Towards an Agile Design of the Enterprise Architecture Management Function

Sabine Buckl, Florian Matthes, Ivan Monahov,
Sascha Roth, Christopher Schulz, Christian M. Schweda
Technische Universität München (TUM)
Chair for Informatics 19 (sebis)
Boltzmannstr. 3, 85748 Garching bei München, Germany
{sabine.buckl, florian.matthes, ivan.monahov
{sascha.roth, christian.m.schweda, christopher.schulz}@mytum.de

***Abstract*—Enterprise architecture (EA) management provides an engineering approach for the continuous advancement of the enterprise as a whole. The high number of involved components and their dense web of interdependencies nevertheless form a major challenge for such approach and demand high initial investment into documentations, communications, and analysis. Aforementioned fact has in the past been an impediment for successful EA management in practice. In the field of software engineering recently lightweight and *agile* methods have become more and more important. These methods aim at quickly creating results, while staying flexible in respect to the design goals to attain.**

In this article we explore to which extent the de-facto standard for agile methods, namely *Scrum*, can be applied to EA management. Thereby, we derive challenges for an agile EA management approach and revisit current approaches regarding their agility. Finally, we outline how agile EA management can be implemented based on the method of Scrum.

***Keywords*-enterprise architecture, EA management, agile methods, scrum**

I. Introduction

In recent years *enterprise architecture* (EA) and its management have gained considerable attention from the academic as well as practical audience. EA management targets the enterprise in an embracing manner and seeks to evolve the enterprise to facilitate the alignment of business and information technology (IT). Thereto, EA management provides specific EA products in the sense of van der Raadt and van Vliet [30], namely *enterprise architectures* and *EA policies*. The architectures describe “the fundamental organization of [the enterprise] embodied in its components, their relationships to each other, and to the environment” [14]. Distinct states of an EA form the products of EA management the *current state* describing the status quo, the *target state*, forming an envisioned long-term picture, and intermediate *planned states*. The policies delineate standards, rules, and guidelines for developing the EAs. The embracing nature of the EA and the volatile environment, in which EA management takes place, rises typical challenges in creating these EA products, whose stakeholders come from diverse backgrounds and commit to diverse terminologies.

A similar situation applies in the context of software development, where diverse stakeholders with domain-specific backgrounds must work in cooperation to attain the development of a specific software product. “Classic software development” is in this sense confronted with an unstable environment that hampers the effective development of products. Agile project management focuses on the implementation and delivering of projects in a highly flexible and interactive manner [13]. In particular the agile development of software products seizes this suggestion even if current literature often applies this term as a synonym to agile project management [11].

Agile software development applies an iterative and incremental development method, where software requirements and solutions evolve through collaboration between stakeholders and product developers. Initiated and furthered in the past two decades by Cockburn, Coplin, Highsmith, Schwaber, and Beck [8], [13], [24], the approach itself is rooted in the “Toyota Production System” [19], [17] and knowledge management [11]. Up from the year 2001 when the “Manifesto for Agile Software Development” was released [5], it is notable to see that agile project management for the development of software and in particular *Scrum* has made its way into small to mid-sized enterprises, universities, but also into global acting software engineering companies (e.g. [16], [27]). In being one famous representative for agile software development, Scrum can be considered as one of the vanguards of the new way to buy and manage software development when business conditions are rapidly changing [25].

Against this background, it seems worthwhile to consider, whether agile development methods can also be applied in the context of EA management . In this article, we investigate this topic in more detail by answering the following two research questions:

1) What typical challenges of EA management can be addressed applying agile methods, e.g. provided by Scrum?
2) Do prevalent EA management frameworks and approaches account for agile methods?

The article is structured as follows: In Section II we outline typical challenges of EA management as found in recent literature. We subsequently suggest how agile methods, especially from Scrum, can be beneficially employed for an EA management function. In Section III we discuss to which extent prevalent EA management frameworks and approaches incorporate agile principles in their structure. The short literature review confirms that there is only limited research so far with regards to the topic. Section IV outlines an agile approach to EA management based on the agile method of *Scrum* along a brief example. Lastly, Section V presents several hints towards further research areas.

