

RESEARCH

Open Access



A measurement model to analyze the effect of agile enterprise architecture on geographically distributed agile development

Yehia Ibrahim Alzoubi^{1*} , Asif Qumer Gill¹ and Bruce Moulton²

* Correspondence: yizoubi1977@gmail.com

¹School of Software, University of Technology Sydney, Building 11-Level 7/ 15 Broadway/ Sydney-Ultimo, Sydney, NSW 2007, Australia

Full list of author information is available at the end of the article

Abstract

Efficient and effective communication (active communication) among stakeholders is thought to be central to agile development. However, in geographically distributed agile development (GDAD) environments, it can be difficult to achieve active communication among distributed teams due to challenges such as differences in proximity and time. To date, there is little empirical evidence about how active communication can be established to enhance GDAD performance. To address this knowledge gap, we develop and evaluate a measurement model to quantitatively analyze the impact of agile enterprise architecture (AEA) on GDAD communication and GDAD performance. The measurement model was developed and evaluated through developing the AEA driven GDAD model and associated measurement model based on the extensive literature review, model pre-testing, pilot testing, item screening, and empirical evaluation through a web-based quantitative questionnaire that contained 26 different weighted questions related to the model constructs (AEA, GDAD active communication, and GDAD performance). The measurement model evaluation resulted in validated research model and 26 measures: 7 formative items for AEA, 5 reflective items for communication efficiency, 4 reflective items for communication effectiveness, 2 reflective items for each on-time and on-budget completion, and 3 reflective items for each software functionality and quality. The results indicate the appropriateness and applicability of the proposed measurement model to quantitatively analyze the impact of AEA on GDAD communication and performance.

Keywords: Geographically distributed agile development, Communication, Enterprise architecture, Performance

1 Background

Agile methods have been introduced to address a number of issues related to the development and delivery of software projects. These issues include projects running over budget, projects running behind schedule, and projects not meeting customers' needs and expectations (Chow & Cao 2008). Agile methods emerged over a period of time to increasingly influence future trends in software and information system development in both the local and distributed contexts (Gill 2015a). According to Ramesh

et al. (2006), GDAD can be defined as an agile development that involves distributed teams over different time zones and/or geographical locations. Hence, GDAD teams could be globally distributed or distributed within the same country in different locations (Ramesh et al. 2006). GDAD faces many challenges. The most noticeable challenge is the communication and coordination between dispersed stakeholders (Herbsleb & Mockus 2003; Korkala & Abrahamsson 2009).

Communication refers to the process of exchanging information between senders and receivers (McQuail 1987). Clark and Brennan (1991) defined communication as a collective activity that "requires the coordinated action of all the participants. Grounding is crucial for keeping that coordination on track." Communication grounding helps in achieving rapid communication with minimum effort (i.e. efficiency), and understandable message (i.e. effective) (Clarke & Brennan 1991; Modi et al. 2013). Herbsleb and Mockus (2003) reported two general types of communication in agile software development; informal and formal communication (Herbsleb & Mockus 2003). Informal communication is defined as a conversation (personal face-to-face) between software developers that takes place outside the formal structure or management's knowledge (Herbsleb & Mockus 2003). Since informal communication can quickly address changes in customer's requirements, it is more important than formal communication in agile software development (Henderson-Sellers & Qumer 2007). Herbsleb and Mockus (2003) defined the formal communication as the communication form that follow explicit and clear steps (e.g. backlog and card walls). According to Gill et al. (2012), although informal communication is more effective within co-located agile development teams, formal communication may be critical for GDAD success. Whether the communication is formal or informal, there is a need to understand the two important dimensions of active communication (Gill 2015b): communication efficiency and communication effectiveness (Alzoubi et al. 2016; Pikkariainen et al. 2008). To address customer's requirements and to mitigate the uncertainty in requirements, communication among agile development team should be active. This is even more critical in GDAD environment where face-to-face communication is hard to achieve among distributed teams due to numerous number of challenges (e.g. differences in geographical locations, time zones, cultures and languages) (Herbsleb & Mockus 2003).

Prior literature reports that active communication may enhance GDAD performance (on-time completion, on-budget completion, functionality and quality of software) by reducing the cost and time of project, and increase customer satisfaction (Paasivaara et al. 2009). However, there is a lack of empirical evidence to support this claim. To address this knowledge gap, there is a need to empirically examine how active communication can be achieved to enhance GDAD performance (Korkala et al. 2009). This paper addresses this important research gap, and uses agile enterprise architecture (EA) driven approach (Gill 2015b) for developing a communication model to enhance GDAD performance. This paper uncovers the relationships between the AEA, GDAD active communication and GDAD performance. Further, this paper evaluates the measurement model in order to examine the research model. This paper describes research which addresses the following research question:

RQ: How to quantitatively analyze the impact of AEA on GDAD communication and performance?

The main contribution of this paper is to fill the above small research gap by proposing and evaluating a measurement model that involves AEA, GDAD communication, and GDAD performance. This paper investigates if AEA can enhance GDAD communication and GDAD performance. Moreover, this paper clarifies the importance role of GDAD communication on GDAD performance.

The paper is structured as follows: Section 2 discusses the theoretical background of the research. Section 3 discusses the research model and hypotheses. Section 4 discusses the research method of validating the measurement model. Section 5 discusses the research findings and future directions. Section 6 concludes the paper.

2 Theoretical background

This section discusses the relevant literature and identifies three constructs of the research model: AEA (including one antecedent or independent variable: AEA), GDAD active communication (including two dimensions or dependent variables: efficiency and effectiveness), and GDAD performance (including four dimensions or dependent variables: on-time completion, on-budget completion, software functionality and software quality). Table 1 synthesizes the literature review and presents the resultant AEA driven GDAD communication model variables. The literature carefully reviewed the research model constructs (Fig. 1).

This study is an output of our ongoing research in the area of AEA and GDAD communication. It has gone through three stages. Firstly, we had conducted a detailed systematic literature review to identify the GDAD communication challenges (Alzoubi et al. 2016). We identified 17 challenges of GDAD communication and we categorized them into six categories: (1) *Distance Differences* (different time zones and different geographical areas), (2) *Team Configuration* (team size, number of teams, and coordination among teams), (3) *Project Characteristics* (project domain and project architecture), (4) *Customer Communication* (involvement of customer and involvement of customer representative), (5) *Organizational Factors* (project management process, communication tools, communication infrastructure, and organizational culture), and (6) *Human Factors* (language, national culture, trust, and personal practice). Secondly, we have proposed AEA as a potential facilitator and enhancer of GDAD communication (Alzoubi et al. 2015). AEA is used for two reasons: (1) it is more suitable to the people and active communication-driven agile development ways of working than the traditional documentation-driven and heavy process-centric EA approach, and (2) it offers a holistic and evolving shared view of the integrated information of business and IT architecture domains to enable effective and efficient communication among GDAD stakeholders. Usually, development teams rely on isolated software or IT architecture. EA as a holistic and integrated business and IT information will ensure that the important points of the whole EA are not overlooked by the GDAD teams. EA is perceived to be a glue to keep the GDAD teams aligned towards a shared vision (Edwards 2007). Thirdly, we have proposed the integrated AEA driven GDAD communication model (Alzoubi & Gill 2015). The fourth stage, which is the focus of this paper, is to validate the measurement model.

