to note here that the AEA practices in the first iteration were created by the main team and did not include distributed technical agile teams. The main team, involves the most experienced member, as was described by interviewees at the Sydney location. This team includes the business team, AEA architects, different architects, program managers, iteration managers, and team leaders (e.g., Scrum, analyst, and tester). Therefore, the main team can collaboratively create AEA principles in the first iteration, and then all distributed teams will be included in evolving AEA practices to support the specific implementations of AEA principles. This way, the main team guarantees that even if new distributed teams or members join the project will start on solid predefined AEA principles. This finding supports the findings of (Britto, Smite et al. 2019), which insists on the role of the main team in initiating and creating the AEA principles.

Moreover, the findings of this research indicate that documentation of the AEA principles (e.g., models, artifacts) is an essential aspect that enables sharing and communicating AEA principles among dispersed team members. This result was not empirically investigated or reported in previous work. Also, AEA principles are made available in a common repository to all distributed teams. This finding is consistent with previous studies (e.g., (Batra 2009; Uludag, Kleehaus et al. 2019)).

Furthermore, the findings of this paper add to the literature by providing another insight, which indicates that AEA can be successfully implemented in GDAD to overcome the challenges of the traditional top-down architecture approach (i.e., forcing technical teams such as developers to use enterprise architecture) (Guo, Li et al. 2021). This is consistent with the findings of (Cammin, Heilig et al. 2021). On the contrary, an alternative cross-team development approach (i.e., including all teams in creating architecture principles) could be applied in GDAD to get the highest commitment, which may then reflect positively on GDAD project success (Samiei and Habibi 2020). In other words, all stakeholders should be involved in the creation of these architecture principles, which could provide the much-needed common ground for GDAD teams. This finding is consistent with (Canat, Català et al. 2018; Hauder, Roth et al. 2014; Uludag, Kleehaus et al. 2019), which urge the GDAD organization to allow the technical teams to participate in the creation of AEA principles to achieve the technical teams' commitment to increase project success (Kaddoumi and Watfa 2021).

5.2. Implications for Practice

This research also has some practical implications. Despite the many benefits of AEA, however, communicating AEA is a hard task and required big effort. Hence, the role of the architecture owner is important in creating and sharing AEA with the other agile team members. The AEA owner or architect is an important part of the GDAD with the role of focusing on sharing and communicating AEA principles among distributed teams. Moreover, his/ her role in collecting the feedback from his/ her technical team and sharing it with the AEA team. Distributed team and with the other AEA team members in other teams). AEA architects and architecture owners work collaboratively with the business team to create AEA principles. Communicating AEA principles is a big task that needs more attention. This is consistent with the findings of (Canat, Català et al.

2018) that the communication between the AEA team and technical teams is an issue. Poor communication of these principles may result in a lower level of commitment or not “really following” the AEA by the agile development team, thus the development team may not get the full benefits of AEA, and the project will go behind schedule or over budget, which might explain the high rate of failure of GDAD projects.

The findings also reveal a positive impact of AEA on the success of GDAD. AEA will keep GDAD teams on track by communicating project goals and scope, allowing projects to be completed on schedule, on budget, and with high quality (Cammin, Heilig et al. 2021). Exchanging AEA items and settling disagreements amongst GDAD teams are also among AEA benefits. By enabling communication among GDAD teams, AEA can result in better organizational alignment, information availability, agile team empowerment, and resource allocation, as demonstrated in Figure 2. These findings are consistent with (Canat, Català et al. 2018; Uludag, Kleehaus et al. 2019). AEA view aims to provide a high-level overview (common view) of the enterprise in the form of blueprints or models. This can highlight dependencies and support coordination among GDAD teams, and thus, improve GDAD success. However, to get the full benefits, AEA must be a part of the project's progress and emerging design and planning throughout its iterations, and it must be conveyed routinely in both ways between the technical team and the business team. Furthermore, this view can be used to direct and maintain communication among dispersed teams, resulting in improved teamwork and coordination (Brewer and Holmes 2016).

The findings suggest that an AEA governance model can help large-scale agile development projects by bringing together all key stakeholders. The findings stress that engaging technical teams as the primary stakeholders enhance acceptability and motivation was echoed by interviewees. This bottom-up approach enhances technical teams' acceptance of AEA governance

initiatives while simultaneously broadening the scope of application and significance of concepts and guidelines. External stakeholders, on the other hand, should monitor and verify compliance with rules, according to the majority of participants. They believe that external stakeholders, such as EAs, are important in monitoring their compliance to principles and regulations so that they can focus on developing software rather than wasting time. These findings are consistent with (Uludag, Nägele et al. 2019). To summarize, while establishing AEA in GDAD, we propose the following.

