

Adaptive Enterprise Architecture Driven Agile Development

Asif Qumer Gill

*University of Technology Sydney (UTS)
Sydney, Australia.*

asif.gill@uts.edu.au

Abstract

Agile development practices focus on developing and delivering working software systems in small iterations with minimal documentation. However, locally project focused agile practices overlook the need for holistic enterprise architecture. Lack of enterprise architecture in agile, especially in the large agile environments, may lead to a number of problems such as technical debt, unnecessary re-work, inconsistent communication, locally focused isolated architecture, design and implementation. There is a missing link between the enterprise architecture and agile development. Enterprise architecture is a strategic capability that should enable and enhance agility of agile development. However, organisations are not sure how best to approach strategic enterprise architecture capability for supporting agile development. This paper proposes and demonstrate the applicability of an integrated adaptive enterprise architecture driven agile development approach for large agile environments.

Keywords: Agile Development, Adaptive Enterprise Architecture, Solution Architecture

1. Introduction

Architecture is defined as the “fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution” [12]. Here, the term “system” is referred to an organisation or enterprise. Enterprise architecture (EA) is a blueprint that describes the overall structural, behavioural, social, technological, and facility elements of an enterprise’s operating environment that share common goals and principles [8]. EA is a strategic discipline, which is critical for developing and realising enterprise strategy and roadmap [18]. A sustainable EA capability is important for embracing, managing and responding to always changing business landscape [5].

An EA capability applies different methods, practices, models and tools to support the development and management of enterprise architecture process and artefacts [9]. An EA capability should not work in isolation and requires the engagement of other capabilities such as enterprise strategy, project development and service management, etc. Project development is an important discipline in information system (IS) intensive organisations. There are a number of traditional plan-based [3] and modern agile methods [1] that have been suggested by both the practitioners and academic community in the past to deal with the complex undertaking of IS development projects. Modern agile methods focus on developing and delivering working software systems in small iterations with minimal upfront design. In agile, system design is emerged from self-organizing agile teams working on the individual project iterations.

Locally project iteration focused self-organised agile teams seem to overlook the strategic EA capability. Lack of engagement of EA capability with agile development, especially in the large development environment, may lead to a number of problems such as technical debt, unnecessary re-work, inconsistent communication, locally focused isolated design [21]. EA is a strategic capability that should enable and enhance agility of agile development; and therefore, EA should not be seen as an impediment to agility in agile development [25]. An EA driven approach to agile development is required to provide a shared vision of the enterprise architecture design and provide the necessary information to guide the evolving

architecture and emergent design of different independent and dependent projects in the large scale agile environments [7, 17]. An EA driven approach seems attractive, however, the challenge is that agile teams are not sure how best to use strategic EA in agile development projects [24]. This paper addresses this challenge and proposes an integrated adaptive or agile EA [25] driven agile development approach to guide and align the agile project level architecture and design with the holistic EA.

The paper is organised as follows. Firstly, it provides the literature review and research context. Secondly, it presents the integrated adaptive EA driven agile development approach and its application. Finally, it discusses the research contributions before concluding how the findings from this study can be further used in future research endeavors.

2. Literature Review and Research Context

The objective of this paper is to investigate and propose the integration of adaptive EA and agile development. This section presents only those aspects of the adaptive EA and agile development approaches that are relevant to the context of the research presented in this paper. This section does not merely provide an overview of the different EA and development approaches. This section is strictly focused on the adaptive EA capability and its integration with agile development.

2.1. Adaptive Enterprise Architecture

The development and management of a complex enterprise architecture is not an easy task. Traditional top-down approaches to EA are continuously challenged by the dynamic business environment. Traditional top-down approaches to EA are criticized for not delivering or showing the value early as the traditional top-down approach takes few months to year to develop EA [22, 24]. Organisations need to establish an adaptive EA capability for developing and managing EA process and its artifacts for modern complex adaptive enterprises [25]. An EA capability, in contrast to traditional EA, is said to be an agile or adaptive EA when it is responsive (scans, senses and reacts appropriately to expected and unexpected changes), flexible (adapts to expected or unexpected changes 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, transformation and innovation) (based on [16]). There are a number of well-known architecture frameworks such as Zachman [20], Federal Enterprise Architecture (FEA) [4], and The Open Group Architecture Framework (TOGAF) [9] that can be used for establishing an adaptive EA capability. However, the challenge is that these frameworks are unlikely to be able to be used or adopted off-the-shelf for any specific organisation. Organisations need to tailor their own situation-specific adaptive EA capability, which can be established by selecting different EA elements from these well-known frameworks. The tailored adaptive EA capability also needs to be integrated with the development capability in the context of IS intensive organisation. The development capability is concerned about the software system needs of an organisation. The next section reviews the agile development approaches.