II. Challenges for EA management and their agile solutions

This section outlines four major challenges for EA management, as found in current literature [32]. Afterwards, suggestions spawned by the agile software development literature and addressing these challenges are presented. Central to our subsequent considerations are the notions of *stakeholder*, who has a specific interest in the EA to be addressed, and *information provider*, who can provide information about a part of the EA. The *enterprise architect* contributes by processing the received information to the intended EA product (architecture description) that the stakeholders need.

A. Challenges

A key challenge in EA management endeavors is the decoupling of requirements on the one hand and EA products on the other hand. Van der Raadt et al. [29] allude to this as the "ivory tower syndrome", when a complex, abstract, and over-sized EA model is created [7] "for the modeling's sake" [1]. This especially arises, as stakeholders and information providers have their own languages, which impedes the communication between these groups [10]. Such plurality of languages is, as Lankhorst et al. discuss in [15], a frequent phenomenon caused by the multi-disciplinary background of EA stakeholders. A lack of a shared understanding or misunderstandings between the stakeholders can easily lead to low stakeholder satisfaction given a high modeling effort.

Challenge 1 (C1): The EA management endeavor has to be aligned with the stakeholders' interests expressed in a shared terminology.

Another issue is the time dimension: starting and fostering EA management has to be considered as medium to long term investment. Different figures about the payoff of EA management endeavors exist in the literature, but as EA management entails cultural and technical changes throughout the enterprise, we agree with the estimation of Ross [23], expecting minimum two years for realizing the full-scale benefits of EA management. During the period of the build-up, high workload in information gathering is created, most often without immediate results being visible. This in turn is likely to lead to dissatisfied information providers who regard their efforts as wasted.

Challenge 2 (C2): An EA management endeavor has to ensure an early and periodical delivery of concrete EA products.

EA management activities are often understood as additional projects instead of being understood as continuous management support [4]. Information providers already engaged in their day-to-day duties and projects are likely to experience reoccurring and sometimes tedious documentation tasks as overhead. In addition, as discussed by Buckl et al. in [6] the information provides performing the labor-intensive documentation tasks are not identical with the stakeholders benefiting from the gathered data and thus cannot see the advantage of the work they are demanded to perform. This results in a lack of commitment from this relevant group of people, thus depriving the EA management endeavor of the necessary information sources. Low buy-in of stakeholders not seeing their specific interests addressed is a common consequence [4], leading to low overall acceptance for EA management in the enterprise.

Challenge 3 (C3): An EA management endeavor has to ensure commitment and involvement of all parties.

EA management endeavors operate in a highly volatile environment, where both stakeholders and their corresponding EA interests change. As Lucke et al. analyze in [18] this is caused by changing conditions under which EA-related problems have to be addressed. In addition changes pertaining to the enterprise itself, as shifts in the market environment, new technologies, or novel regulations [31], can seriously impact the EA and its stakeholders' interests. "Stakeholders impacted by transformation results" can, as Op't Land et al. put it in [20], react with changing EA-relevant interests. Thereby, they can change the level of detail expected from the EA products, but can also re-scope the entire EA management initiative.

Challenge 4 (C4): An EA management endeavor has to continuously adapt to a volatile environment with changing criteria for goal fulfillment.

B. Ideas for agile EA management

Agile methods offer different techniques and principles that are useful to address challenges as the ones raised above. The *pull-principle* as discussed in [12] means that any product creation is driven by the stakeholder's actual demands. For an EA management endeavor, this implies that an EA model is created to satisfy the specific interests of the stakeholders. Therefore, a shared vision between the producer and the consumer of the information is created on a daily-basis [5]. Agile development methods are aware of problems like wrong scoping and cope with this topic in introducing the principle of impediment reduction [12] rooted in the lean development of Toyota [19]. Mapped on a EA management, problems arising throughout the design and implementation of the function should be addressed instantaneously. The close interaction on a daily basis and the periodical delivery of products [25] further facilitates the early pay off of the EA management endeavor and makes it more easy to communicate the benefits of the EA management endeavor.

