An Agile Enterprise Architecture Driven Model for Geographically Distributed Agile Development

Yehia Ibrahim Alzoubi *yehia.i.alzoubi@student.uts.edu.au*
School of Software, University of Technology Sydney
Sydney, Australia

Asif Qumer Gill *asif.gill@uts.edu.au*
School of Software, University of Technology Sydney
Sydney, Australia

Abstract

Agile development is a highly collaborative environment, which requires active communication (i.e. effective and efficient communication) among stakeholders. The active communication in geographically distributed agile development (GDAD) environment is difficult to achieve due to many challenges. Literature has reported that active communication play critical role in enhancing GDAD performance through reducing the cost and time of a project. However, little empirical evidence is known about how to study and establish active communication construct in GDAD in terms of its dimensions, determinants and effects on GDAD performance. To address this knowledge gap, this paper describes an enterprise architecture (EA) driven research model to identify and empirically examine the GDAD active communication construct. This model can be used by researchers and practitioners to examine the relationships among two dimensions of GDAD active communication (effectiveness and efficiency), one antecedent that can be controlled (agile EA), and four dimensions of GDAD performance (on-time completion, on-budget completion, software functionality and software quality).

Keywords: Geographically distributed agile development, Communication effectiveness, Communication efficiency, Enterprise architecture.

1. Introduction

Agile methods have been introduced to address a number of issues related to project development and delivery, such as over-budget or behind schedule projects, and not meeting customer's needs and expectations [9]. Agile methods emerged over a period of time to increasingly influence future trends in software development in both the local and distributed contexts [21]. GDAD refers to the agile development that includes teams or/and team members distributed over different locations and time zones [26]. GDAD faces many challenges. The most noticeable challenge is the communication and knowledge sharing between dispersed teams and customers [1,28,40].

Communication is defined as the process of exchanging information between senders and receivers [33]. Communication can also be defined as the way to manage relationships between developers and consumers [32]. These definitions draw our attention to the importance and effectiveness of communication between the parties included in agile development. However, agile methods promise faster development thus improving the communication efficiency too [37]. Thus, agile methods require effective and efficient communication (i.e. active communication) among stakeholders to achieve the highest software quality and customer satisfaction [2,21]. Herbsleb and Mockus [25] divide communication in agile software development into two general types; formal and informal communication. Formal communication can be defined as the explicit clear communication such as the agile requirements backlog and card walls [25]. Informal communication refers to personal peer-oriented conversation among developers which takes place outside the official structure and sometimes without the knowledge of management [25]. Informal communication helps in filling and correcting mistakes quickly, which supports and ensures agile principles [25].

To overcome the uncertainty and changeable customer's requirements, active communication is considered vital in a co-located agile development team. The vitality is greater in GDAD due to less chances of informal face-to-face communication [21]. In GDAD, active communication is harder to achieve due to many challenges such as differences in language, culture, distance, time-zone, architecture used, management process, and communication infrastructure between distributed teams [5].

It has been reported in literature that active communication may enhance GDAD design and quality by reducing the project development time and cost [37]. The empirical knowledge on the subject seems to be scarce. To address this knowledge gap, there is a need to empirically examine how active communication can be achieved to enhance GDAD performance [28]. This paper addresses this important gap and proposes an agile EA driven model for enabling GDAD active communication and examining how this model can enhance GDAD performance. The aim of this paper is to uncover the relationships between the agile EA, GDAD communication and GDAD performance. This paper is an incremental output of our ongoing research in the area of agile enterprise architecture and GDAD communication.

The paper is structured as follows: Section 2 discusses the research method. Section 3 presents the theoretical foundation. Section 4 discusses the agile EA driven GDAD communication research model and hypotheses. Section 5 discusses the preliminary evaluation of research model. Section 6 discusses the research findings, limitations and future directions before concluding.