2.2. Agile Development

Traditional planned-based development approaches (Waterfall, Spiral, etc.) focus on detailed upfront planning, requirements analysis, architecture, design, development and deployment phases [3]. Traditional plan-based development methods work well if the software project requirements are fixed [14]. Here, the assumption is that the requirements, architecture and design are fixed or defined upfront. A lot of time and resources are spent upfront for achieving this illusion of a fixed or a complete list of requirements, architecture and design without actually delivering a single component as working software. Having said that, by the time software requirements, architecture and design are completely defined, signed off and

developed; business focus and market competition are already moved few paces further in response to changing business landscape. Organisations need to move fast and transform their business and IT services in response to always changing new business demands. This encouraged the development of new ways of working or agile methods for the iterative and incremental software delivery. There are a number of agile methods that have been proposed over the last two decades and are being continually updated, such as Extreme Programming (XP) [2], Feature Driven Development (FDD) [15], Adaptive Software Development (ASD) [10], and Scrum [19]. Agile methods are argued to have several benefits over traditional plan-based SE methods, in particular, their ability to handle volatile software project requirements that are not fixed [14]. In traditional development environment, EA follows a top-down heavy document driven approach, which is well understood by the traditional project development teams. However, agile project teams are unsure how to use EA in agile development environment since the agile development focus is on delivering working software in contrast to producing detailed upfront architecture and design documentation. This draws our attention to the following key research question:

How best to approach strategic EA capability for supporting large scale agile development?

In order to address this important research question, this paper proposes and demonstrates an integrated adaptive EA driven approach to agile development. The adaptive EA [22], contrary to traditional heavy process and documentation driven EA, fits to the agile development principles and practices. The next section discusses the research context.

2.3. Research Context

Adaptive EA driven agile development approach is developed as a part of the larger adaptive or agile enterprise research project, which is focused on developing and enhancing a framework for Adaptive Enterprises. This framework is intended to be used by the organisations to tailor an adaptive EA capability by using the elements from the existing well-known EA frameworks. This EA project is applying the design research method [11] and the “Theory Triangulation” approach [13] approach. Design research method is used for developing this adaptive framework. This framework has been developed based on the review of the existing literature published in the public domain (e.g. Zachman [20], FEAF [4], TOGAF [9], DoDAF [6]) and also empirical study conducted in the industrial settings. Theory Triangulation approach is used to analyse and extract concepts from agility, complex adaptive multi-agent systems (systems thinking), and service science theories. These concepts provided a theoretical foundation for developing the framework for the Adaptive Enterprises. Theoretical Triangulation [13] approach provided an opportunity to discover hidden meanings and include different perspectives from different theoretical viewpoints in this study. The detailed discussion of this framework is beyond the scope of this paper. However, additional details on this framework can be provided as a separate document to reviewers on request. The scope of this paper is to present the adaptive EA driven agile development approach from this framework. The next section discusses the proposed approach and demonstrates its application.

3. Adaptive EA Driven Agile Development

The adaptive EA driven agile development approach has two key parts: adaptive EA and agile development. Firstly, this section discusses the adaptive EA with the help of a well-known “TOGAF 9.1” as an adaptive EA framework example. TOGAF 9.1 is one of the most comprehensive architecture frameworks, and therefore, it has been used here as an example case to explain the Adaptive EA driven agile development approach. Secondly, it discusses the agile development with the help of a well-known and widely used agile framework “Scrum” as an example case. Finally, this section discusses the adaptive EA driven agile

development approach and its application by integrating the EA (TOGAF 9.1) and agile (Scrum) frameworks. Please note that the proposed integrated adaptive approach is not restricted to TOGAF and Scrum frameworks, we used these frameworks as a running case example to clearly explain the proposed approach. Readers may use, extend and modify this approach and then may apply in integrating their own EA and agile frameworks for their local context.