Agile method in general and especially *Scrum* build on the values and principles of *commitment*, *focus*, *openness*, *respect*, and *courage* [25]. To promote these fundamental values, the agile methods motivate the *one-piece* flow, meaning that there is no interference during the work on the product [25]. With regards to an EA management function, the conscientious consideration of these values and the undisturbed implementation of the function can help to address the challenge of commitment and involvement. Further applying the idea of autonomous self-directed [19] and self-organized [5] work teams can be useful to keep administrative overhead low. The incremental nature of agile development allows to develop different products of EA management at different rates [8], while iterative development facilitates quick reaction on changing environment. Agile approaches strive for continuous improvement [12], [25] by speeding up the plan-do-check-act cycle introduced by Deming [9] and Shewhart [26].

III. Related Work

When searching for the key words 'enterprise architecture (management)' and 'agile' only a few scientific publications are found. However, this topic seams to spark interest by many practitioners and consultants, since there are many white papers and blog discussions available. Therefore we will investigate both sources – scientific papers as well as practitioner papers in this article. In section II-A we defined four challenges (C1, C2, C3 and C4) for EA management and their agile solution. We will examine which of these challenges are addressed by the suggested solution from each literature source reviewed.

In [22] Rhubart discusses that both the enterprise architecture and the agile development methodologies are decision-making frameworks and therefore share the same common ground. The author differentiates between top-down architecture, which works toward a long-range term, and a bottom-up architecture, which addresses questions from ongoing project that have to be answered quickly. Thereafter stakeholders are interviewed regarding the idea of agile EA management. However, no explicit hints are provided about how to integrate agile techniques within existing EA approaches and thus all four challenges from section II-A remain unsolved.

Scott W. Amber outlines in his white paper [2] typical problems in practice with existing EA approaches, e.g. project teams don't know the EA exists, the EA is outdated. Afterwards he introduces an 'agile model driven development approach (AMDD) at the enterprise level', where first an initial architecture vision is created and is then communicated to the architecture stakeholders and to the developers. Thereafter, the feedback is collected and the architecture is updated. However, the AMDD approach does not provide information about how to ensure a shared terminology (C1). The approach does not define a clear time-frame for the results of iterative cycles (C2). Further, the framework does not provide prescriptions about how to gain the commitment from the involved parties, i.e. the different stakeholders and information providers (C3). However, the paper contains many recommendations which remain on a very high and abstract level.

Kaisler et al. investigate on [3] the transition and the implementation planning of EA projects. This paper introduces a framework for supporting both the transaction planning and the implementation planning of a target EA. The authors claim, that companies often plan more activities for a transition phase than resources available. Therefore, they suggest, that the enterprise architects team should use an agile approach in which it determines which project has to be proceeded in the current iteration phase or to be deferred. On the one hand, no information is provided by this framework about how to ensure early delivery (C2). On the other hand, the framework enables the architects to quickly respond to moving targets (C4). However the paper does not make clear prescriptions about how to gain the commitment of the involved parties (C3) and how to ensure a shared terminology (C1).

In Pulkkinen et al. [21], the authors claim that many existing EA process models represent different abstraction levels and suggest a cyclic approach, without any explicit definition of the phases. Moreover, the authors describe three different decision-making levels and four main architecture types. They suggest a cyclic EA process covering the macro level development going through all decision making levels and reviewing all enterprise architecture types. The authors then provide a case study as evaluation of their framework. The framework does take the stakeholders input into account, but only sketches the challenge of a shared terminology (C1) and how to ensure the commitment of the involved parties (C3). A clear definition of how to ensure early and periodical delivery of concrete EA products is not provided (C2). Last but not least, no clear prescriptions are made about how to adapt to moving targets (C4).