Table 1 The Research Model Variables Literature Review (Alzoubi & Gill 2015)

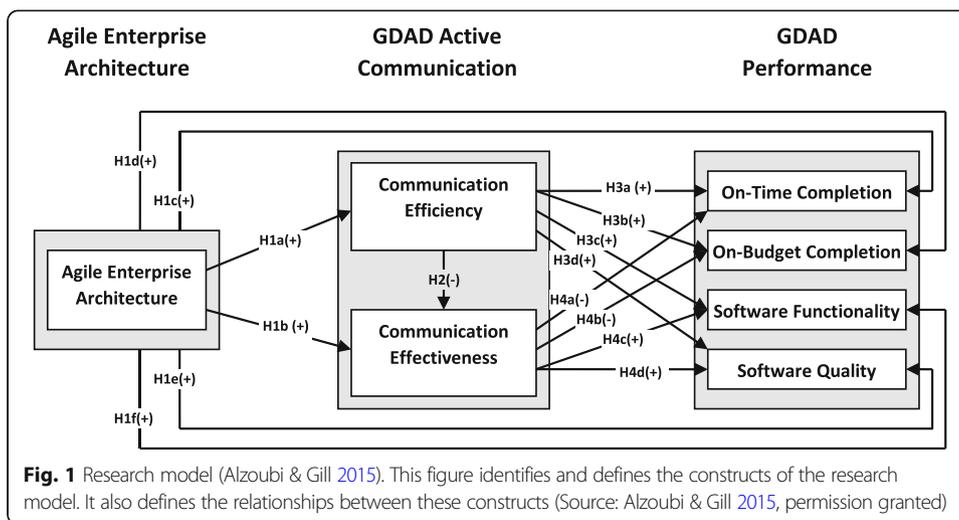
Variable	Literature	Relevant Definitions/Concepts/Ideas
Agile Enterprise Architecture	Ambller 2014	AEA should follow the strategy of “everyone owns the architecture” and be a team effort. AEA should use a minimum documentation and avoid big up-front design
	Bass et al. 2013	System quality can be predicted based solely on an evaluation of its architecture
	Gill 2013	AEA is a blue print that the overall structural, technical, social, behavioral, and facility elements of an enterprise
	Niemi & Pekkola 2015	EA artefacts can be used as a communication medium in many situations
	Ovaska et al. 2003	The architecture represents an important communication tool and a coordination mechanism in multi-site development
	Sauer (2010)	EA description can enhance communication in global software environment since EA can play as a common language among distributed developers
	Smolander 2002	Architecture can be assumed as a language metaphor such that architecture description about structures and solutions can be used as communication enabler between different stakeholders
	Svensson et al. 2012	Using architecture was perceived as delivering big amount of rich information in global sites and enhancing active communication by providing a common vocabulary among distributed teams
Communication Efficiency	Franke et al. 2010	Efficiency concerns with short manufacturing times, cycle times, lead times and work times
	Herbsleb & Mockus 2003)	Splitting work across distributed sites slows the work down Communication efficiency can be enhanced by timely communication and right people to communicate with
	Lee & Xia 2010	Efficiency relates to the cost, time, resources and effort associated with software team responses
	Melo et al. 2011	Efficiency concerns with doing things right of any task, even if it is not important to the job, that meets all the standards of time, quality, etc.
	Misra et al. 2009	Fast communication is a success factor of GDAD practices Fast communication is hindered in larger team context
Communication Effectiveness	Bhalerao & Ingle 2010	GDAD requires effective communication by adopting tools like teleconference and instant feedback from the customer
	Cannizzo et al. 2008	Communication effectiveness concerns with minimum disruption, waiting time and misunderstanding to receive the message Communication effectiveness requires immediate feedback that reduces waiting time and helps team members to address problems
	Dorairaj et al. 2011	Communication effectiveness facilitates rapid knowledge transfer between teams, allows team members to understand customer’s requirements and helps team members

Table 1 The Research Model Variables Literature Review (Alzoubi & Gill 2015) (Continued)

Variable	Literature	Relevant Definitions/Concepts/Ideas
On-Time Completion	Herbsleb & Moitra 2003	perform development activities more efficiently Communication effectiveness can be increased by reducing the effect of communication challenges such as time-zone differences and language barrier, and increasing effective formal and informal communication Communication effectiveness is defined as delivering a complete, adequate and accurate message Communication effectiveness requires more communication frequency and coordination between GDAD teams
	Melo et al. 2011	Effectiveness refers to doing the right things for the tasks that are important to the job, even if they are completed without meeting standards of time, quality, etc.
	Chow & Cao 2008 Drury-Grogan 2014	Delivering software project (system) on time Refers to the scheduling of tasks and completion dates
	Lee & Xia 2010 Melo et al. 2011	The extent to which a software project meets its time baseline goals Refers to meeting datelines, overtime needed to complete the work, and other time related issues
On-Budget Completion	Chow & Cao 2008	Delivering software project within estimated cost
	Lee & Xia 2010	The extent to which a software project meets its cost baseline goals
	Mahaney & Lederer 2006	The extent to which a software project is completed within the estimated budget
Software Functionality	Chow & Cao 2008	Meeting customer's requirements and objectives
	Lee & Xia 2010	The extent to which software project meets its functional goals, user needs and technical requirements
	Mahaney & Lederer 2006	meeting the technical goals of the software project
Software Quality	Bartelt & Dennis 2014	Different communication tools (e.g., IM and forum) result in significant different decision quality and team outcome
	Chow & Cao 2008	Delivering good product or project
	Conboy & Fitzgerald 2004	Achieving high standards of the software, supporting documentation and the development team
	Drury-Grogan 2014	Refers to how well the finished product functions
	Mahaney & Lederer 2006	Improving the project performance
	Misra et al. 2009	Quality criteria are productivity, customer satisfaction, business processes and functionality

2.1 Agile Enterprise architecture

Traditional EA is defined as "the organizing logic for business processes and IT infrastructure, reflecting the integration and standardization requirements of the company's operating model" (Ross et al. 2006, p. 9). Traditional EA provides a long-term view of



an organization’s processes, technologies, and systems, which enables individual projects to build capabilities rather than just fulfil immediate needs (Ross et al. 2006). The effective use of EA standards can provide cost and efficiency advantages by standardizing the different platforms, technologies, and application architectures among distributed sites (Boh & Yellin 2006; Ross et al. 2006). This can potentially reduce the organizational operational complexity, minimize waste and replication of system components, enable reuse of system components, and control the number of skilled individuals (e.g., developers) required to maintain the systems (Boh & Yellin 2006). Moreover, using EA standards enables integrating applications and sharing data across distributed sites. This helps distributed sites to integrate their business processes, develop key applications faster, and make effective use of organizational data (Bass et al. 2013).

However, in contrast to traditional process and documentation focused EA, AEA offers an incremental and people focused approach that aims to enhance agility (Gill 2013; Mthupha 2012). Agility is not only an outcome of technological achievement, advanced organizational and managerial structure and practice, but also an outcome of human skills, abilities, and motivations (Edwards 2007). Therefore, AEA should respond to changes in an effective and efficient manner to handle potential changes (Batra et al. 2010). Moreover, AEA should focus on the process inside an organization (i.e. improving the operations of the organization) as well as people since they have the biggest role in agile development (Edwards 2007). In order to ensure that AEA is not only developed in EA process (as in traditional EA), agility characteristics should be embedded in the end products and in the process, itself (Gill 2013). Agile software development practices with fine-tune of agile principles make it possible to apply agility into the process of EA (Edwards 2007).

AEA can be defined as the systematic process of following agile development principles while interpreting business strategy and vision into an effective enterprise (i.e. create, communicate and improve requirements and principles in flexible manner) (Gill 2013; Mthupha 2012). The scope of AEA includes people, processes, information and technology of the enterprise, and their relationships among each other and to the external environment (Ross et al. 2006). AEA provides holistic solutions that address the business challenges of the enterprise and support the governance needed to implement them (Edwards 2007).

2.2 GDAD active communication

Communication between stakeholders is core to the agile development (Agile Manifesto 2001). To overcome the issues of development time and cost, and customers' requirements changes, agile development focuses on the role of people and communication. People and interactions are valued over processes and tools, and customer collaboration over contract negotiation (Henderson-Sellers & Qumer 2007). Agility, the core of agile development, identifies how the agile team should communicate and respond to requirements changes. Lee and Xia (2010) p. 90, defined software development agility as "the software team's capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle." Conboy (2009) defined software development agility as the continued readiness "to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment" (Conboy 2009; p. 3400). According to the above agility definitions, communication among agile teams and team members should be efficient and effective (Gill 2013; Mthupha 2012).

As shown in Table 1, previous literature provides several theoretical concepts of communication efficiency and effectiveness. There is a common theme underlying the various definitions and descriptions in that communication is generally defined in terms of exchanging the adequate information in short time (Bhalerao & Ingle 2010; Cannizzo et al. 2008; Dorairaj et al. 2011; Melo et al. 2011; Misra et al. 2009). Furthermore, the previous literature views communication efficiency and communication effectiveness as the two different scopes of active communication. Efficiency focuses on short manufacturing times, work times, lead times and cycle times (Franke et al. 2010). Efficiency concerns with time, cost, resources or effort associated with communication (Lee & Xia 2010). Melo et al. (2011) defines efficiency and doing thing or task right (i.e. the task is completed meeting all the standards of time, quality, etc.), even if it is not important to the job. Accordingly, we define communication efficiency as delivering a message to a receiver with high quality and with minimal time, cost, effort, and resources required to establish communication. Effectiveness concerns with the practices or ways to effectively respond to market and customer demands (Franke et al. 2010). Communication effectiveness refers to minimal disruption, misunderstanding and waiting time to exchange the required information (Cannizzo et al. 2008). Melo et al. (2011) defined effectiveness as doing the right things just to the tasks that are important to the job, even if they are completed without meeting standards of quality, time and so on. Accordingly, we define communication effectiveness as delivering a message to the receiver who understands it as it was intended with minimal disruption and misunderstanding, even if it takes a long time.