2. Research Method

This section describes the overall research methodology that we are applying to iteratively develop and evaluate the proposed model. We are applying an integrated multi-method approach that uses both qualitative and quantitative techniques [20]. This approach consists of three phases: (1) building the research model, which includes two stages: build the theoretical research model (i.e. agile EA driven GDAD communication model) from the literature review and preliminary model evaluation, (2) survey data collection, which includes two stages: conducting pilot study (i.e. measurement validation) and analysing the main survey data (i.e. hypothesis testing), and (3) final model evaluation by conducting semi-structured interviews using case study approach. Using this multi-method approach helps in addressing limitations for both qualitative and quantitative methods by providing the objectivity of the statistics and deeper understanding of the study context [20]. The scope of this paper is limited to phases 1 of this large multi-year project. In the first stage of phase 1, the agile EA driven GDAD communication model was built based on the previous related literature. In the second stage of phase1, preliminary model evaluation was conducted by involving five experts from both academia and industry. This paper presents the refined version of the model for further feedback from the research community. This preliminary evaluation is done to identify any issues and get directions before proceeding further in the research.

3. Theoretical Foundation

This section discusses the relevant literature and identifies three constructs of the proposed agile EA driven GDAD communication model: agile EA (including one antecedent or independent variable: agile EA), 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 agile EA driven GDAD communication model variables.

Table 1. The agile EA driven GDAD communication model variables.

Variable	Literature	Relevant Definitions/Concepts/Ideas
Agile Enterprise Architecture	[6]	▪ Agile EA should be a team effort following the strategy of "everyone owns the architecture" where big up-front design is not required and a minimum documentation is required
	[24]	▪ Agile EA describes the overall structural, behavioral, social, technological, and facility elements of an enterprise
	[36]	▪ The architecture is an important communication tool ▪ The architecture is a coordination mechanism in multi-site development
	[38]	▪ Architecture can be assumed as a language metaphor, where architecture description about structures and solutions serve as communication enabler between different stakeholders
	[39]	▪ Using architecture was perceived as delivering large volumes of rich information in global sites and enhances active communication through a common vocabulary
Communication Efficiency	[18]	▪ Efficiency concerns with short manufacturing times, lead times, cycle times and work times
	[25]	▪ Splitting work across sites slows the work down ▪ Enhance communication efficiency through timely communication and right people to communicate with
	[29]	▪ Efficiency relates to the time, cost, resources, or effort associated with software team responses
	[34]	▪ Efficiency refers to doing things right of any task, even if it is not important to the job, that meets all the standards of time, quality, etc.
	[35]	▪ Rapid communication is a success factor of GDAD practices ▪ Larger team might pose great hindrance to fast communication
Communication Effectiveness	[11]	▪ GDAD requires effective communication (e.g., teleconference) and instant feedback from the customer
	[13]	▪ Communication effectiveness means minimal disruption, waiting time, and misunderstanding to get the information ▪ Communication effectiveness requires immediate feedback which reduces waiting time, helps team members to address problems, and minimize clashes
	[16]	▪ Communication effectiveness facilitates knowledge transfer rapidly between team members, allows team members to understand the requirements from clients, and helps team members perform development activities 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
	[25]	▪ Communication effectiveness refers to delivering an complete, adequate and accurate message ▪ Communication effectiveness requires communication frequency and coordination between GDAD team
	[34]	▪ Effectiveness accounts for doing the right things. Refers just to the tasks that are important to the job, even if they are completed without meeting standards of time, quality, etc.
On-Time	[14]	▪ Delivering software project on time

Completion	[29]	▪ The extent to which a software project meets its baseline goals for duration
	[34]	▪ Accounts for meeting datelines, overtime needed to complete the work, and other time related issues
On-Budget Completion	[14]	▪ Delivering software project within estimated cost and effort
	[29]	▪ The extent to which a software project meets its baseline goals for cost
	[31]	▪ The extent to which a software project is completed within or near the estimated budget
Software Functionality	[14]	▪ Meeting all requirements and objectives
	[29]	▪ The extent to which the delivered software system meets its functional goals, user needs, and technical requirements
	[31]	▪ The extent to which a software project meets its technical goals
Software Quality	[14]	▪ Delivering good product or project outcome
	[14]	▪ Achieving high standards in terms of the software and supporting documentation produced, and the development team
	[31]	▪ The extent to which the project performance is improved
	[35]	▪ Productivity, customer satisfaction, business processes, and functionality can be perceived as quality criteria