The Open Group[1] published the current version 9.0 of the *The Open Group Architecture Framework (TOGAF)* in October 2009 [28]. The most-known part of TOGAF is the *architecture development method* (ADM), which describes an iterative process consisting of eight phases, which are

[1] See http://www.opengroup.org/overview, last accessed 2011-03-29

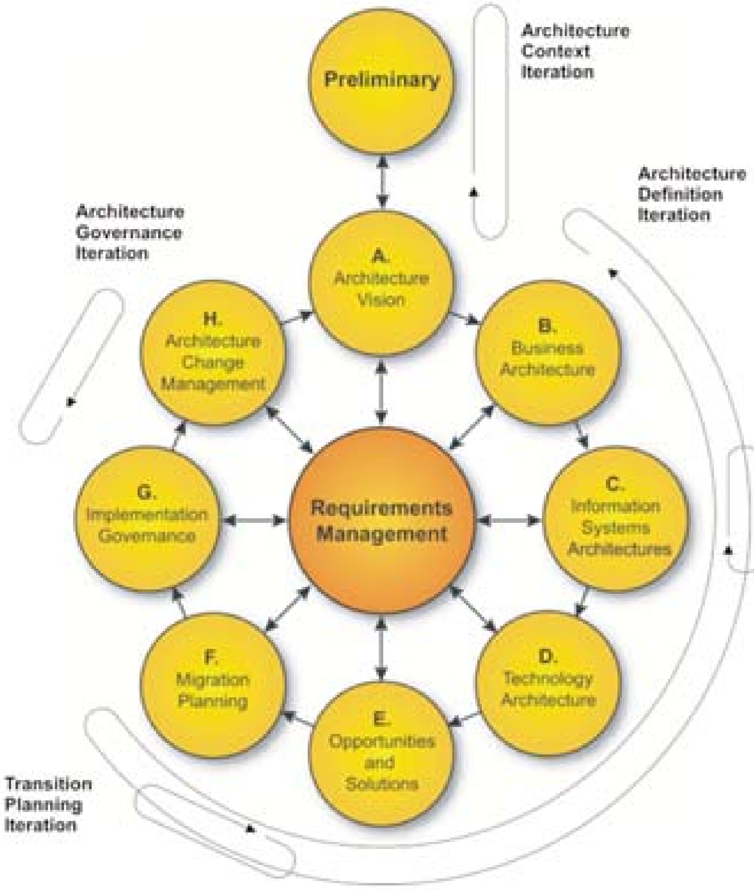
Preliminary
Architecture Context Iteration
Architecture Definition Iteration
Architecture Governance Iteration
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management
Transition Planning Iteration

Figure 1. ADM cycle and iteration cycles

complemented by a preliminary preparation phase and the central activity of requirements management. The framework defines an explicit stakeholder management and the enterprise content metamodel provides a terminological basis. The terms used therein are neither organization-specific nor complement with an in-depth definition (C1). Further, TOGAF defines artifacts as result after each phase of the ADM cycle. This framework even provides an approach for applying iterations to the ADM cycle. Four iteration cycles are presented and a mapping of TOGAF's ADM phases to the defined iteration cycles is proposed as shown in Figure 1[2]. However there is no clear time-frame defined for the iterations (C2). TOGAF refers to an explicit commitment of all stakeholders, but does not make prescriptions how to gain this commitment (C3). Regarding C4, TOGAF introduces a continuous requirements management, but fails to provide a continuous adaptation activity for the EA management process. Such adaptations are performed "after the fact" in phase H ("Architecture change management") in the ADM cycle.

IV. An Agile Approach to EA Management

In the following we outline an agile approach towards EA management building on the agile software development method of *Scrum* [25]. Figure 2 illustrates the iterative nature of the development method based on the plan-do-check-act cycle of Deming [9] and Shewhart [26]. In Scrum, a *product owner* defines the *product backlog* stating the requirements for the product to be achieved. In cooperation with the *scrum team* estimations about the effort necessary to build the product are made. It is crucial that the *Product Owner* represents the voice of the *customer* and ensures that the *Scrum Team* delivers value to the customer's business. Therefore, the *Product Owner* writes so-called user stories, prioritizes them, and adds them to the *Product Backlog*. Then, the *Product Owner* presents the top priority *Product Backlog Goal* to the *Scrum Team*.