2.3 GDAD performance

Researchers have diverse interpretations of software development performance. Some have referred to it as a project success (Mahaney & Lederer 2006; Misra et al. 2009). Project is assumed to be successful if it is completed within or close to the success criteria boundary such as the estimated time/schedule, budget/cost, functionality and

acceptable level of quality (Mahaney & Lederer 2006). Time, budget and quality are the key components of any project's success (Misra et al. 2009). Other authors have referred to performance as project effectiveness (e.g., Dyba et al. 2007; Jiang & Klein 2000). Project is assumed to be effective if it meets the speed, schedule and efficiency standards (Jiang & Klein 2000). Aspects related to effectiveness are project duration, effort and quality (Dyba et al. 2007). Wallace et al. (2004) define performance through two pillars: product performance (i.e. reliability, functionality, satisfaction, quality, and user requirements) and process performance (i.e. on-time and on-budget).

Prior literature (agile and traditional software development), in general, assume three major dimensions (i.e. on-time completion, on-budget completion and functionality) that make and distinguish software development performance (Lee & Xia 2010). However, according to Chow and Cao (2008), quality is the fourth important dimension of performance. Hence, this study refers to four dimensions of software development performance: functionality, quality, on-time completion and on-budget completion. Functionality refers to the extent to which the software meets its functional goals, technical requirements and user needs (Lee & Xia 2010). Chow and Cao (2008) defined quality as delivering a good working product (Chow & Cao 2008). On-time completion refers to delivering a software according to its duration baseline goals (Lee & Xia 2010). On-budget completion refers to delivering a software according to its cost baseline goals (Lee & Xia 2010).

3 Research model and hypotheses

Building on the guidelines of Lewis et al. (2005), the first stage in developing constructs is to identify and define the constructs, and evaluate the constructs by academics and practitioners' experts. This was done and introduced in our previous paper (Alzoubi & Gill 2015). The output of this stage is a refined model with its related hypotheses as shown in Fig. 1. Therefore, the research constructs and the hypotheses are briefly discussed in this paper. The research model identifies three constructs and seven variables: (1) AEA (independent variable: AEA), (2) GDAD active communication (dependent variables: communication efficiency and communication effectiveness), and (3) GDAD performance (dependent variables: on-time completion, on-budget completion, software functionality and software quality).

3.1 Relationship between AEA and GDAD active communication

Agile principles emphasize that self-organizing teams, business people and agile developers must work together throughout the project to deliver the best architectures and (Batra et al. 2010). In a small co-located agile team (i.e. development team and business people work together in daily basis to work out the best project architecture and design through active communication and continuous collaboration), this principle is very successful (Ambler 2014). However, in GDAD environment, this principle is not easy to be achieved (Batra et al. 2010). In such complex GDAD environment, different silo GDAD teams need to be efficiently and effectively communicated with different changes to their and other dependent project(s) architectures and requirements in order to align their work. According to Ovaska et al. (2003), using the overall AEA holistic integrated shared view can help achieving the best design and architecture. The integrated view of AEA provide the "possibility to see and discuss how different parts (the ICT systems, the processes, etc.) are interconnected and interplay. Understanding means not only

knowing what elements the enterprise consists of and how they are related from different aspects, but also how the elements work together in the enterprise as a whole” (Karlsen 2008, p. 219).

This integrated shared view may serve as a common information model for enabling clear communication among GDAD teams and can provide a single view of the AEA information to GDAD stakeholders (Ambler 2014; Gill, 2015b; Ovaska et al. 2003). “Architecture provides a common language in which different concerns can be expressed, negotiated, and resolved at a level that is intellectually manageable even for large, complex systems. Without such a language, it is difficult to understand large systems sufficiently to make the early decisions that influence both quality and usefulness” (Bass et al. 2013, p. 29). Moreover, it can provide a rich source of information shared by all GDAD teams (Madison 2010; Svensson et al. 2012). This integrated view helps GDAD team members to coordinate their work through interfaces of their components (i.e. different components can be developed separately). This means that considering development of other components and the frequencies of communication with other team members are decreased (Ovaska et al. 2003). Therefore, we propose.

H1a: Agile Enterprise Architecture positively affects the efficiency of the GDAD communication.

H1b: Agile Enterprise Architecture positively affects effectiveness of the GDAD communication.

3.2 Relationship between AEA and GDAD performance

It is possible to predict system quality based solely on an evaluation of its architecture (Bass et al. 2013). AEA draws from a uniform infrastructure, platform, application, and communicates the architecture value and status with all stakeholders (Madison 2010). It improves implementation consistency and reduces the number of errors by providing the basis for architecture rules to the involved teams (Kornstadt & Sauer, 2007). AEA may enhance GDAD performance since it is assumed as a placeholder for software quality, security, reliability and modifiability (Kornstadt & Sauer, 2007). Therefore, we propose.

H1c: Agile Enterprise Architecture positively influences on-time completion of GDAD project.

H1d: Agile Enterprise Architecture positively influences on-budget completion of GDAD project.

H1e: Agile Enterprise Architecture positively influences GDAD project quality.

H1f: Agile Enterprise Architecture positively influences GDAD project functionality.

3.3 Relationship between GDAD active communication dimensions (efficiency and effectiveness)

Due to GDAD communication challenges, the message may not be received as effectively as intended. Considering the impacts of time, cost and effort on communication,

a GDAD team tends to first decide what and how much they would communicate, which affects communication effectiveness (Dorairaj et al. 2011). Clear communication may not be achieved by sending short message (Clarke & Brennan 1991). Accordingly, increasing the communication effectiveness may decrease the communication efficiency and vice versa. Therefore, we propose.

H2: GDAD communication efficiency negatively affects effectiveness of the GDAD communication.

3.4 Relationship between GDAD active communication and GDAD performance

Fast communication may lead to fast responding to customer requirements, which results in high agile development performance (Cockburn 2007; Misra et al. 2009). Delay in identifying project impacts, dependencies and resultant changes in GDAD environment may lead to longer development duration and extra cost (Boehm & Turner 2003). If the efficiency of GDAD communication is low, the amount of extra time and costs required for handling customer requirements changes is high (Cockburn 2007). This may increase the additional time and cost, and not meeting the assigned time and budget targets (Lee & Xia 2010). Therefore, we propose.

H3a. Communication efficiency positively influences on-time completion of GDAD project.

H3b. Communication efficiency positively influences on-budget completion of GDAD project.

H3c. Communication efficiency positively influences GDAD project functionality.

H3d. Communication efficiency positively influences GDAD project quality.

According to Dyba et al. (2007), higher communication effectiveness comes at the price of considerably longer time and higher cost, while shorter and faster communications come at a price of substantially lesser effectiveness. To effectively communicate about many different customer requirements and requirements' changes, GDAD team may need new capabilities and resources or reconfigure existing capabilities and resources (Lee & Xia 2010). This requires a considerable amount of extra cost and time (Lee & Xia 2010). Furthermore, communication about customer's requirements and requirements' changes helps in correcting system configuration, and improve design and product quality (Bhalerao 2010). The functionality and quality of the system will not satisfy "up-to-date" customer needs if the team fails to embrace important changes (Lee & Xia 2010). Therefore, we propose.

H4a. Communication effectiveness negatively influences on-time completion of GDAD project.

H4b. Communication effectiveness negatively influences on-budget completion of GDAD project.

H4c. Communication effectiveness positively influences GDAD project functionality.

H4d. Communication effectiveness positively influences GDAD project quality.

4 Measurement model evaluation

The measurement model evaluation was done through developing and testing an instrument that evaluates the constructs of the model and its related variables using a set of items (statements) related to each variable (Lewis et al. 2005; Straub et al. 2004). The measurement model analysis refers to the pre-testing of a specific research instrument of investigation (Baker 1994). Performing the measurement model analysis is important and helps in refining the research model, instrument, and hence increasing the accuracy of the research method and its results (Baker 1994; Straub et al. 2004). As part of our research, the initial analysis involves three sequential steps: (1) developing and evaluating the instrument, (2) pre-testing the instrument, pilot testing the instrument, and instrument item screening, and (3) exploratory assessment (Lewis et al. 2005). These steps are discussed in the following sub-sections.