This research adopts a challenge driven approach. Firstly, we had conducted a detailed systematic literature review to identify the GDAD communication challenges [5]. Seven challenges categories were identified in the systematic literature review: (1) *People Differences* (refer to four communication challenges: cultural difference, people attitude, language, and trust), (2) *Distance Differences* (refer to two communication challenges: different time zones and different geographical areas), (3) *Team Issues* (refer to four challenges: team size, team distribution, cross-team work, and cross-team communication), (4) *Technology Issues* (refer to four challenges: communication tools, infrastructures, communication bandwidth, and communication cost), (5) *Architectural Issues* (refer to four challenges: architectures used, organizational structure, managerial structure, and project domain), (6) *Process Issues* (refer to three challenges: process, control, and commitment-level to communication, and (7) *Customer Communication* (refers to involvement and transparency with customer). We focused our research on agile EA (see (5) *Architectural Issues*), which is the least investigated area in the context of GDAD. This research adopts an agile EA driven approach as a potential facilitator and enhancer of communication in GDAD environment. Agile EA [22] seems more appropriate and fit to the people driven and light-weight agile ways of working, and therefore, it has been adopted for this research.

3.1. GDAD Active Communication: Efficiency and Effectiveness

Communication between developers and with customers is core to the agile development [2]. Agile software development approaches have been introduced as the alternative methods to the traditional "heavyweight" methods that have not gotten enough ability to address the current issues such as development time and cost, and respond to uncertain changeable customer's requirements [10,15,26]. To overcome these issues, agile development focuses on the role of people and communication. It values people and interactions over processes and tools, and customer collaboration over contract negotiation [2]. It promotes close collaboration and communication between empowered development teams and customers [2].

As shown in table 1, prior literature provides various 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 [11,13,16,34,35].

Furthermore, it appears that prior literature tends to view communication as consisting of two important elements that correspond to our conceptualization of the two communication dimensions: communication efficiency and communication effectiveness. Efficiency concerns with short manufacturing times, lead times, cycle times and work times [18]. Efficiency relates to time, cost, resources, or effort associated with communication [29]. It also refers to doing things (i.e. any task) right, even if it is not important to the job (i.e. the task is completed meeting all the standards of time, quality, etc.) [34]. Effectiveness concerns with the practices or ways to effectively respond to market and customer demands [18]. Communication effectiveness means as little as possible disruption, minimal waiting time to get the required information and minimal chances of misunderstanding [13]. It also refers to doing the right things just to the tasks which are important to the job, even if they are completed without meeting standards of time, quality, etc. [34]. To avoid any confusion in the definitions of effectiveness and efficiency from the previous literature, 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. Moreover, 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.

3.2. Agile Enterprise Architecture

The EA is defined as “a blueprint that describes the overall structural, behavioral, social, technological, and facility elements of an enterprise’s operating environment that share common goals and principles” [24, p. 1]. Agile enterprise is defined as " an entity is said to be an agile enterprise when an enterprise is responsive (scans, senses and reacts appropriately to expected and unexpected changes), flexible (adapts to expected or unexpected change at any time), speedy (accommodates expected or unexpected changes rapidly), lean (focuses on reducing waste and cost without compromising on quality), and learning (focuses on enterprise fitness, improvement and innovation)” [24, p. 3]. Hence, agile EA can be defined as "a blueprint that describes the overall structural, behavioral, social, technological, and facility elements of an enterprise’s operating environment that share common goals and principles with the ability of responsiveness, flexibility, speediness, leanness, and learning". Unlike traditional process-focused heavy architecture frameworks (e.g., Zachman [41]), agile architecture frameworks (e.g., The Gill Framework® [23]) provide human-centric, align to agile principles, and adaptive capabilities to adapting, defining, operating, managing and supporting an agile EA.

Agile principles make it clear that the best architectures, requirements, and designs emerge from self-organizing teams [2]. Moreover, business people and agile developers must work together daily throughout the project [2]. These two principles work well for a small co-located agile team where developers work side by side and communicate face-to-face with business people [19]. This helps developers and business people to work out the best project architecture and design through effective collaboration [19]. However, in GDAD environment, the opportunity for this effective collaborative and continuous communication among developers and with business people is limited due to many barriers, as discussed above [5]. This situation becomes even more challenging when the organization deploys many GDAD teams that need to work simultaneously on different dependent features or projects. In such complex GDAD environment, efficient and effective communication between different silo GDAD teams is required for alignment and continuous delivery of working features or projects.