Translated into EA management, a *Product Owner* has to represent the *EA Stakeholders* interested

[2] Source: http://pubs.opengroup.org/architecture/togaf9-doc/arch/Figures/19_adm_iteration.png

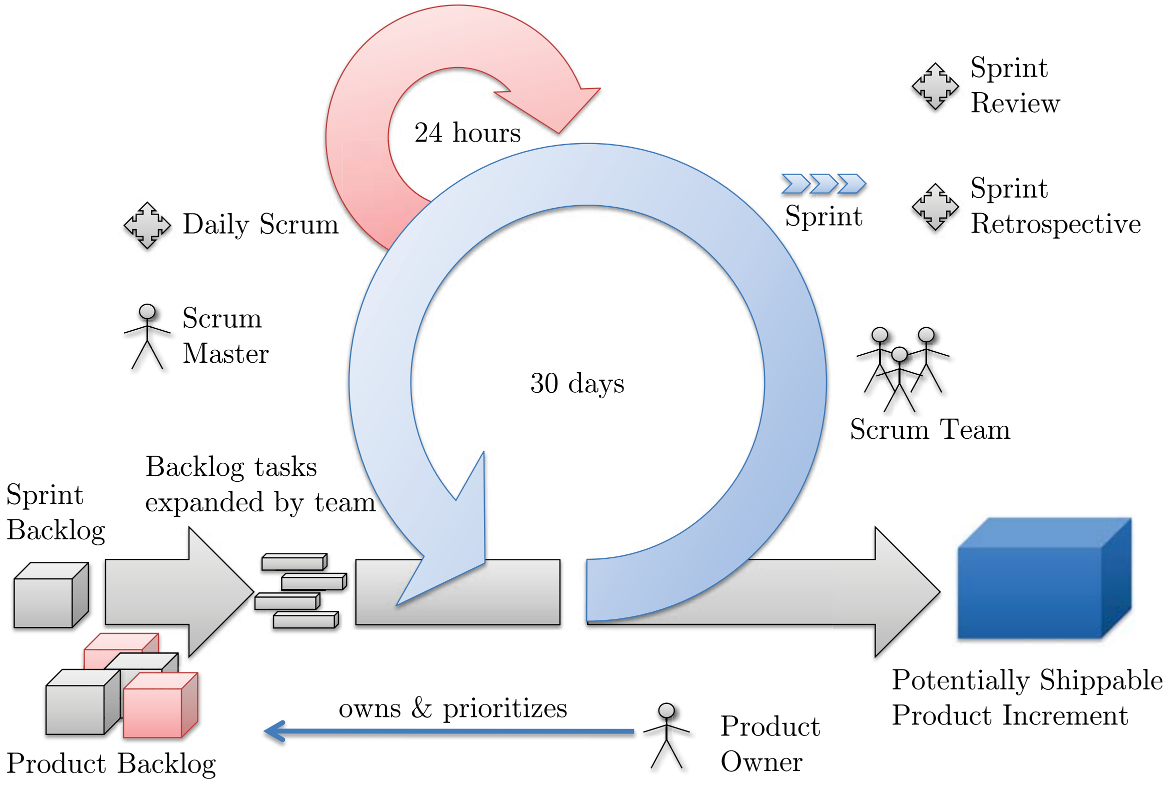

Figure 2. Overview of Scrum (adapted from [25])

> in gathering information about the EA. Thus, the different *EA Stakeholders* represent the *customer*. The *Product Owner* has to be a person uniting knowledge of EA management and organizational units in order to refine the *Product Backlog* in close cooperation with the *EA stakeholders* to capture the actual demands (cf. *C1*). This *Product Backlog* for an EA has to contain the different EA products, which are requested by the stakeholders commonly supporting the decision making process. Based on the business impact a decision implies, the *Product Owner* prioritizes the different *Product Backlog Goals* in order to identify the top priority *Goal*. Also, the *Product Owner* has to estimate the effort that an item in the *Product Backlog* will take. That means the *Product Owner* talks to 1) the relevant employees possibly having the information at hand, the information providers, or knowing where it resides and 2) the *Scrum Team* which is concerned with collecting and analyzing EA products.