4.1 Research measures development

The main measures validity is done to ensure that a group of measurement items appropriately represent the concept (i.e. construct) under investigation (Straub 2004). The initial research measurement items (50 items) were distilled from the previous empirical studies (e.g., Herbsleb & Mockus 2003; Lee & Xia 2010; Mahaney & Lederer 2006). Then, the initial research measures were emailed to a group of five experts from both academia and industry in the field of agile software development. Three of them were from GDAD industry; a Scrum Master, a developer and an architect. Two of them worked as agile developers and now are assistants' professors teaching agile development and agile enterprise architecture subjects. The experts helped in evaluating the fit between each item and associated construct.

Based on the feedback, we redesigned new set of items for AEA variable where more attention had been paid to the role of solution architecture. One expert wrote: *"in an ideal organization, EA is used to produce a solution architecture which will be used to guide agile teams."* The items of communication efficiency and effectiveness were fine-tuned to focus on communication and GDAD performance enhancement. One expert wrote: *"...the questions should focus on asking how communication is going to be enhanced using "AEA driven GDAD communication" rather than asking about the efficiency and effectiveness inside an organization."* The way of how GDAD communication and AEA are related was reconsidered such that new items were included. The architect wrote: *"I think the definition of the EA should be clarified and then the link between the two (EA and GDAD communication) should be clarified to specify how EA is going to address the stated problems."* The output of this evaluation was 40 items left for the next evaluation.

The instrument was then sent to an academic expert on measurements who assessed their quality and the statements of items to ensure they reflect the intended sample frame. This procedure ensured the content validity for the measurement items (Straub et al. 2004).

4.2 Pre-test, pilot test and item screening

Pre-test, pilot test and item screening are the three initial tests for the measurement items under development (Lewis et al. 2005). The aim of this step is to further appraise and purify the measures that also ensure their content validity. These three tests were conducted sequentially and discussed as follows:

1. Pre-test was conducted by sending the measurement items (of step 1) to three PhD scholars affiliated with the university who were asked to complete and critique the instrument. This is important for initial instrument design, such as format, content, understandability, terminology, and ease and speed of completion (Nunnally 1978). As an output of this test, the format of the questionnaire was improved and wordings of the items were fine-tuned.
2. Using snowball sampling technique, which is recommended for exploratory research (Gregor & Klein 2014), pilot test was conducted by sending the questionnaire to five respondents based on the pre-established unit of analysis (the unit of analysis for this study is an EA driven GDAD environment individual). Three of those respondents were contacts to one of the researchers. Those three respondents nominated another two respondents. The respondents were asked to complete the questionnaire, provide comments on difficulties in completing the questionnaire, and offer suggestions for improvement, including specifying any additional item statements they felt were missing or items that should be deleted. As an output of this test, the adjusted number of items was dropped to 35 items.
3. The updated measurement items were sent to a group of experts (the same group as step 1) for item screening process. The purpose of this step is to empirically screen the items for content validity applying a quantitative procedure based on the procedure developed by Lawshe (1975). The experts were asked to evaluate the relevance of each item to its related construct on a three-point scale: '1 = Not Relevant,' '2 = Useful (but not Essential),' '3 = Essential' (Lewis et al. 2005). The following formula was used to compute the content validity ratio (CVR) based on the data provided by the expert panel.

$$\text{CVR} = (n - N/2) / (N/2).$$

Where "N" is the total number of respondents. "n" is the frequency count of the number of panelists rating the item as appropriate, either '3 = Essential' or '2 = Useful (but Not Essential)' and '1 = Not Relevant'. Only '3 = essential' in the screening process was included in this study (Lawshe 1975). We evaluated the CVR for each item for statistical significance (score more than 51%). If the item is statistically significant, its content validity is of accepted level. On the other hand, non-significance indicates that the content validity is unacceptable (i.e. item is rejected). We dropped all items that found not statistically significant (i.e. based on CVR value) from the instrument. The final number of items was 26 items (see [Appendix](#)). The updated measurement items are shown in Table 2. The overall content validity score of the all items was 0.85.

All indicators of communication efficiency, communication effectiveness, on-time completion, on-budget completion, software functionality, and software quality were

Table 2 The Validated Version of the Research Instrument

Variable	Items	Source
Enterprise Architecture	7 items	Designed by the research team, based on experts' feedback
Communication Efficiency	5 items	Herbsleb & Mockus, 2003; Piccoli et al. 2004
Communication Effectiveness	4 items	Herbsleb & Mockus, 2003; Piccoli et al. 2004; Yadav et al. 2007
On-Time Completion	2 items	Jiang & Klein, 2000; Lee and Xia, 2010
On-Budget Completion	2 items	Jiang & Klein, 2000; Lee and Xia, 2010
Software Functionality	3 items	Lee & Xia, 2010
Software Quality	3 items	Mahaney & Lederer, 2006

modeled as reflective (caused by their latent constructs [Petter et al. 2007]) measures. However, the indicators of AEA was modeled as formative measures since these indicators are not expected to have covariation within the same latent construct and they are causes of, rather than caused by, their latent construct (Petter et al. 2007). All Items of the questionnaire were measured using a seven-point Likert scale. The scale is ranging from “strongly disagree” to “strongly agree” or “to no extent” to “to a great extent”.

4.3 Measurement model assessment using survey questionnaire

As mentioned above, the unit of analysis for this study is individual who work in GDAD environment and using EA in his/ her development. A snowball sampling technique was utilized to identify the individuals, which is preferable for exploratory study (Gregor & Klein 2014). In this technique GDAD team members invite or link us with other members that can provide us with rich information. One of the authors provided some names of his contacts who work in GDAD. The criteria for the individual selection included that the individual should be an AEA driven GDAD team member and has a willingness to participate.

Since the population of this study is all individuals or organizations who use AEA driven GDAD, a number of GDAD team members from different countries were targeted. The questionnaire was sent to potential respondents in various industrial sectors such as finance, telecommunications and healthcare in order to gather experiences from different AEA driven GDAD members. Due to the nature of this research, the questionnaire was not restricted to any project size, particular organization or nationality. Moreover, to allow respondents completing the survey any time they wish and take their time to finish it, the questionnaire was made available online using the SurveyMonkey tool.

Although there is no common agreement about the sample size of the study, Hunt et al. (1982) recommended sample size between 12 and 30 subjects. This study used SPSS 16.0 and Partial Least Squares PLS 3.0 package (Ringle et al. 2015) to test the measurement model. According to Hair et al. (2014), the minimum sample size should be 10 times the maximum number of arrowheads pointing at a latent variable anywhere in the PLS model. Applied to this study, a sample size of at least 30 ($10 * 3 = 30$, maximum arrows number was 3 – see Fig. 1) is needed.

A random sample of 60 GDAD team members who use AEA driven GDAD were contacted by email to complete the questionnaire. A total of 45 surveys were returned, achieving 75% survey response rate. 8 incomplete surveys were exempted from the

analysis. Thus, 37 of the returned surveys were usable responses. Of the surveys analyzed, as Table 3 shows, 20 respondents (54%) were developers, 7 (19%) were architects, 4 (10.8%) were team leaders/Scrum Masters and 4 analysts, and 2 (5.4%) QA/test. Most of the respondents had (2–4) years' experience in GDAD.

The assessment of the measurement model includes the evaluation of reliability, convergent validity, and discriminant validity. Reflective and formative constructs are treated differently; for formative constructs, different indicators are not expected to demonstrate internal consistency and correlations, unlike reflective constructs (Chin & Todd 1995). The relevance and level of contribution of each indicator was assessed by examining the absolute indicator weights. Taking into account the recommendations of Lewis et al. (2005), the evaluation for reflective and formative constructs, and issues relating to bias, are discussed in the following sub-sections.

4.3.1 Reflective measurement model

The reflective measurement model was estimated by calculating four values (Hair et al. 2014; Straub et al. 2004): (1) individual indicator reliabilities (the degree to which the item is consistent and stable measures over time and free of random errors), (2) composite reliability (CR) (a measure of internal consistency reliability), (3) convergent validity (the extent of positive correlation between a measurement item and the alternate measures of the same construct), and (4) discriminant validity (the extent to which a construct is truly distinct from other constructs by empirical standards).

First, we assessed the indicator reliability of reflective items by testing the outer loadings of each item on its own latent construct. The outer loadings of the reflective constructs should be above the recommended threshold of 0.708 and t-statistical significance value should be above than 1.96 (Hair et al. 2014). As Table 4 shows, all loadings were above the threshold value and were significant (the diagonal in bold font values), and the cross loadings (the off-diagonal values) are less than the outer loading for any construct. Henseler et al. (2009) recommended that the reflective items should be removed only if their outer loadings are less than 0.4.