- Enterprise architects and their teams frequently underinvest in the significant work required to communicate EA. The first move in implementing AEA is to spread awareness of it across teams, teach team members how to use it, and demonstrate its advantages.
- A common perspective of AEA should be generated and distributed to development teams, using simple and unambiguous artifacts (diagrams, tables, terminology, high-level design, and so on). This view is maintained and updated in a centralized repository that also contains questions, cases, answers, a calendar of events, tracking information, and personal and team blogs.
- Checklists, standardized standards, and demonstrations are all available in the AEA view. Also, it establishes a unified framework and a variety of software delivery techniques.
- During the first iteration, the primary on-site team (most senior members) builds and solidifies the application's core architecture. Off-shore teams (for example, developers) can start developing on a robust architectural foundation.
- During development iterations, architecture owners or enterprise architects should collaborate with different teams. They should offer assistance and clear up any misconceptions between the teams.

5.3. Limitations and Future Directions

As experienced in any empirical study, the present research has some limitations. The recommendation provided by Runeson and Höst (2009) to assess the possible validity threats of the case study were followed in this paper. First, *Internal validity*, which represents a threat if causal relations are examined, isn't a problem since this paper present an exploratory case study and does not investigate causal relationships. Second, *Construct validity*, which represents a threat if the study followed a proper design to investigate the research questions. Third, *Reliability*, which represents a threat if the results depend on a specific researcher. In order to decrease the impact of the construct validity and reliability threats, we collected the data from several sources; 1) semi-structured interviews with different team members, 2) one researcher transcribed and coded the interviews and then a second researcher reviewed the results, and 3) the academic representative in SUNC reviewed the results of the interviews. Finally, *External validity* relates to the results' generalizability and the extent to which they are of relevance to other individuals rather than the investigated case. This paper's conclusions are focused on a single case study. As a consequence, the observations may not be relevant to other organized GDAD teams. Hence, future work might look at multi-case studies, larger organizations, or different structured GDAD teams. However, in order to decrease this threat, we emphasized precise replication of the interviewees and our goal to enable analytical generalization by offering a comprehensive view of the case, and hence, this threat can be minimized by comparing the similarities between this case and other organizations. It's vital to point out that, despite efforts, the number of interviewees was based on their availability and willingness to participate. The future work should include more members, especially from the distributed teams.

Despite the above contribution (discussed in Section 5.1 and Section 5.2), yet, more research should be conducted to investigate how other large GDAD firms implement their AEA and how

AEA owners and agile teams collaborate to build AEA principles. Future research on the importance of AEA and how to use it in agile development methodologies is also still needed (Canat, Català et al. 2018; Uludag, Kleehaus et al. 2019). In addition, future research may investigate other organizational structures. Also, future research may conduct longitudinal and cross-case studies on how AEA is emerged and evolved throughout the GDAD project. All that may help the GDAD industry and academia to build a more comprehensive approach.

6. Conclusions

The GDAD has been adopted by software development firms as a result of the success of a co-located setting. GDAD, on the other hand, confronts several obstacles. To address these issues, it was recommended that AEA be deployed in GDAD. However, there is a scarcity of empirical studies in this area. Accordingly, this study aims to fill that void by answering the two research questions of how AEA is successfully implemented in GDAD (RQ1) and what the role of AEA in GDAD is (RQ2). This was done to build a conceptualization view of implementing AEA in GDAD. The paucity of empirical studies on the successful use of AEA in GDAD prompted this study. This paper empirically investigated this gap through in-depth case study interviews and observation in order.

The findings reveal that AEA can be successfully and collaboratively implemented in GDAD. AEA definition, AEA communication, AEA view, AEA availability, Involvement of all stakeholders in creating AEA principles, AEA assistance to technical teams, and AEA feedback were identified and assessed as key concepts for successful AEA implementation in the GDAD environment. The findings also reveal that the AEA viewpoint is beneficial to the GDAD project's success. However, sharing, communicating, and training all members on using AEA artifacts

should be paid more attention to. Moreover, this study clarifies that the success of AEA in GDAD should follow the team-oriented approach in order to satisfy the expectations.

The findings of this paper may persuade GDAD practitioners to focus more on the principles of developing and distributing AEA among their dispersed members. Furthermore, the tactics and suggestions offered in this paper may be useful to GDAD companies in using AEA as a tool for GDAD project success. The findings of this article may also encourage additional research into GDAD firms with different structures or firms that successfully use AEA to validate the findings of this paper.