The *Scrum Team* selects an item from the *Product Backlog* and crafts a *Sprint Goal*. Thereto, the *Sprint Goal* selected is expanded in a *Sprint Backlog* specifying work packages that have to be performed to achieve the *Sprint Goal*. The *Scrum Team* "has full authority to do whatever it decides is necessary to achieve the [Sprint] Goal" [25].

> Contrarily, in EA management, a *Scrum Team* is represented by enterprise architects involved in collecting information about the EA. These enterprise architects are individuals with an interdisciplinary background, who should act autonomously in order to achieve a *Sprint Goal*. The *Sprint Goal* could be derived by deciding how many EA products are addressed at once, e.g. EA information is iteratively collected (cf. *C2*) for a cross-grained information model. That means, a piece of information is gathered in order to support the decision making process whereas the information and its underlying information model is refined in each iteration. Within the *Sprint Backlog*, it should be defined how to gather the relevant information to fulfill the *Sprint Goal*, e.g. gather information by interviews.

The *Scrum Master* is responsible for the success of Scrum. If decisions are necessary during a Daily Scrum, the Scrum Master decides immediately even on uncertainty. Note that the *Scrum Master* is not part of the *Scrum Team* detailed below.

> In EA management, a *Scrum Master* is an experienced *Enterprise Architect* guiding the *Scrum Team*, which is meant to develop EA products derived from the information provided by EA information providers. The *Scrum Master* decides, e.g., how to react, if certain information is missing or cannot be gathered because of, e.g., legal issues or uncooperative employees in their role as EA information providers. Early decisions of the *Scrum Master* maintain a clear focus of the *Scrum Team* (cf. *C4*).

The *Scrum Team* commits itself to implement a specific *Sprint Goal* selected to achieve during *Sprint*. A daily meeting called the *Daily Scrum* focuses on 1) the development of the day before, i.e. what the team members actually achieved in the last iteration 2) the current development, i.e. what the team members plan to achieve today, and 3) what problems the team members currently see.

> In EA management, a *Scrum Team* also commits to a specific *Sprint Goal*. Due to the communication of issues and what exactly has been done on the day before, the *Scrum Master* receives early feedback in the *Daily Scrum* meeting and may decide, if interventions or changes are necessary (cf. *C4*), i.e. the *Scrum Master* decides how to react, if certain information is missing or cannot be gathered.

The *Sprint* focuses on development activities which can be achieved in 30 days aiming to deliver a potentially shippable product to the customer. Therefore, integration and testing has to be optimized to be able to deliver quickly. At the end of a *Sprint*, the *Sprint Goal* should be reached.

> Transfered to EA management, reaching a *Sprint Goal* means having enough information, i.e. an EA product, collected to support a decision. Whether or not the EA product is able to support the customer in a particular decision making process is thereby decided by the *Product Owner*. Again, the *Product Owner* represents the single point making sure everyone's needs are met (cf. *C1*).

After each *Sprint*, the *Scrum Team* presents the product increment during a *Sprint Review* meeting.

> Mapped to EA management, a *Scrum Review* is concerned with presenting the developed EA product to the management, customers, and the *Product Owner*, i.e. all involved stakeholders (cf. *C3*).

During a *Sprint Retrospective* meeting [24], the *Scrum Team*, the *Scrum Master*, and optionally the *Product Owner* discuss what went well during a *Sprint*, and aspects that could be improved.

> In EA management, a *Scrum Retrospective* meeting is concerned with eliminating problems, e.g. avoiding waste, which leads to continuous improvement of the EA management function (cf. *C4*).

Agile methods, as Scrum require cultural change [5], [25], [24].