3.1. TOGAF as an Adaptive EA Framework

TOGAF 9.1 is a well-known and comprehensive EA framework from The Open Group and is used here as an adaptive EA framework example to explain and demonstrate the use of the integrated adaptive EA driven agile development approach. An adaptive EA can be divided into three major categories (based on TOGAF 9.1 – see Figure 1): Vision Architecture, Domain Architecture and Solution Architecture. TOGAF also includes a preliminary step, which is not shown in Figure 1. The key EA role's categories could be chief architect, enterprise architect, domain architect, solution architect. These architecture and role categories can be customised and additional can also be specified to suit the particular organisational context. Vision Architecture provides the summary (e.g. rolled up summary) or holistic view of the underlying details from the domain architectures. Domain Architecture is divided into further three sub-categories (based on the ArchiMate 2.0 Specifications [23] from The Open Group): Business Architecture, Application Architecture and Technology Architecture.

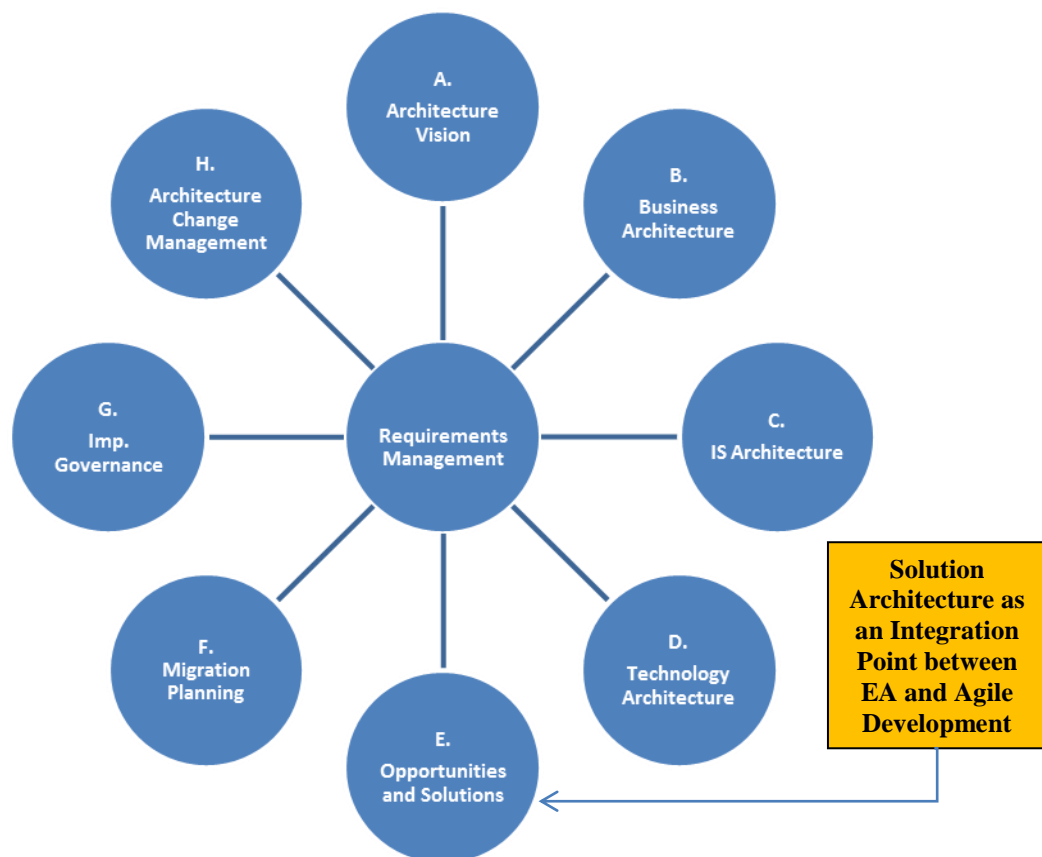


Figure 1: The Open Group Architecture Framework (based on TOGAF 9.1 [9])