GDAD teams need to be continuously communicated with different changing to their and other dependent project(s) architectures and requirements for alignment [19]. This could be achieved with some sort of overall holistic and integrated EA [4]. Using holistic and integrated agile EA along with available communication tools may facilitate and enhance communication between GDAD teams. However, unlike traditional process-focused EA approaches, which are often considered too heavy for agile development, agile development requires an adaptive people-focused EA to provide the integrated shared view of the enterprise projects for GDAD teams [6]. This paper proposes one such agile EA driven GDAD communication model. The holistic agile EA may serve as a common information model and integrated shared view for enabling clear communication among GDAD teams [36,22]. The agile EA driven GDAD communication approach can enable communication via different architectural views at different enterprise project management levels [4,23]: (1) distributed teams share the "project solution architecture view", (2) different projects share the "program solution architecture view", (3) the same is applied to the holistic "enterprise solution architecture view", which can have "N" number of program architectures, (4) each architecture updates the architecture above, and (5) all architectures are then updated and shared from the holistic agile EA integrated shared view. This ensures that all distributed stakeholders are updated with the latest changes (i.e. project or program changes, dependencies within and across distributed projects) [4].

3.3. GDAD Performance

Researchers have diverse interpretations of software development performance. Some have referred to it as a project success [31,35]. 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, scope (functionality) and acceptable level of quality [35]. Time, budget and quality are the key components of any project’s success [35]. Others have referred to it as project effectiveness [17,27]. Project is assumed to be effective if it meets the speed, schedule and efficiency [27]. Aspects related to effectiveness are project duration, effort and quality [17].

Both traditional software development literature and agile literature have looked at software development performance dimensions as on-time completion, on-budget completion, and software functionality [3,29]. This study adopts these three dimensions of the software development performance; however, we argue that quality is an important dimension of performance. Therefore, this study refers to on-time completion, on-budget completion, functionality and quality as the four performance dimensions [14] (see table 1), which can be depicted in Figure 1. On-time completion refers to the extent to which a software project meets its baseline goals for duration [29]. On-budget completion refers to the extent to which a software project meets its baseline goals for cost [29]. Functionality refers to the extent to which the delivered software project meets its functional scope goals, user needs, and technical requirements [29]. Quality refers to delivering a good working product [14].

4. The Agile EA Driven GDAD Communication Model

The refined and updated agile EA driven GDAD communication model and related hypotheses (based on theoretical review and the preliminary expert evaluation) are shown in Figure 1. The central construct of the research model is GDAD active communication. Hypothesis 1 posits that agile EA has positive effects on GDAD communication efficiency and effectiveness, and on GDAD performance. Hypothesis 2 posits a trade-off relationship between GDAD communication efficiency and effectiveness. Finally, hypotheses 3 and 4 posit that GDAD communication efficiency and effectiveness have differential effects on the four dimensions of GDAD performance: on-time completion, on-budget completion, software functionality, and software quality.

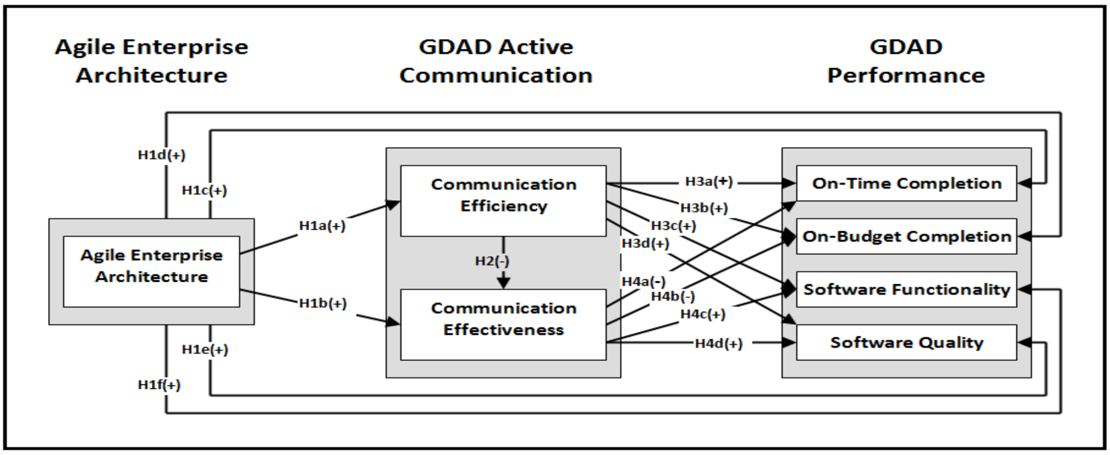

Fig. 1. The agile EA driven GDAD communication model.