> For a successful EA management, a cultural change is also required. Core values of Scrum, i.e. commitment, openness, focus, respect, and courage should become corporate values. A common foundation Scrum and also EA management rely on is delivering business value to customers. Thus, the cooperate culture has to be open for transparency and commitment to the EA endeavor (cf. *C3*).

V. Conclusion and further work

The present article examined four major challenges EA management is often confronted with before proposing tangible principles and techniques of agile methods coping with those particularities. In doing so, both the challenges as well as the methods addressing them originate from current literature in the field of EA management, agile software development, and agile project management. To date, there exists only a few publications which suggest concrete artifacts helping to perform EA management in an agile spirit. This contribution analysis related work and reveals whether and how prominent EA and EA management approaches like TOGAF already account for an agile management of an EA. Based on a popular agile software development approach called Scrum, the article afterwards presents pointers towards an agile EA management in mapping Scrum concepts (roles, activities, deliverables) to their counterparts in EA management. Thereby, it is shown to which extent above identified challenges are met by the agile approach.

Since the agile EA management approach has not been evaluated yet, the contribution is presently in the state of research in progress. Roles, duration, activities, desired results, e.g. are directly borrowed from Scrum. Their applicability and utility however have to be subsequently validated in the context of EA management. After having sketched how EA management can be performed in a more agile and lightweight manner, we are currently presenting the idea to different industry partners all facing one or more of the priorly described challenges. As a start, a small, well-defined, and independent EA management problem could serve as a valuable proof of concept for our approach helping to sharpen the responsibilities of roles, to detail the definition of activities, as well as to specify and adjust the content of necessary deliverables, i.e. EA products.

This article examined only a limited amount of EA management frameworks and approaches with regards to their capacity to embrace agile principles. On the one hand, future work should consider a larger number of current EA management literature, on the other hand this literature should be considered in greater detail. Such a literature survey could make use of a predefined ”agile requirements scheme” onto which these approaches are mapped.

Priorly mentioned input from industry combined with the knowledge gained from in-depth literature studies allow for a refinement of the types of stakeholders and roles (e.g. scrum master, product owner) necessary to effectively implement the agile EA management approach.

Lastly, we vividly advice to compare the results and key characteristics (e.g. time required, money spent, stakeholder satisfaction rate) of an EA management approach with the one not benefiting from agile techniques and principles. In the best case, a similar problem is tackled twice once with and once without the application of our suggested agile approach. In particular, we assume that the introduced method could prove to be worthwhile during transition phases where the EA management function is adjusted. However, this remains to be demonstrated in the course of future studies.

REFERENCES

[1] S. W. Ambler. Enterprise modeling anti-patterns. http://www.agilemodeling.com/essays/enterpriseModelingAntiPatterns.htm (accessed at 2011-06-08), 2008.

[2] S. W. Ambler. Agile enterprise architecture. http://www.agiledata.org/essays/enterpriseArchitecture.html (accessed at 2011-24-03), 2009.

[3] F. J. Armour and S. H. Kaisler. Enterprise architecture: Agile transition and implementation. *IT Professional*, 3:30–37, November 2001.

[4] F. J. Armour, S. H. Kaisler, and S. Y. Liu. Building an enterprise architecture step by step. *IT Professional*, 1:31–39, July 1999.

[5] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, and D. Thomas. Manifesto for agile software development. http://agilemanifesto.org/ (cited 2011-15-03), 2001.

[6] S. Buckl, A. M. Ernst, J. Lankes, F. Matthes, and C. M. Schweda. State of the art in enterprise architecture management 2009. Technical report, Chair for Informatics 19 (sebis), Technische Universität München, Munich, Germany, 2009.

[7] S. Buckl, A. M. Ernst, F. Matthes, and C. M. Schweda. How to make your enterprise architecture management endeavor fail! In *Pattern Languages of Programs 2009 (PLoP 2009), Chicago*, 2009.

[8] A. Cockburn. *Agile Software Development: The Cooperative Game*. Addison-Wesley Professional, Boston, MA, USA, 2nd edition, 2006.

[9] E. W. Deming. *Out of the Crisis*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1982.