Business Architecture describes the business principles, policy, strategy, organisation, capabilities, services, processes, etc. Application Architecture describes the software application principles, policy, strategy, organisation, capabilities, services, processes and their relationships that support the Business Architecture. Technology Architecture describes the

technology platform and infrastructure principles, policy, strategy, organisation, capabilities, services, processes and their relationships that support both the Business and Application Architectures. Domain Architectures are not tied to any project-specific architecture. However, a subset and combination of the Domain Architecture elements can be used to describe the Solution Architecture for particular project. Hence, Solution Architecture is mainly developed for a specific project or problem. Here, the Solution Architecture is a potential integration point between the adaptive EA and agile development project (see Figure 1). Solution Architecture from the adaptive EA can be used to guide the evolving architecture and emergent design in agile development projects [24]. However, the Solution Architecture from the EA should be adaptive and accommodate the “evolving architecture and emerging design” needs of the agile project.

3.2. Scrum as an Agile Framework

Scrum is a well-known and comprehensive framework and is used here as an agile framework example to explain and demonstrate the application of the adaptive EA driven agile development approach. Agile framework is divided into three parts (based on Scrum – see Figure 2): The Scrum Process, Artifacts and Roles. The Scrum Process is an iterative process that includes Release Planning, Sprint or Iteration Planning, Sprint, Daily Scrum Stand-up, Sprint Review, and Retrospective activities. Release planning is concerned with the planning of number of (e.g. four releases) and frequency (e.g. month, quarterly, six monthly, yearly releases, etc.) of project releases. One project release may have 1 or many sprints or iterations. Sprint or Iteration Planning is concerned with the planning of iteration in hand (e.g. prioritizing and selecting user stories for the Sprint). Sprint is concerned with the actual development of the user stories or requirements. No detailed up-front design is done in Scrum. Design is emerged during Sprints. Daily Stand-up is a short daily planning meeting of usually 10-15 minutes, where team members meet and discuss:

- What did they do yesterday?
- What will they do today?
- Are there any impediments?

Daily Stand-up meetings are mainly conducted for the daily planning and communication purposes and are a not place for reporting the work status. At the end of each Sprint, a working product is presented to the stakeholders to get feedback, which is known as a Sprint Review activity or showcasing. Sprint Retrospective meeting is a 1-2 hours meeting, where agile team members meet and discuss the Scrum process and potential opportunities for process improvement. There are three main Scrum Artifacts: Product Backlog, Sprint Backlog and Burn-down Charts. Product Backlog is a set of user stories or requirements that needs to be delivered by following The Scrum Process. Sprint Backlog is a sub-set of user stories or requirements selected from the main Product Backlog for the Iteration or Sprint in hand. Burn-down chart is used to track the progress of the Scrum project in terms of a number of user stories implemented over a period of time and a number of remaining time and user stories. In practice, Scrum teams also maintain Agile Walls that show the process and progress of the Scrum project. There are three main Scrum Roles: Scrum Master, Product Owner and Team. Scrum Master facilitates the overall Scrum project process. Product Owner is responsible for developing and managing the user story Product and Iteration Backlogs. Team is mainly responsible for actually implanting and testing the user stories from Iteration Backlog during the Sprint. Here, we can observe that there is no such activity or artifact or role related to architecture. Scrum is focused on developing and delivering working increments of the software. Software design is emerged while developing software in small increments or sprints. However, it overlooks the architecture, which is very important for avoiding or reducing the problems related to technical debt, re-work, inconsistent communication, and project focused isolated design and solution development. As discussed earlier, Solution Architecture is a potential integration point between the EA and agile

development. The next section discusses the Solution Architecture from the EA and its use in agile development such as Scrum within the overall context of adaptive EA driven agile development approach.

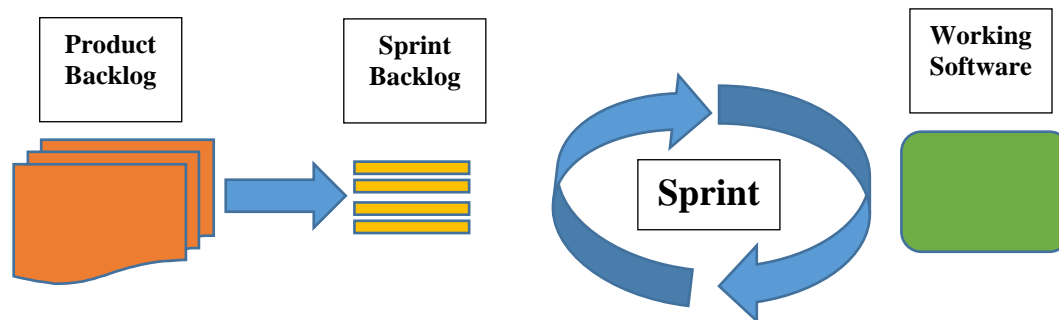


Figure 2: The Scrum Framework (based on Scrum process [19])