Second, we assessed the reliability of reflective constructs with Cronbach's alpha coefficient and composite reliability (see Table 5). All reflective constructs in our study

Table 3 The Demographic Data for Exploratory Assessment

Variable	Value	Frequency	Percent %
Job Title	Developer	20	54
	Team leader/Scrum Master	4	10.8
	Analyst	4	10.8
	Architect	7	19
	QA/test	2	5.4
Industry	Banking and financial	20	54
	Automotive	5	13.5
	Telecommunications	8	21.7
	Other	4	10.8
GDAD Experience	< 2 years	5	13.5
	2–4 years	25	67.5
	5–10 years	7	19

Table 4 Cross Loadings and Outer Loadings

Scale Items	EFFIC	EFFECT	TIME	BUDGT	FUNC	QLTY
EFFIC1	0.928***	0.321	0.445	0.329	0.511	0.544
EFFIC2	0.905***	0.412	0.423	0.298	0.533	0.567
EFFIC3	0.886***	0.512	0.388	0.344	0.574	0.582
EFFIC4	0.854***	0.433	0.456	0.421	0.433	0.511
EFFIC5	0.880***	0.522	0.423	0.412	0.428	0.564
EFFECT1	0.399	0.869***	0.522	0.377	0.541	0.587
EFFECT2	0.333	0.883***	0.340	0.433	0.587	0.612
EFFECT3	0.412	0.860***	0.511	0.375	0.486	0.678
EFFECT4	0.421	0.844***	0.545	0.388	0.487	0.487
TIME1	0.512	0.365	0.968***	0.612	0.435	0.422
TIME2	0.562	0.398	0.968***	0.588	0.412	0.387
BUDGT1	0.436	0.411	0.611	0.861***	0.511	0.395
BUDGT2	0.469	0.433	0.633	0.935***	0.574	0.355
FUNC1	0.355	0.542	0.645	0.377	0.894***	0.588
FUNC2	0.322	0.387	0.322	0.322	0.811***	0.468
FUNC3	0.378	0.455	0.332	0.411	0.906***	0.611
QLTY1	0.413	0.512	0.412	0.322	0.487	0.919***
QLTY2	0.391	0.532	0.356	0.465	0.455	0.887***
QLTY3	0.458	0.511	0.298	0.395	0.467	0.847***

EFFIC Communication Efficiency, EFFECT = Communication Effectiveness, TIME = On-Time Completion, BUDGET = On-Budget Completion, FUNC=Functionality, QLTY = Quality; $p < .05$, *** $p < .01$

scores above 0.70 (the recommended value for both Cronbach's alpha and composite reliability) (Chin & Todd 1995; Hair et al. 2014).

Third, we assessed convergent validity by testing the average variance extracted (AVE) value and factor analysis. In the exploratory factor analysis (see Table 6), six factors corresponding to the reflective constructs in our model were extracted. Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy was found 0.78, which is well above the recommended value of 0.50. All item loadings on stipulated constructs were greater than 0.50, and all eigenvalues were greater than one as required. Convergent validity was also measured using the AVE that should be at least 0.5 (Hair et al. 2014), which indicates that the construct explains more than half the variance of its indicators. All AVE values were found well above 0.5 (see Table 5).

Finally, we assessed discriminant validity by testing the square root of each construct's AVE values (Hair et al. 2014). Discriminant validity ensures that the correlation

Table 5 Cronbach's Alpha (α), Composite Reliability (CR), Average Variance Extracted (AVE), Correlations

Construct	α	CR	AVE	EFFIC	EFFECT	TIME	BUDGET	FUNC	QLTY
EFFIC	0.935	0.954	0.805	0.897					
EFFECT	0.886	0.889	0.670	0.144	0.864				
TIME	0.932	0.980	0.961	0.209	0.674	0.968			
BUDGET	0.768	0.954	0.895	0.345	0.499	0.348	0.899		
FUNC	0.841	0.925	0.804	0.090	0.679	0.121	0.469	0.871	
QLTY	0.862	0.961	0.891	0.201	0.624	0.093	0.781	0.483	0.885

Table 6 Factor Analysis of Reflective Constructs

Construct Items	Component					
	1	2	3	4	5	6
Communication efficiency						
EFFIC1	0.854	0.294	0.222	-0.120	-0.024	0.100
EFFIC2	0.796	0.355	0.301	-0.051	0.001	-0.093
EFFIC3	0.791	0.224	0.151	-0.448	-0.082	0.075
EFFIC4	0.816	0.167	-0.014	-0.212	0.070	-0.189
EFFIC5	0.855	0.143	0.068	-0.006	-0.078	-0.187
Communication effectiveness						
EFFECT1	-0.458	0.698	0.193	0.012	0.299	-0.177
EFFECT2	-0.355	0.698	0.056	-0.107	0.500	-0.090
EFFECT3	-0.195	0.826	-0.130	-0.278	0.023	0.267
EFFECT4	-0.466	0.592	0.255	-0.357	0.189	0.250
On-Time completion						
TIME1	0.138	0.095	0.120	0.357	0.813	0.192
TIME2	0.178	0.034	0.319	0.314	0.818	0.125
In-Budget completion						
BUDGT1	0.248	0.160	-0.363	0.402	-0.002	0.668
BUDGT2	-0.096	0.240	-0.185	0.047	-0.156	0.819
Functionality						
FUNC1	-0.235	-0.257	0.193	0.712	0.309	0.287
FUNC2	-0.362	-0.328	0.202	0.625	0.402	-0.184
FUNC3	-0.059	-0.325	0.197	0.667	0.475	-0.160
Quality						
QLTY1	-0.082	-0.123	0.809	-0.342	-0.178	-0.018
QLTY2	0.156	0.032	0.773	-0.366	0.172	-0.287
QLTY3	-0.204	0.176	0.740	-0.345	-0.258	-0.045
Eigenvalue	3.971	2.989	2.352	2.279	1.874	1.625
Variance Extracted	26.887	20.613	13.258	13.004	6.525	4.476
Cumulative Variance (%)	26.887	47.500	60.759	73.763	80.288	84.764
Unrotated Variance (%)	34.439	30.079	6.946	5.569	4.481	3.250

of indicator to its latent construct should be greater than its highest correlation with any other construct as it was suggested by Fornell-Larcker criterion (Fornell & Larcker 1981). It is based on the idea that a construct shares more variance with its associated indicators than any other construct. This is confirmed in our findings, as shown in Table 5, where the numbers along the diagonal in bold font (square root of AVE) are greater than the correlation between latent constructs (off-diagonal elements). Overall, the reliability and convergent and discriminant validity (construct validity) were supported in the research model; therefore, all reflective items were retained.

4.3.2 Formative measurement model

According to Hair et al. (2014), we assessed the reliability and convergent and discriminant validity of the formative construct AEA by testing: (1) collinearity (indication of high correlations between formative indicators), and (2) indicator validity. First, we assessed the collinearity of AEA construct by testing the value of tolerance or VIF

(variance inflation factor = $1/\text{tolerance}$). Tolerance represents the amount of variance of one formative indicator not explained by other indicators in the same construct. In PLS, a tolerance value of 0.2 (i.e. VIF value of 5 or higher) indicates a potential collinearity problem (Hair et al. 2014). Consequently, the indicator with $\text{VIF} > 5$ should be removed as long (Hair et al. 2014). The tolerance and VIF estimates for the measures of AEA are shown in Table 7. The results suggested that the reliability of AEA is supported and all indicators can be retained.

Finally, to determine the relative contribution of each indicators constituting AEA construct (Chin & Todd 1995), we assessed the indicator validity by testing the relevance and significance of indicators outer weights. Bootstrapping procedure (5000 samples) was used to calculate the weight's significance and t values for each indicator. All indicator values were significant as shown in Table 7. The results suggested that AEA exhibited adequate convergent and discriminant validity; therefore, all formative indicators were retained.

4.3.3 Nonresponse bias and common method bias

Discrepancies between researcher and respondent perceptions of the meanings of items' statements can cause response bias for individual items across a sample (Lewise et al. 2005). To test for nonresponse bias issue, the sample was split into two groups based on the time of the collected responses (Sivo et al. 2006). The early and late response groups demographics were then compared. The comparison showed no significant differences between the two groups, which indicates that response bias is not likely to be a serious issue.