Funding: This research received no external funding.

Conflicts of interest: The authors declare no conflict of interest.

Ethical statement: The manuscript does not report results of a questionnaire.

References

Abdallah, A., et al. (2021). "Enterprise architecture measurement: A systematic literature review." Journal of Theoretical and Applied Information Technology **99**(6): 1257-1268.

Acharya, S. (2019). "Beyond learning outcomes: Impact of organizational flexibility on strategic performance measures of commercial e-learning providers." Global Journal of Flexible Systems Management **20**(1): 31-41.

AgileManifesto (2001). Manifesto for agile software development. https://agilemanifesto.org/, viewed 20 September 2020.

Ali Babar, M., et al. (2009). An industrial case of exploiting product line architectures in agile software development. In Proceedings of the 13th International Software Product Line Conference, Carnegie Mellon University. http://hdl.handle.net/10344/1142: 171-179.

Alzoubi, Y. I. and A. Q. Gill (2020). "An empirical investigation of geographically distributed agile development: The agile enterprise architecture is a communication enabler." IEEE Access **8**: 80269-80289.

Alzoubi, Y. I. and A. Q. Gill (2021). "The critical communication challenges between geographically distributed agile development teams: Empirical findings." IEEE Transactions on Professional Communication **64**(4): 322-337.

Alzoubi, Y. I., et al. (2015). "Distributed agile development communication: An agile architecture driven framework." Journal of Software **10**(6): 681-694.

Alzoubi, Y. I., et al. (2016). "Empirical studies of geographically distributed agile development communication challenges: A systematic review." Information and Management **53**(1): 22-37.

Alzoubi, Y. I., et al. (2018). "A measurement model to analyze the effect of agile enterprise architecture on geographically distributed agile development." Journal of Software Engineering Research and Development **6**(4): 1-24.

Amin, S. M. and B. M. Horowitz (2008). "Toward agile and resilient large-scale systems: Adaptive robust national/international infrastructures." Global Journal of Flexible Systems Management **9**(1): 27-39.

Ammad, G., et al. (2019). "An Empirical study to investigate the impact of communication issues in GSD in Pakistan's IT industry." IEEE Access **7**: 171648-171672.

Batra, D. (2009). "Modified agile practices for outsourced software projects." Communications of the ACM **52**(9): 143-148.

Batra, D., et al. (2010). "Balancing agile and structured development approaches to successfully manage large distributed software projects: A case study from the cruise line industry." Communications of the Association for Information Systems **27**(1): 379-394.

Bente, S., et al. (2012). Collaborative enterprise architecture: Enriching EA with lean, agile, and enterprise 2.0 practices, Morgan Kaufmann, USA.

Brewer, E. C. and T. L. Holmes (2016). "Better communication= better teams: A communication exercise to improve team performance." IEEE Transactions on Professional Communication **59**(3): 288-298.

Britto, R., et al. (2019). Performance evolution of newcomers in large-scale distributed software projects: An industrial case study. In Proceedings of the 14th International Conference on Global Software Engineering (ICGSE 2019), Montreal, QC, Canada, IEEE: 1-11.

Buchan, J., et al. (2019). Coordination in distributed agile software development: Insights from a COTS-based case study. In Proceedings of the Australasian Conference on Information Systems (ACIS 2019), Perth, Australia, AISel: 942-952.

Cammin, P., et al. (2021). Assessing requirements for agile enterprise architecture management: A multiple-case study. In Proceedings of the 54th Hawaii International Conference on System Sciences, Grand Wailea, Maui, Hawaii, ScholarSpace: 6007-6016.

Canat, M., et al. (2018). Enterprise architecture and agile development: Friends or foes? In Proceedings of the 22nd International Enterprise Distributed Object Computing Workshop (EDOCW 2018), Stockholm, Sweden, IEEE: 176-183.

Cataldo, M., et al. (2008). Socio-technical congruence: A framework for assessing the impact of technical and work dependencies on software development productivity. In the Proceedings of the Second ACM-IEEE international symposium on Empirical software engineering and measurement (ISESEM 2008), ACM: 2–11.

Chikhale, M. M. and M. Mansouri (2015). "An agile and collaborative framework for effective governance to enhance management in large-scale enterprise business systems: The case of Apple Inc." Global Journal of Flexible Systems Management **16**(3): 283-293.

Dingsøyr, T., et al. (2018). "Coordinating knowledge work in multiteam programs: Findings from a large-scale agile development program." Project Management Journal **49**(6): 64-77.