4. Adaptive EA driven Agile Development: Discussion and Analysis

The proposed integrated adaptive approach points out a top-down EA assets' information sharing and bottom-up EA involvement via a Solution Architecture. It has two parts: EA and agile development. These parts are divided into three streams: UpStream, MidStream and BottomStream (see circles in Figure 3). The integration point between the EA and agile development is the midstream Solution Architecture (see Figure 3). The Solution Architecture is adaptive in nature instead of a traditional fixed and upfront detailed Solution Architecture. The adaptive Solution Architecture in the midstream would enable agility in both the upstream (Vision and Domain Architectures) and BottomStream (agile development).

The integrated adaptive approach suggests that the adaptive Solution Architecture, as a part of the larger EA capability (see Figure 3), can be evolved and used before, during and after each Scrum Sprint or Iteration. For instance, an initial high-level adaptive Solution Architecture can be developed by combining existing elements from the EA Domain Architectures to address the specific user stories or requirements before the execution of Scrum Sprints (e.g. PreSprint). Here, the EA driven agile development approach is suggesting re-using the EA assets (e.g. Domain Architecture elements such as business process models, application communication diagrams) for developing user stories and the initial adaptive Solution Architecture. The high-level adaptive Solution Architecture would then evolve as the design is emerged during different project iterations (e.g. Sprint). The adaptive Solution Architecture should be flexible and enable the design emergence during each Sprint. It should be reviewed and be updated or re-factored after each Sprint (e.g. PostSprint), if required, to reflect any changes due to software development Sprint.

The elements from the evolving adaptive Solution Architecture should be continuously reflected in the Domain Architectures and the overall summary Vision Architecture to maintain the holistic view of the EA. It can be observed that the Vision, Domain and Solution Architectures would evolve as the Scrum process is executed at the project level. As discussed earlier, most of the agile Scrum teams, in practice, maintain the physical/and or virtual Agile Walls to display and communicate the project activities and progress status (e.g. in backlog, in progress, done). Agile teams actively use the Agile Wall in discussing, tracking and managing the project progress. Agile Scrum locally project focused teams do not have any visibility or information related to EA on their Agile Walls. The Agile Wall should display the Vision (summary or overall EA) and adaptive Solution Architecture (e.g. project specific) diagrams or information along with the traditional Agile Wall contents. Agile Wall with Vision and adaptive Solution Architecture will provide a shared vision to both on-site and off-site (geographically dispersed locations) agile development teams. Agile Wall with EA information would help the agile teams to analyse and link their user stores to evolving

architecture and emerging design within the overall larger context of EA. Hence, the EA driven agile development approach suggests establishing and using the integrated Adaptive EA Wall.