Common method biases may occur when data is collected via only one method (survey in the case of this study), or via the same method but only at one point in time (Straub et al. 2004). This may result in a variance that the items have in common with each other due to the data collection method rather than the hypothesized relationships between constructs or between measures and constructs. To assess whether or not potential common method bias was a significant issue (Malhotra et al. 2006), we performed Harman's one-factor test to all reflective items (Podsakoff et al. 1986). The test was done by entering all constructs into an unrotated principal components factor analysis and examining the resultant variance. Common method bias threat is high if a single factor accounts for more than 50% of the variance (Podsakoff et al. 1986). The analysis revealed that there was no single factor which explained a substantial amount of variance (the most covariance explained by one factor is only 35%, see the last row of Table 6), which indicates that common method bias does not pose a significant threat to the measurement validity of this study (Chin et al. 2012). To decrease the

Table 7 Collinearity Test Values; Outer Weights; AEA = AEA

Formative indicator	Tolerance	VIF	Outer weights	t-value	Sig	p-value
AEA1	0.445	4.073	0.201	2.206	**	0.037
AEA2	0.293	3.405	0.214	2.349	**	0.028
AEA3	0.445	2.246	0.369	4.758	***	0.004
AEA4	0.362	2.758	0.329	3.675	***	0.008
AEA5	0.293	3.410	0.224	2.462	**	0.015
AEA6	0.403	2.481	0.333	3.789	***	0.006
AEA7	0.290	3.445	0.188	2.015	**	0.043

** $p < .05$, *** $p < .01$

possibility of common method bias, we also distributed (not grouped together) the items that represent one construct in the questionnaire (Gregor & Klein 2014).

5 Discussion and future directions

This paper presented the AEA driven GDAD communication model based on the review of relevant body of knowledge and experts' evaluation. This model includes three constructs: AEA, GDAD active communication, and GDAD performance. The central construct is GDAD active communication, which includes two dimensions: communication efficiency and communication effectiveness. This model employs AEA as GDAD communication enabler. It provides a new perspective of AEA as a comprehensive integrated shared view and a common language to enhance GDAD communication and performance (Gill 2015a). AEA evolves as different interdependent and independent portfolios, programs and projects architectures are delivered in short increments. AEA represents up-to-date real-time information about integrated business and IT architecture elements, which provides the shared guiding vision to synchronize the GDAD teams' tasks in different time zones and locations. As discussed earlier, the parts of the whole AEA (e.g. business and IT capabilities, services and products) are developed and delivered by the GDAD teams in different increments at different times. Real-time information about evolving EA is appropriate for synchronizing the work and enabling active communication among GDAD teams.

AEA seems beneficial for GDAD teams. However, we need to empirically analyze the impact of AEA on GDAD. This paper is an attempt to address this important question and proposed the required measurement model to analyze the impact of AEA on GDAD communication and performance. The initial measurement model items (50 items) were distilled from existing empirical studies, and then evaluated by groups of experts which resulted in decreasing the number of items to 40 items. In addition, we conducted pre-testing and item screening, which resulted in further decreasing the number of items to 26 items. We conducted a quantitative measurement model assessment using a web-based survey, and collected the data from 37 subjects, and analyzed it. The assessment results indicate the validation and applicability (fit-for-purpose) of the measurement model for effectively analyzing the impact of AEA on GDAD communication and performance. This is an important contribution and fills a small research gap in the existing knowledge body by proposing and evaluating measures that exhibit discriminant validity, between one dimension of AEA (new measures were developed), two dimensions of GDAD communication (communication efficiency and communication effectiveness), and four dimensions of performance (on-time completion, on-budget completion, functionality, and quality). Like any other research, this work needs to be viewed with its limitation. One may argue the limited number of subjects for model assessment – 37 subjects. Those subjects may answer these questions from their point view or may be from same geographical context (e.g. Asia and Europe). According to Hunt et al. (1982), this number (i.e. 37) is sufficient to evaluate the measurement model. This does not claim to test the measurement model from all possible perspectives. This study represents the first step in investigating and presenting new perspectives of AEA and its impact on GDAD communication and performance. Future research would further investigate and refine the measurement method items and scales based on the inter-rater reliability and manipulation validity (Straub

et al. 2004). In the next phase of this research, inter-rater reliability will be tested by the means of interviews, and manipulation validity will be performed using the experimental data.

6 Conclusion

There is growing interest among practitioners about the use of AEA for large and complex GDAD. However, it is not clear how AEA does affect the GDAD? In order to address this important research question, this paper, as a part of a larger research project, reported an AEA driven GDAD model and associated measurement model to analyze the impact of AEA on GDAD communication and performance. Initially, we developed AEA driven GDAD model based on the extensive literature review, which is a first contribution of this paper. Then a measurement model was developed to actually analyze the impact of AEA on GDAD, which is a second contribution of this paper. This measurement model was then evaluated through preliminary tests and a survey in the field. The evaluation results indicate that the proposed AEA driven GDAD model and related measurement model are reliable “fit-for-purpose” to assess the impact of AEA on GDAD communication and performance. It is anticipated that this study will serve as a starting point for developing and testing theories for guiding communication in GDAD environment so that organizations can effectively build and sustain communication that will ultimately improve their GDAD performance.

7 Appendix

7.1 Measurement scales and items

The respondent was asked to refer to the following definitions while completing this survey because different titles or definitions can be used by different organizations to communication, enterprise architecture and performance.

Agile development: Software development that rapidly creates change, proactively or reactively embraces change, and learns from change while contributing to perceived customer value. Scrum and XP are two examples of agile methods.

Geographically distributed agile development (GDAD): Agile development that includes a number of teams or/and team members distributed over different locations and time zones.

Enterprise architecture (EA): A blueprint that describes the overall social, structural, behavioral, technological, and facility elements of an enterprise’s operating environment that share common goals and principles. Enterprise architecture includes different architecture domains such as Application architecture, Platform architecture, Infrastructure architecture, Business architecture, Solution architecture, and Information architecture.

Communication: Exchanging information or messages between two parties (i.e. sender and receiver) efficiently and effectively.

Communication efficiency: Delivering high quality messages with minimal time, cost, effort, and resources.

Communication effectiveness: Delivering a message as it was intended with minimal disruption and misunderstanding, even if it takes a long time.

On-time completion: The extent to which a software project meets its duration baseline goals.

On-budget completion: The extent to which a software project meets its cost baseline goals.

Functionality: The extent to which the delivered project meets its user’s needs, functional goals and technical requirements.

Quality: Measure how good the work is (according to ISO 8402, it is “the totality of characteristics of an entity that bear on its ability to satisfy stated and implied needs”).

Agile enterprise architecture (AEA) can be used as an integrated shared view in GDAD. Here, we refer to this approach as (**EA driven GDAD approach**) as shown in Fig. 2.

This diagram explains how “EA driven GDAD approach” can be used in GDAD environment. There are different architectural views according to different architectural levels, as shown in Fig. 2.

- Distributed teams (up to N teams) share the “project architecture view”.