4.1. *Effect of Agile EA on GDAD Active Communication*

Agile EA as an integrated shared view may provide a comprehensive view (i.e. holistic understanding and knowledge) and a common language for GDAD teams' members [8,36]. This may enhance GDAD active communication and overcome problems related to different spoken languages and different cultures [7]. As a result, communication efficiency and effectiveness may be increased. Using EA in distributed development was found to provide rich information source in large volumes [39]. This indicates that agile EA can be used as a communication mechanism enabler [39], and as a communication tool between different GDAD stakeholders [36]. Moreover, by using agile EA, as an integrated shared view (as proposed in this paper), GDAD developers can coordinate their work through interfaces of their components such that each component can be developed separately. This means that the frequency of communication as well as considering the developments of other components are decreased [36]. However, agile EA artefact should be communicated (e.g., by architect), both informally and through formal descriptions, to all GDAD stakeholders [36]. Without adequate communication and common understanding about EA among GDAD stakeholders, a project may fail technically and organizationally [38]. In a nutshell, we propose that agile EA may enhance GDAD active communication. Therefore, at a broad level, we propose the following hypotheses (1a - 1b):

Hypothesis 1a: Agile Enterprise Architecture positively affects the efficiency of the GDAD communication.

Hypothesis 1b: Agile Enterprise Architecture positively affects effectiveness of the GDAD communication.

4.2. *Effect of Agile EA on GDAD Performance*

Agile EA is important for GDAD project [6]. It draws from a uniform infrastructure, platform, application, and communicates the architecture value and status with all stakeholders [30]. Moreover, it improves implementation consistency and reduces the number of errors by providing the basis for architecture rules to involved teams [8]. Agile EA may enhance software performance as it is the placeholder for software quality, modifiability, security, and reliability [8,30]. This means that EA may have a positive impact on the GDAD performance, which means increasing the agility of GDAD project, according to agile principles [2]. Therefore, we propose the following hypotheses (1c - 1f):

Hypothesis 1c: Agile Enterprise Architecture positively influences on-time completion of GDAD project.

Hypothesis 1d: Agile Enterprise Architecture positively influences on-budget completion of GDAD project.

Hypothesis 1e: Agile Enterprise Architecture positively influences GDAD project quality.

Hypothesis 1f: Agile Enterprise Architecture positively influences GDAD project functionality.

4.3. Relationship between GDAD Communication Efficiency and Effectiveness

Considering the impacts of time, cost and effort on communication, GDAD team tends to first choose what and how much they would communicate. This choice in turn affects communication effectiveness. Furthermore, the extensively engaged GDAD team leads to more effectiveness of the communication [16]. Moreover, due to GDAD communication challenges, the message may not be received as it was effectively intended. The shortness may be insufficient to deliver clear message. In other words, efficiency may decrease the effectiveness of GDAD communication. Therefore, we propose

Hypothesis 2: GDAD communication efficiency negatively affects effectiveness of the GDAD communication.

4.4. Effect of GDAD Active Communication on GDAD Performance

The whole idea behind agility is being fast (e.g., fast delivery, fast communication). Fast communication and informal communication may lead to fast responding to customer requirements, which results in high agile development performance [12,15,35]. Delay in identifying project impacts, dependencies and resultant changes in GDAD environment may lead to longer development duration and extra cost. If the efficiency of GDAD communication is high, the amount of extra time and costs required for handling ongoing changes is minimal. This may reduce the additional time and cost, and meet the assigned time and budget targets [29]. Furthermore, as the GDAD team repeatedly implements responses to similar types of requirement changes, communication efficiency as well as optimizing and perfection of their work increase. Therefore, efficient GDAD communication is expected to effectively satisfy user requirements, which may result in high software functionality. Moreover, efficient GDAD communication may result in faster response to project changes [15]. This may help in delivering better working system (i.e. better system quality). Therefore, we propose

Hypothesis 3a. Communication efficiency positively influences on-time completion of GDAD project.

Hypothesis 3b. Communication efficiency positively influences on-budget completion of GDAD project.

Hypothesis 3c. Communication efficiency positively influences GDAD project functionality.