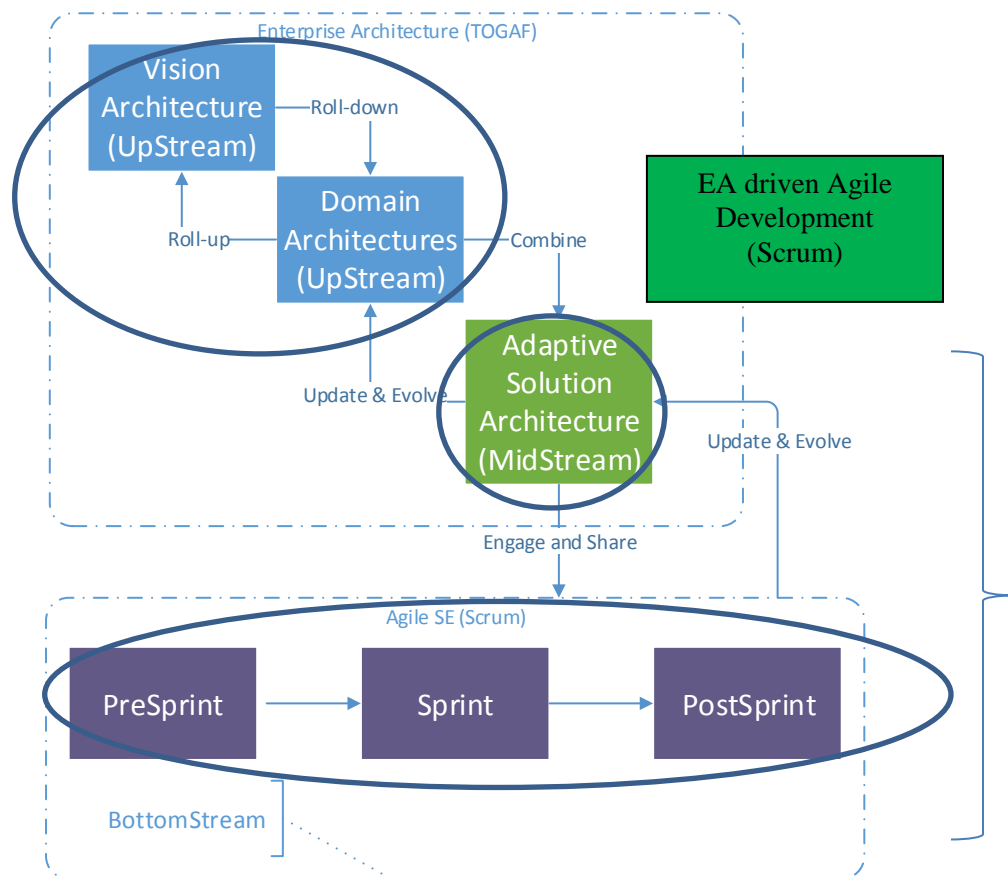


Figure 3: Integrated Adaptive EA driven Agile Development Approach

The inclusion of the adaptive Solution Architecture artifact in agile development (see Figure 3) has a number of implications. Firstly, it draws our attention to the need for a new role at least at the project level, which is called an “Architecture Owner.” Architecture Owner should actively engage and share information about EA with the agile development teams. This is in contrast to developing and providing a traditional detailed upfront Solution Architecture document to traditional project development teams. Secondly, Architecture Owner would require participatory architecture design approach (e.g. engage Scrum Master, Product Owner, Team etc.) instead of working in isolation when developing and updating the adaptive Solution Architecture. For instance, Architecture Owner can work with the Product Owner in identifying the user stories or requirements. Essentially, Product Owner’s user stories need to be linked to the architecture. The impact of the user stories needs to be looked into through the lens of the relevant solution architectural building blocks. Architecture integrated with the user stories would not only result in a less technical debt, but it will also provide a clear link between the EA and agile development. Architecture Owner should be able to accommodate requirements, skills and experiences of the agile development teams in the adaptive Solution Architecture design. The Architecture Owner, in the down-stream, may attend Sprint Planning, Daily Stand-up, Sprint Review and Retrospectives Meetings; and provide the updated information about the EA assets (via Adaptive Solution Architecture) to help the agile development teams to stay focused and make quick and effective informed decisions about the project planning and design in the overall context of EA. Finally, Architecture Owner, in the up-stream, can actively engage with the Domain and Enterprise Architecture Owners and provide the updated information regarding the EA assets (via

adaptive Solution Architecture). In summary, the purpose of the adaptive EA driven agile development approach is to encourage the participatory architecture design (PAD) practices and keep the agile development teams focused on the work-in-hand (developing working software) while keeping the bigger EA picture alive in their minds and on their agile Walls.

5. Conclusion

In the traditional top-down EA approach, it takes few months to years to develop an effective EA capability and demonstrate its value. This paper discussed the need for an adaptive EA in the context of agile development; and proposed an integrated adaptive EA driven development approach. This paper explained the integrated adaptive approach and demonstrated how the adaptive Solution Architecture from the adaptive EA capability can be used in the agile development environments. The agile development teams will not be burdened by the adaptive Solution Architecture development and updating activities as these activities will be performed by the “Architecture Owner” by applying the participatory architecture design approach. The Architecture Owner will facilitate the top-down EA assets’ information sharing and bottom-up EA evolution via an adaptive Solution Architecture. The integrated adaptive approach showed that in order to establish an adaptive EA in the up-stream, and agile development in the down-stream, we need an adaptive Solution Architecture as an integrator in the mid-stream. The adaptive Solution Architecture is a pivot integration point in lining the holistic strategic EA capability with the local project solution-focused agile development. In future, the adaptive approach meta-model will be developed and evaluated by the means of an empirical study.