Dreesen, T., et al. (2020). The impact of modes, styles, and congruence of control on agile teams: Insights from a multiple case study. In Proceedings of the 53rd Hawaii International Conference on System Sciences, Grand Wailea, Hawaii, AISel: 6247-6256.

Ghantous, G. B. and A. Q. Gill (2019). "An agile-DevOps reference architecture for teaching enterprise agile." International Journal of Learning, Teaching and Educational Research **18**(7): 128 - 144.

Gill, A. Q. (2013). Towards the development of an adaptive enterprise service system model. In Proceedings of the 19th Americas Conference on Information Systems, Chicago, Illinois, 1-9.

Gill, A. Q. (2014). The Gill Framework - creating agile or adaptive enterprise project management capability for enterprise agility. http://aqgill.blogspot.com/, viewed 5 October 2020.

Gill, A. Q. (2015). Adaptive cloud enterprise architecture, World Scientific, Singapore.

Gill, A. Q., et al. (2018). "Scaling for agility: a reference model for hybrid traditional-agile software development methodologies." Information Systems Frontiers **20**(315–341).

Gill, A. Q. and D. Maheshwari (2021). Applying DevOps for distributed agile development: A case study. Advances in Software Engineering, Education, and e-Learning. Transactions on Computational Science and Computational Intelligence. H. R. Arabnia, L. Deligiannidis, F. G. Tinetti and Q. Tran. Cham, Springer**:** 719-728.

Guo, H., et al. (2021). Agile enterprise architecture by leveraging use cases. In Proceedings of the 16th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE 2021), Online Streaming, SCITEPRESS: 501-509.

Hanschke, S., et al. (2015). Integrating agile software development and enterprise architecture management. In Proceedings of the 48th Hawaii International Conference on System Sciences 2015, Kauai, HI, USA, IEEE: 4099-4108.

Hauder, M., et al. (2014). Agile enterprise architecture management: An analysis on the application of agile principles. In Proceedings of the 4th International Symposium on Business Modeling and Software Design, .

Highsmith, J. (2009). Agile project management: Creating innovative products. Pearson Education, USA.

Hoda, R., et al. (2017). "Systematic literature reviews in agile software development: A tertiary study." Information and Software Technology **85**: 60-70.

Horlach, B., et al. (2020). Everyone's going to be an architect: Design principles for architectural thinking in agile organizations. In Proceedings of the 53rd Hawaii International Conference on System Sciences, Grand Wailea, Hawaii, AISel.

Kaddoumi, T. and M. Watfa (2021). "A foundational framework for agile enterprise architecture." International Journal of Lean Six Sigma. https://doi.org/10.1108/IJLSS-03-2021-0057.

Korhonen, J. J., et al. (2016). Adaptive enterprise architecture for the future: Towards a reconceptualization of EA. In Proceedings of the 18th Conference on Business Informatics (CBI), Paris, France, IEEE: 272-281.

Kornstadt, A. and J. Sauer (2007). Tackling offshore communication challenges with agile architecture-centric development. In Proceedings of the Working Conference on Software Architecture (WICSA 2007) Mumbai, India, IEEE: 28-31.

Kulak, D. and H. Li (2017). The journey to enterprise agility: Systems thinking and organizational legacy, Springer, Cham.

Lee, J.-C. and C.-Y. Chen (2019). "Investigating the environmental antecedents of organizations' intention to adopt agile software development." Journal of Enterprise Information Management **32**(5): 869-886.

Leffingwell, D. (2007). Scaling software agility: Best practices for large enterprises, Pearson Education, USA.

Madison, J. (2010). "Agile architecture interactions." IEEE Software **27**(2): 41-48.

Mayring, P. (2010). Qualitative inhaltsanalyse. In: Buber R., Holzmüller H.H. (eds) Qualitative Marktforschung. Gabler. https://doi.org/10.1007/978-3-8349-9441-7_42.

McCarthy, S., et al. (2019). Towards a framework for shared understanding and shared commitment in agile distributed ISD project teams. In Proceedings of the 27th European Conference on Information Systems (ECIS 2019), Stockholm & Uppsala, Sweden, AISel: 1-15.

Medeiros, P., et al. (2021). An agile approach for modeling enterprise architectures. In Proceedings of the 23rd International Conference on Enterprise Information Systems (ICEIS 2021), Online Streaming, SCITEPRESS: 659-670.

Nundlall, C. and S. D. Nagowah (2021). "Task allocation and coordination in distributed agile software development: A systematic review." International Journal of Information Technology **13**(1): 321-330.