Hypothesis 3d. Communication efficiency positively influences GDAD project quality.

Effective communication plays a vital role in understanding customer's requirements and feedback. Yet, the higher communication effectiveness come at the price of considerably longer time and higher cost, while the shorter and faster communication come at a price of a noticeably lower effectiveness [17]. We posit that effective communication causes time and cost overruns. To effectively communicate about many different customer requirements and requirements' changes, GDAD team may need new resources and capabilities or reconfigure existing resources and capabilities [29]. This requires a considerable amount of extra time and cost [29]. Furthermore, we posit that effective communication increases system functionality and quality. That is, communication about customer's requirements and requirements' changes helps in the correctness of system configuration; improve design and product quality [11]. The functionality and quality of the system will not satisfy "up-to-date" customer needs if the team fails to embrace important changes [29]. Therefore, we propose

Hypothesis 4a. Communication effectiveness negatively influences on-time completion of GDAD project.

Hypothesis 4b. Communication effectiveness negatively influences on-budget completion of GDAD project.

Hypothesis 4c. Communication effectiveness positively influences GDAD project functionality.

Hypothesis 4d. Communication effectiveness positively influences GDAD project quality.

5. Preliminary Evaluation of Research Model

The initial evaluation of the proposed model was conducted by involving five experts from both academia and industry. Preliminary field interviews were conducted with 5 experts in agile development. Three of them were from agile development 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. Two experts were asked the questions during 60-minute semi-structured face-to-face interviews, and three experts were emailed the model and questions [20]. The asked questions included:

- Does the design of the model clear, well thought out and easy to understand?
- Does it provide the necessary (relevant and important) constructs?
- Does it provide the necessary (relevant and important) relationships between the constructs?
- Does it provide the necessary (relevant and important) hypothesis?
- Is it suitable for its intended purpose?

The feedback supports the model design and its understandability, its constructs and relationships between different variables, and its suitability for the purpose of research. One expert wrote: *“I think the model has been rigorously built and the relationships between different variables have been clearly identified”*. The feedback supports the role of agile EA and the role of the two communication dimensions; efficiency and effectiveness in GDAD. One expert mentioned: *"Investigating agile EA role in the distributed agile environment seems to be very interesting and has a lot of potential"*. One expert mentioned: *"when we talk about communication, we are assuming quick and focused message"*. We estimated some disagreement on the definitions of functionality and quality variables from the interviews. Some experts refer to functionality as a part of quality. One expert mentioned: *"...functionality is a part of quality since without achieving its functionality, software cannot be assumed of high quality"*. However, it is envisioned that functionality and quality are different concepts at this stage (subject to further research) so we included them in the model as separate variables. Moreover, a direct relationship between agile EA and GDAD performance was included in the model since some feedback assume that there is direct effect of agile EA on project performance. One expert suggested that: *"I believe EA have more effect on project performance than on communication"*. Considering all feedback, the updated model was sent via email to the same above expert group for evaluation. Based on the second feedback, we preliminary validated the Agile EA driven GDAD communication model (Figure 1) for further research.

6. Discussion, Limitations and Future Directions

This paper introduced the agile EA driven GDAD communication model. This model includes three constructs: agile EA, GDAD active communication, and GDAD performance. These constructs and their variables are presented in this paper based on the literature review and the expert evaluation. The central construct is GDAD active communication, which includes two dependent variables: efficiency and effectiveness. While efficiency refers to fast communication, effectiveness refers to quality of communication. Agile EA includes one independent variable: agile EA. GDAD performance includes four dependent variables: on-time completion, on-budget completion, software functionality and software quality. Software functionality and quality are two different concepts, as discussed in this paper. While functionality refers to meeting the goals and requirements of software project, quality refers to good working software.

This model provides a new perspective of agile EA as an integrated shared view to support GDAD communication, which is currently deemed as a gap in literature. Scaling agile approaches for GDAD environment requires scaling GDAD communication at the enterprise level to supporting multiple GDAD teams, projects and their alignment. Agile EA as an integrated shared view may provide a common language for GDAD teams' members. This means that agile EA may facilitate and enhance communication in GDAD environment. Since communication is the core of agile development, enhancing GDAD communication results in enhancing GDAD agility and performance [9]. The findings of this paper are expected to have significant implications on GDAD practitioners and academics through using agile EA as a GDAD communication enabler or tool.