[10] D. Dreyfus. Information system architecture: Toward a distributed cognition perspective. In *ICIS 2007 Proceedings*, 2007.

[11] B. Gleichauf. Planung der unternehmensarchitektur - rückblick und ausblick. Presentation, 2010.

[12] B. Gloger. Scrum. *Informatik-Spektrum*, 33(2):195–200, 2010.

[13] J. Highsmith. *Agile Project Management: Creating Innovative Products*. Addison-Wesley Professional, Boston, MA, USA, 2nd edition, 2009.

[14] International Organization for Standardization. ISO/IEC 42010:2007 Systems and software engineering – Recommended practice for architectural description of software-intensive systems, 2007.

[15] M. M. Lankhorst. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, Berlin, Heidelberg, Germany, 2^{nd} edition, 2009.

[16] B. LaShell. Agile development for sap: Get into the scrum! http://www.sappro.com/article.cfm?id=3339 (cited 2011-23-03), 2007.

[17] J. K. Liker. *The Toyota way: 14 management principles from the world's greatest manufacturer*. McGraw-Hill Professional, 2004.

[18] C. Lucke, S. Krell, and U. Lechner. Critical issues in enterprise architecting - a literature review. In *Proceedings of the Sixteenth Americas Conference on Information Systems (AMCIS 2010)*, Lima, Peru, 2010.

[19] T. Ohno. *Toyota Production System: Beyond Large-Scale Production*. Productivity Press, Portland, OR, USA, 1st edition, 1988.

[20] M. Op 't Land, E. Proper, M. Waage, J. Cloo, and C. Steghuis. *Enterprise Architecture – Creating Value by Informed Governance*. Springer, Berlin, Heidelberg, 2009.

[21] M. Pulkkinen and A. P. Hirvonen. Ea planning, development and management process for agile enterprise development. In *38th Hawaii International Conference on System Sciences*, Big Island, HI, USA, 2005. IEEE Computer Society.

[22] B. Rhubart. Agile Enterprise Architecture. http://www.oracle.com/technetwork/issue-archive/2010/10-nov/o60architect-175580.html (accessed at 2011-24-03), 2010.

[23] J. W. Ross. Creating a strategic it architecture competency: Learning in stages. *MIS Quarterly Executive*, 2(1), 2003.

[24] K. Schwaber. *Agile Project Management with Scrum*. Microsoft Press, Redmond, WA, USA, 1st edition, 2004.

[25] K. Schwaber and M. Beedle. *Agile Software Development with Scrum*. Prentice Hall, New Jersey, NY, US, 1st edition, 2001.

[26] W. A. Shewhart. *Statistical Method from the Viewpoint of Quality Control*. Dover Publication, New York, NY, USA, 1986.

[27] D. K. Taft. Microsoft Lauds Scrum Method for Software Projects. http://www.eweek.com/c/a/IT-Management/Microsoft-Lauds-Scrum-Method-for-Software-Projects/ (cited 2011-23-03), 2005.

[28] The Open Group. TOGAF "Enterprise Edition" Version 9. http://www.togaf.org (cited 2011-06-08), 2009.

[29] B. van der Raadt, S. Schouten, and H. van Vliet. Stakeholder perception of enterprise architecture. In R. Morrison, D. Balasubramaniam, and K. E. Falkner, editors, *ECSA*, volume 5292 of *Lecture Notes in Computer Science*, pages 19–34, Berlin, Heidelberg, Germany, 2008. Springer.

[30] B. d. van Raadt and H. van Vliet. Designing the enterprise architecture function. In F. P. Steffen Becker and R. Reussner, editors, *4th International Conference on the Quality of Software Architectures (QoSA2008)*, volume 5281 of *Lecture Notes in Computer Science*, pages 103–118, Karlsruhe, Germany, 2008. Springer.

[31] R. Wagter, M. van den Berg, J. Luijpers, and M. van Steenbergen. *Dynamic Enterprise Architecture: How to Make IT Work*. John Wiley, 2005.

[32] J. Webster and R. T. Watson. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2):xiii–xxiii, 2002.