Acknowledgements

The author would like to thank to colleagues both from academia and industry who helped him with their valuable feedback and experience in developing the integrated adaptive approach presented in this paper.

References

- [1] Agile Manifesto 2001. Manifesto for Agile Software Development, <http://agilemanifesto.org/>.
- [2] Beck, K. and Andres, C. 2004. *Extreme Programming Explained: Change*, Addison-Wesley. Boston.
- [3] Boehm, B.W. 1988. ‘A Spiral Model of Software Development and Enhancement’, *Computer, IEEE Journal*, vol. 21, no. 5, pp. 61-72.
- [4] CIO Council. 2001. *A Practical Guide to Federal Enterprise Architecture*. Feb 2001.
- [5] Doucet, G., Gotze, J., Saha, P., and Bernard, S. 2008. *Coherency Management: Using Enterprise Architecture for Alignment, Agility, and Assurance*, *Journal of Enterprise Architecture*.
- [6] DoD, DoDAF V2.0. 2010. http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf
- [7] Gill, A. Q. 2015. *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), 57-76.
- [8] Gill, A. Q. 2013. *Towards the development of an adaptive enterprise service system model*. *Proceedings of the Nineteenth Americas Conference on Information Systems*, Chicago, 1-9.
- [9] Harrison, R. 2011. *TOGAF Foundation*. The Open Group.
- [10] Highsmith, J. 2002. *‘Agile Software Development Ecosystems’*, Pearson Education, Inc., Boston.

-
- [11] Hevner, A.R., March, S.T., Park, J., and Ram, S. 2004. Design Science in Information Systems Research, *MIS Quarterly*, 28, 1, 75-105.
- [12] ISO/IEC 42010 .2007. Defining architecture. <http://www.iso-architecture.org/ieee-1471/defining-architecture.html>.
- [13] Patton, M.Q. 2002. *Qualitative Research and Evaluation Methods*. Thousand Oaks, CA: Sage Publications.
- [14] Paetsch, F., Eberlein, A. & Maurer, F. 2003. 'Requirements Engineering and Agile Software Development', Proceedings of the IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Linz, Austria, pp. 308 -313.
- [15] Palmer, S.R. & Felsing, J.M. 2002. 'A Practical Guide to Feature-Driven Development', Prentice-Hall Inc, Upper Saddle River.
- [16] Qumer, A. and Henderson-Sellers, B. 2008. An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering. *Journal of Information and Software Technology (IST)*. Volume 50, Issue 4 (March 2008), Pages 280-295.
- [17] Henderson-Sellers, B., & Qumer, A. 2007. Using method engineering to make a traditional environment agile. *Cutter IT Journal*, 20(5), 30.
- [18] Ross, J.W., Weill, P., and Robertson, D.C. 2006. *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*, Harvard Business Review Press, pages 256.
- [19] Schwaber, K. 2007. 'The Enterprise and Scrum', Microsoft Press, Washington.
- [20] Zachman, J.A. 1987. A framework for information systems architecture. *IBM Systems Journal*, 26,276-292.
- [21] Ozkaya, I., Nord R.L., Bellomo, S. and Brayer, H. 2013. Beyond Scrum + XP: Agile Architecture Practice. *Disciplined Agile Delivery in the Enterprise*, The Journal of Information Technology Management (Cutter IT Journal). 2013. Vol 26, no. 6.
- [22] Edwards, C. 2006. Agile Enterprise Architecture – Part 1”, USA: ProcessWave. <http://www.agileea.com/Whitepapers/2006-12-14-AgileEnterpriseArchitectureV1.00-Part1.pdf>.
- [23] The Open Group. 2013. ArchiMate.<http://pubs.opengroup.org/architecture/archimate2-doc/>.
- [24] Madison, J. 2010. Agile Architecture Interactions, *IEEE Software*, 27(2), (2010), 41-48.
- [25] Bokang, M. 2012. A Framework for the Development and Measurement of Agile Enterprise Architecture, MC Thesis, Rhodes University.