Similar to any other study, this study has some limitations. One may argue that this study investigates only the effect of agile EA on GDAD and does not investigate the other communication challenges categories. This study is specially focused on the potential perspective of using agile EA, which has not been discussed before and marks the need for theoretical and empirical research. Moreover, some of the communication challenges categories (i.e. *People Differences, Distance Differences, and Technology Issues*) have been paid too much attention in the previous literature. Also, studying *Customer Communication* is out of the range of the paper, as our research focus is only on enhancing communication inter and intra geographically distributed teams working on different dependent projects in GDAD environment. In addition, we assume that *Team Issues* and *Process Issues* challenges categories will be enhanced as a result of using agile EA in GDAD. However, the above limitation keeps the door open to investigate other challenges categories such as *Team Issues* and *Process Issues*. In a nutshell, more empirical research is needed in this field.

7. Conclusions

This paper presented an agile EA driven GDAD communication model based on the literature review and preliminary evaluation. This paper draws our attention to the importance of studying agile EA and its effect on GDAD communication and performance. The proposed updated model includes three important constructs and relationships: agile EA, GDAD active communication, and GDAD performance. These constructs were rigorously identified from the previous literature and verified through preliminary evaluation. This study is one of the initial efforts to examine agile EA effect on GDAD communication and GDAD performance. We believe that many questions are yet to be answered in this area. We hope 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.

References

[1] Agerfalk, P., Fitzgerald, B. and Slaughter, S. 2009. "Flexible and Distributed Information Systems Development: State of the Art and Research Challenges," Information Systems Research (20:3), pp. 317-28.

[2] Agile Manifesto. 2001. Manifesto for Agile Software Development. http://www.agilemanifesto.org. Accessed December 20, 2014.

[3] Aladwani, A.M. 2002. "An Integrated Performance Model Information Systems Projects," Journal of Management Information Systems (19:1), pp. 185-210.

[4] Alzoubi, Y.I., Gill, A.Q. and Al-Ani, A. 2015. "Distributed Agile Development Communication: An Agile Architecture Driven Framework," Journal of Software (10:6), pp. 681-694.

[5] Alzoubi, Y.I., and Gill, A.Q. 2014. "Agile Global Software Development Communication Challenges: A Systematic Review," in the 18th PACIS14, China.

[6] Ambler, S. (2014a). Agile Enterprise Architecture. http://www.agiledata.org. Accessed October 20, 2014.

[7] Ambler, S. (2014b). Choose the Best Communication Technique Available. http://disciplinedagiledelivery.com/. Accessed October 25, 2014.

[8] Bass, L. and Kazman, R. 1999. "Architecture-Based Development (Report No. CMU/SEI-99-TR-007, ESC-TR-99-007)," Carnegie Mellon Software Engineering Institute.

[9] Balijepally, V., Mahapatra, R., Nerur, S. and Price, K.H. 2009. "Are Two Heads Better than One for Software Development? The Productivity Paradox of Pair Programming," MIS Quarterly, pp. 91-118.

[10] Beck, K. 2000. Extreme Programming Explained. Addison-Wesley Pearson Education, Boston.

[11] Bhalerao, S. and Ingle, M. 2010. "Analyzing the Modes of Communication in Agile Practices," 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), pp. 391-395.

[12] Boehm, B. and Turner, R. 2003. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley Professional.

[13] Cannizzo, F., Marcionetti, G. and Moser, P. 2008. "Evolution of the Tools and Practices of a Large Distributed Agile Team," Agile Conference. IEEE, pp. 513-518.

[14] Chow, T. and Cao, D.-B. 2008. "A Survey Study of Critical Success Factors in Agile Software Projects," Journal of Systems and Software (81:6), pp. 961-971.

[15] Cockburn, A. 2007. Agile Software Development: The Cooperative Game. Addison-Wesley, Harlow.

[16] Dorairaj, S., Noble, J. and Malik, P. 2011. "Effective Communication in Distributed Agile Software Development Teams," in Agile Processes in Software Engineering and Extreme Programming. Springer, pp. 102-116.

[17] Dyba, T., Arisholm, E., Sjoberg, D.I., Hannay, J.E. and Shull, F. 2007. "Are Two Heads Better than One? On the Effectiveness of Pair Programming," IEEE Software (24:6), pp. 12-15.