Paul, R., et al. (2016). "Global virtual team performance: The effect of coordination effectiveness, trust, and team cohesion." IEEE Transactions on Professional Communication **59**(3): 186-202.

Pedrycz, W. (2006). "Quantitative logic-based framework for agile methodologies." Journal of Systems Architecture **52**(11): 700-707.

Pillai, A. K. R., et al. (2012). "Implementing integrated Lean Six Sigma for software development: A flexibility framework for managing the continuity: Change dichotomy." Global Journal of Flexible Systems Management **13**(2): 107-116.

Ribeiro, F. L. and M. T. Fernandes (2010). "Exploring agile methods in construction small and medium enterprises: a case study." Journal of Enterprise Information Management **23**(2): 161-180.

Ross, J. W., et al. (2006). Enterprise architecture as strategy: Creating a foundation for business execution, Harvard Business Press, Boston, USA.

Runeson, P. and M. Höst (2009). "Guidelines for conducting and reporting case study research in software engineering." Empirical Software Engineering **14**(2): 131.

Samiei, E. and J. Habibi (2020). "The mutual relation between enterprise resource planning and knowledge management: A review." Global Journal of Flexible Systems Management **21**(1): 53-66.

Sauer, J. (2006). Agile practices in offshore outsourcing–an analysis of published experiences. In Proceedings of the 29th Information Systems Research Seminar in Scandinavia **29**: 12-15.

Shameem, M., et al. (2018). "Prioritizing challenges of agile process in distributed software development environment using analytic hierarchy process." Journal of Software: Evolution and Process **30**(11): e1979.

Shameem, M., et al. (2020). "Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process." Applied Soft Computing **90**: 106122.

Sharp, H. and H. Robinson (2008). "Collaboration and co-ordination in mature eXtreme programming teams." International Journal of Human-Computer Studies **66**(7): 506-518.

Stadler, M., et al. (2019). "Agile distributed software development in nine central European teams: Challenges, benefits, and recommendations." International Journal of Computer Science & Information Technology **11**(1): 1-18.

Turnu, I., et al. (2006). "Modeling and simulation of open source development using an agile practice." Journal of Systems Architecture **52**(11): 610-618.

Uludag, Ö., et al. (2019). What to expect from enterprise architects in large-scale agile development? A multiple-case study. In Proceedings of the 25th Americas Conference on Information Systems, Cancún, México.

Uludag, Ö., et al. (2019). Establishing architecture guidelines in large-scale agile development through institutional pressures: A single-case study. In Proceedings of the 25th Americas Conference on Information Systems, Cancún, México.

Yadav, V. (2016). "A flexible management approach for globally distributed software projects." Global Journal of Flexible Systems Management **17**(1): 29-40.

Yin, R. K. (2009). Case study research: Design and methods. Sage Publication, Thousands Oaks, USA.

Zaitsev, A., et al. (2020). "Coordination artifacts in agile Ssftware development." Information and Organization **30**(2): 100288.

Key Questions

1. How can AEA be successfully implemented in GDAD?
2. What impact may AEA have on the GDAD project's success?

Yehia Ibrahim Alzoubi has a Ph.D. in Information Systems from the University of Technology, Sydney. His research focuses on agile development, cybersecurity, and big data analysis. He received his MSc in IT from Central Queensland University in Sydney. He has several published works in scholarly peer-reviewed international scientific journals and conferences such as Information and Management, Journal of Strategic Security, Journal of Network and Computer Applications, Computer and Communications, and IEEE Transactions on Professional

Communication. Currently, he is an assistant professor at AUM. https://orcid.org/0000-0003-4329-4072

Asif Qumer Gill received the M.Sc. degree from the University of Newcastle Australia, Newcastle, Australia, and the Ph.D. degree from the University of Technology Sydney, Sydney, Australia, both in Computing Science. He is an Associate Professor with the School of Computer Science, University of Technology Sydney, Ultimo, NSW, Australia. He has extensive professional IT experience in various sectors, including banking, consulting, software, and telecommunication. His work has appeared in major academic and industry software conferences and in information systems journals, including the *Australasian Conference on Information Systems*, *AMCIS*, the International Conference on Information Systems, the Pacific Asia Conference on Information Systems, the *Communications of the Association of Information Systems*, *Information Systems*, *Information and Management*, and *Information Systems Frontiers*. He is a coordinating editor, section editor, and editorial board member of several academic journals. https://orcid.org/0000-0001-6239-6280