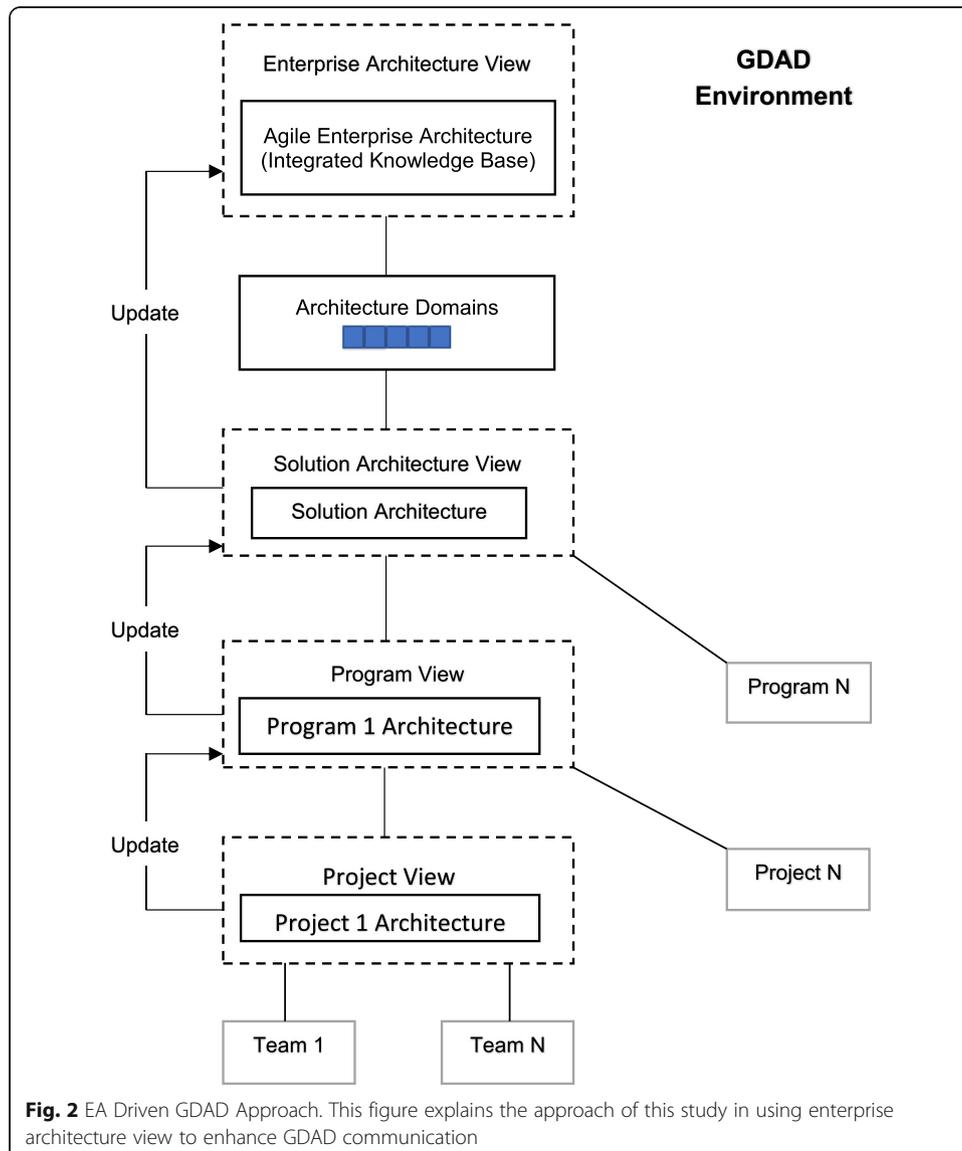


Fig. 2 EA Driven GDAD Approach. This figure explains the approach of this study in using enterprise architecture view to enhance GDAD communication

- Different projects (up to N projects) share the “program architecture view”. The same is applied to the “solution architecture view”, which can have “N” number of program architectures.
- Each architecture updates the architecture above. All architectures are then updated and shared using the enterprise architecture integrated knowledge base. This knowledge base can be represented in multiple repositories which grant access to all distributed stakeholders. This way ensures that all stakeholders are updated with the latest changes (i.e. project or program changes).

Using EA driven DAD approach, identify to what extent do you agree or disagree with the following statements.

AEA (formative) (1 = very much; 7 = very little)

1. Enterprise architecture framework and the framework of GDAD are aligned (AEA1)
2. Enterprise architecture documentations are regularly updated to align with the projects in GDAD (AEA2)
3. Enterprise architecture is used to define projects/programs (e.g., business/IT gap analysis) in GDAD (AEA3)
4. Enterprise architecture is used to assess major project investment in GDAD (AEA4)
5. Solution architecture, as a part of enterprise architecture, guides the projects at program levels and project levels in GDAD (AEA5)
6. Solution architecture evolves from small iterations, and the changes in solution architecture are reflected in enterprise architecture (AEA6)
7. Enterprise architecture is used to govern project implementation in GDAD (AEA7)

Communication efficiency (reflective) (1 = strongly agree; 7 = strongly disagree)

1. Information needed about GDAD project is achieved quickly (EFFIC1)
2. Information needed about GDAD project is achieved easily (EFFIC2)
3. The stakeholders needed to communicate with is reached quickly (EFFIC3)
4. The stakeholders needed to communicate with is reached easily (EFFIC4)
5. The cost of communication (e.g., less travelling to meet face-to-face) is decreased (EFFIC5)

Communication effectiveness (reflective) (1 = strongly agree; 7 = strongly disagree)

1. All GDAD team members are clear about their tasks (EFFECT1)
2. Enough information is provided about customer requirements and project progress to GDAD team members (EFFECT2)
3. Detailed information is provided from distributed stakeholders (EFFECT3)
4. Accurate information is provided from distributed stakeholders (EFFECT4)

On-time completion (reflective) (1 = strongly agree; 7 = strongly disagree)

1. GDAD projects is completed on-time according to the original schedule (TIME1)
2. GDAD tams complete their tasks on-time according to the original schedule (TIME2)

On-budget completion (reflective) (1 = strongly agree; 7 = strongly disagree)

1. GDAD projects is completed on-budget according to the original budget (BUDGET1)
2. GDAD tams complete their tasks on-budget according to the original budget (BUDGET2)

Functionality (reflective) (1 = strongly agree; 7 = strongly disagree)

1. GDAD project achieves its functional goals (FUNC1)
2. GDAD project meets its technical functional requirements (FUNC2)
3. GDAD project meets customer's functional requirements (FUNC3)

Quality (reflective) (1 = strongly agree; 7 = strongly disagree)

1. GDAD project solves the given problem (QLTY1)
2. GDAD project improves the way of customers' use to perform their activities (QLTY2)
3. GDAD project achieves customer's satisfaction (QLTY3)

Abbreviations

AVE: Average variance extracted; CVR: Content validity ratio; EA: Enterprise architecture; GDAD: Geographically distributed agile development; KMO: Kaiser-Meyer-Olkin; VIF: Variance inflation factor

Acknowledgements

We would like to thank the associate editor and the reviewers for their constructive comments and suggestions on the earlier drafts of this paper.

Funding

The research was not funded by any organization.

Availability of data and materials

Data will be available on request.

Authors' contributions

The first author (YA) contributed to 60% of the manuscript. The second author (AG) contributed to 30% of the manuscript. The third author (BM) contributed to 10% of the manuscript. All authors read and approved the final manuscript.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Author details

¹School of Software, University of Technology Sydney, Building 11-Level 7/ 15 Broadway/ Sydney-Ultimo, Sydney, NSW 2007, Australia. ²School of Electric, Mechanic and Mechatronic Systems, University of Technology Sydney, Sydney, Australia.