[18] Franke, U., Ekstedt, M., Lagerström, R., Saat, J. and Winter, R. 2010. "Trends in Enterprise Architecture Practice–A Survey," in Trends in Enterprise Architecture Research. Springer, pp. 16-29.

[19] Fruhling, A. and Vreede, G.-J.D. 2006. "Field Experiences with eXtreme Programming: Developing an Emergency Response System," Journal of Management Information Systems (22:4), pp. 39-68.

[20] Gable, G.G. 1994. "Integrating Case Study and Survey Research Methods: An Example in Information Systems," European Journal of Information Systems (3:2), pp. 112-126.

[21] Gill, A. Q. 2015a. "Distributed Agile Development: Applying a Coverage Analysis Approach to the Evaluation of a Communication Technology Assessment Tool," International Journal of e-Collaboration (IJeC) (11:1), pp. 57-76.

[22] Gill, A.Q. 2015b. Adaptive Cloud Enterprise Architecture. World Scientific.

[23] Gill, A. Q. (2014). The Gill Framework®. http://www.aqgill.com/adoms. Accessed October 20, 2014.

[24] Gill, A.Q. 2013. "Towards the Development of an Adaptive Enterprise Service System Model," In the 19th Americas Conference on Information Systems, Chicago, Illinois, pp. 1-9.

[25] Herbsleb, J.D. and Mockus, A. 2003. "An Empirical Study of Speed and Communication in Globally Distributed Software Development," IEEE Transactions on Software Engineering (29:6), pp. 481-494.

[26] Highsmith, J.A.I. 2000. Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House Publishing, New York.

[27] Jiang, J. and Klein, G. 2000. "Software Development Risks to Project Effectiveness," Journal of Systems and Software (52:1), pp. 3-10.

[28] Korkala, M., Pikkarainen, M. and Conboy, K. 2009. "Distributed Agile Development: A Case Study of Customer Communication Challenges," in Agile Processes in Software Engineering and Extreme Programming. Springer, pp. 161-167.

[29] Lee, G. and Xia, W. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data," MIS Quarterly (34:1), pp. 87-114.

[30] Madison, J. 2010. "Agile Architecture Interactions," IEEE Software (27:2), pp. 41-48.

[31] Mahaney, R.C. and Lederer, A.L. 2006. "The Effect of Intrinsic and Extrinsic Rewards for Developers on Information Systems Project Success," Project Management Journal (37:4), pp. 42-54.

[32] Malone, T.W. and Crowston, K. 1994. "The Interdisciplinary Study of Coordination," ACM Computing Surveys (CSUR) (26:1), pp. 87-119.

[33] McQuail, D. 1987. Mass Communication Theory: An Introduction. Sage Publications.

[34] Melo, C., Cruzes, D.S., Kon, F. and Conradi, R. 2011. "Agile Team Perceptions of Productivity Factors," Agile Conference. IEEE, pp. 57-66.

[35] Misra, S.C., Kumar, V. and Kumar, U. 2009. "Identifying Some Important Success Factors in Adopting Agile Software Development Practices," Journal of Systems and Software (82:11), pp. 1869-1890.

[36] Ovaska, P., Rossi, M. and Marttiin, P. 2003. "Architecture as a Coordination Tool in Multi-Site Software Development," Software Process: Improvement and Practice (8:4), pp. 233-247.

[37] Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P. and Still, J. 2008. "The Impact of Agile Practices on Communication in Software Development," Empirical Software Engineering (13:3), pp. 303-337.

[38] Smolander, K. 2002. "Four Metaphors of Architecture in Software Organizations: Finding Out the Meaning of Architecture in Practice," International Symposium on Empirical Software Engineering. IEEE, pp. 211-221.

[39] Svensson, R.B., Aurum, A., Paech, B., Gorschek, T. and Sharma, D. 2012. "Software Architecture as a Means of Communication in a Globally Distributed Software Development Context," in Product-Focused Software Process Improvement. Springer, pp. 175-189.

[40] Vidgen, R. and Wang, X. 2009. "Coevolving Systems and the Organization of Agile Software Development," Information Systems Research (20:3), pp. 355-376.

[41] Zachman, J.A. 1987. "A Framework for Information Systems Architecture," IBM systems journal (26:3), pp. 276-292.