Received: 10 February 2017 Accepted: 9 February 2018

Published online: 02 March 2018

References

- Alzoubi Y I, Gill A (2015) An agile enterprise architecture driven model for geographically distributed agile development. In: the 24th international conference on information system development, Harbin, China, 2015
- Alzoubi YI, Gill AQ, Al-Ani A (2015) Distributed agile development communication: an agile architecture driven framework. *J Softw* 10(6):681–694
- Alzoubi YI, Gill AQ, Al-Ani A (2016) Empirical studies of geographically distributed agile development communication challenges: a systematic review. *Inf Manag* 53(1):22–37
- Ambler S (2014) Agile enterprise architecture. <http://agiledata.org/essays/enterpriseArchitecture.html>. Accessed 20 Jan 2015.
- Baker TL, Risley AJ (1994) *Doing social research*. McGraw-Hill, New York
- Bartelt VL, Dennis AR (2014) Nature and nurture: the impact of automaticity. *MISQ* 38(2):521–538
- Bass L, Clements P, Kazman R (2013) *Software architecture in practice*. Addison-Wesley, Upper Saddle River
- Batra D, Xia W, VanderMeer D, Dutta K (2010) 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* 27(1):379–394
- Bhalerao, S, Ingle M (2010) Analyzing the modes of communication in agile practices. In: the 3rd international conference on computer science and information technology (ICCSIT), pp 391–395, 2010
- Boehm B, Turner R (2003) *Balancing agility and discipline: a guide for the perplexed*. Addison-Wesley professional
- Boh WF, Yellin D (2006) Using enterprise architecture standards in managing information technology. *J Manag Inf Syst* 23(3):163–207
- Cannizzo F, Marconetti G, Moser P (2008) Evolution of the tools and practices of a large distributed agile team. In: agile conference, pp 513–518, 2008
- Chin WW, Thatcher JB, Wright RT (2012) Assessing common method bias: problems with the ULMC technique. *MISQ* 36(3):1003–1019
- Chin WW, Todd PA (1995) On the use, usefulness, and ease of use of structural equation modeling in MIS research: a note of caution. *MISQ* 19:237–246
- Chow T, Cao D-B (2008) A survey study of critical success factors in agile software projects. *J Syst Softw* 81(6):961–971
- Clark HH, Brennan SE (1991) Grounding in communication. *Perspect Soc Shared Cognit* 13(1991):127–149
- Cockburn A (2007) *Agile software development: the cooperative game*. Addison-Wesley, Harlow
- Conboy K (2009) Agility from first principles: reconstructing the concept of agility in information systems development. *Inf Syst Res* 20(3):329–354
- Conboy K, Fitzgerald B (2004) Toward a conceptual framework of agile methods. In: Zannier C, Erdogmus H, Lindstrom L (eds) *Extreme Programming and Agile Methods-XP/Agile Universe 2004*. Springer Berlin Heidelberg, pp 105–116
- Dorairaj S, Noble J, Malik P (2011) Effective communication in distributed agile software development teams. In: Sillitti A et al (eds) *Agile processes in software engineering and extreme programming*. Springer-Verlag, Berlin Heidelberg, pp 102–116
- Drury-Grogan ML (2014) Performance on agile teams: relating iteration objectives and critical decisions to project management success factors. *Inf Softw Technol* 56(5):506–515
- Dyba T, Arisholm E, Sjoberg DI, Hannay JE, Shull F (2007) Are two heads better than one? On the effectiveness of pair programming. *IEEE Soft* 24(6):12–15
- Edwards C (2007) Building a case for an Agile Enterprise Architecture Process –Part 2 USA: ProcessWave. <http://www.agileea.com/Whitepapers/2007-02-04-AgileEnterpriseArchitectureV1.00-Part2.pdf>
- Fornell C, Larcker DF (1981) Structural equation models with unobservable variables and measurement error: algebra and statistics. *J Mark Res*:382–388
- Franke U, Ekstedt M, Lagerström R, Saat J, Winter R (2010) Trends in enterprise architecture practice—a survey. In: *Trends in enterprise architecture research* Springer, Heidelberg, 16–29
- Gill A Q (2013) Towards the development of an adaptive enterprise service system model. In: 19th Americas conference on information systems, Chicago, Illinois, pp 1–9, 2013
- Gill AQ (2015a) Distributed agile development: applying a coverage analysis approach to the evaluation of a communication technology assessment tool. *Int J e-Collab* 11(1):57–76
- Gill AQ (2015b) Adaptive cloud enterprise architecture. World Scientific
- Gill A Q, Bunker D, Seltsikas P (2012) Evaluating a communication technology assessment tool (Ctat): a case of a cloud based communication tool. In: 16th Pacific Asia conference on information systems (PACIS), paper 88, 2012
- Gregor S, Klein G (2014) Eight obstacles to overcome in the theory testing genre. *J Assoc Inf Syst* 15(11):i-xix
- Hair J, Hult T, Ringle C, Sarstedt M (2014) *A primer on partial least squares structural equation modeling (PLS-SEM)*. Sage Publications
- Henderson-Sellers B, Qumer A (2007) Using method engineering to make a traditional environment agile. *Cutter IT J* 20(5):30
- Henseler J, Ringle CM, Sinkovics RR (2009) The use of partial least squares path modeling in international marketing. *Adv Int Mark (AIM)* 20:277–319
- Herbsleb JD, Mockus A (2003) An empirical study of speed and communication in globally distributed software development. *IEEE Trans Softw Eng* 29(6):481–494
- Hunt SD, Sparkman RD, Wilcox JB (1982) The pretest in survey research: issues and preliminary findings. *J Mark Res*:269–273
- Jiang J, Klein G (2000) Software development risks to project effectiveness. *J Syst Softw* 52(1):3–10
- Karlsen A (2008) A research model for enterprise modeling in ICT-enabled process change. In: Stirna J, Persson A (eds) *The Practice of Enterprise Modeling*. Springer Berlin Heidelberg, pp 217–230.
- Korkala M, Pikkariainen M, Conboy K (2009) Distributed agile development: a case study of customer communication challenges. In: Abrahamsson P, Marchesi M, Maurer F (eds) *Agile processes in software engineering and extreme programming*. Springer-Verlag, Berlin, Heidelberg, pp 161–167

- Kornstädt A, Sauer J (2007) Mastering dual-shore development: the tools and materials approach adapted to agile offshoring. In: Meyer B, Joseph M (eds) *Software engineering approaches for offshore and outsourced development LNCS 4716*. Springer, Heidelberg, pp 83–95
- Lawshe CH (1975) A quantitative approach to content validity. *Pers Psychol* 28(4):563–575
- Lee G, Xia W (2010) Toward agile: an integrated analysis of quantitative and qualitative field data. *MISQ* 34(1):87–114
- Lewis BR, Templeton GF, Byrd TA (2005) A methodology for construct development in MIS research. *Eur J Inf Syst* 14(4): 388–400
- Madison J (2010) Agile architecture interactions. *IEEE Softw* 27(2):41–48
- Mahaney RC, Lederer AL (2006) The effect of intrinsic and extrinsic rewards for developers on information systems project success. *Proj Manag J* 37(4):42–54
- Malhotra NK, Kim SS, Patil A (2006) Common method variance in IS research: a comparison of alternative approaches and a reanalysis of past research. *Manag Sci* 52(12):1865–1883
- Agile Manifesto. (2001) Manifesto for agile software development. <http://www.agilemanifesto.org/>. Accessed 9 Oct 2014
- McQuail D (1987) *Mass communication theory: an introduction*. Sage Publications
- Melo C, Cruzes D S, Kon F, Conradi R (2011) Agile team perceptions of productivity factors. In: *Agile Conference*, pp 57–66, 2011
- Misra SC, Kumar V, Kumar U (2009) Identifying some important success factors in adopting agile software development practices. *J Syst Softw* 82(11):1869–1890
- Modi S, Abbott P, Counsell S (2013) Negotiating common ground in distributed agile development: a case study perspective. In: the 8th international conference on global software engineering (ICGSE), IEEE, pp 80–89, 2013
- Mthupha B (2012) *A framework for the development and measurement of agile enterprise architecture*. Rhodes University, Dissertation
- Niemi E, Pekkola S (2015) Using enterprise architecture artefacts in an organisation. *Enterp Inf Syst*:1–26
- Nunnally JC (1978) *Psychometric theory*. McGraw-Hill, New York
- Ovaska P, Rossi M, Martiin P (2003) Architecture as a coordination tool in multi-site software development. *Softw Process Improv Pract* 8(4):233–247
- Paasivaara M, Durasiewicz S, Lassenius C (2009) Using scrum in distributed agile development: a multiple case study. In: the 4th international conference on global software engineering (ICGSE), pp 195–204, 2009
- Petter S, Straub D, Rai A (2007) Specifying formative constructs in information systems research. *MISQ* 31(4):623–656
- Piccoli G, Powell A, Ives B (2004) Virtual teams: team control structure, work processes, and team effectiveness. *Inf Technol People* 17(4):359–379
- Pikkarainen M, Haikara J, Salo O, Abrahamsson P, Still J (2008) The impact of agile practices on communication in software development. *Empir Softw Eng* 13(3):303–337
- Podsakoff PM, Organ DW (1986) Self-reports in organizational research: problems and prospects. *J Manag* 12(4):531–544
- Ramesh B, Cao L, Mohan K, Xu P (2006) Can distributed software development be agile? *CACM* 49(10):41–46
- Ringle C M, Wende S, Becker J-M (2015) *SmartPLS 3*. SmartPLS GmbH: Boenningstedt. <https://www.smartpls.com/>. Accessed 19 Oct 2015
- Ross JW, Weill P, Robertson D (2006) *Enterprise architecture as strategy: creating a foundation for business execution*. Harvard business press
- Sauer J (2010) Architecture-centric development in globally distributed projects. In: D. Šmite and et al (ed) *agility across time and space*. Springer-Verlag, Berlin Heidelberg, pp 321–329
- Sivo SA, Saunders C, Chang Q, Jiang JJ (2006) How low should you go? Low response rates and the validity of inference in IS questionnaire research. *J Assoc Inf Syst* 7(6):351–413
- Smolander K (2002) Four metaphors of architecture in software organizations: finding out the meaning of architecture in practice. In: *International symposium on empirical software engineering*, IEEE, pp 211–221
- Straub D, Boudreau M-C, Gefen D (2004) Validation guidelines for IS positivist research. *Commun Assoc Inf Syst* 13(1): 379–427
- Svensson RB, Aurum A, Paech B, Gorschek T, Sharma D (2012) Software architecture as a means of communication in a globally distributed software development context. In: Dieste O, Jedlitschka A, Juristo N (eds) *Product-focused software process improvement*. Springer, Heidelberg, pp 175–189
- Wallace L, Keil M, Rai A (2004) Understanding software project risk: a cluster analysis. *Inf Manag* 42(1):115–125
- Yadav V, Adya M, Nath D, Sridhar V (2007) Investigating an 'agile-rigid' approach in globally distributed requirements analysis. In: 11th Pacific Asia conference on information systems (PACIS), paper 12, 2007