Enterprise Architecture and Knowledge Perspectives on Continuous Requirements Engineering

Marite Kirikova
Institute of Applied Computer Systems, Riga Technical University, 1 Kalku, Riga, LV-1658, Latvia
marite.kirikova@rtu.lv

Abstract. When looking at the term "continuous requirements engineering", there is a limited number of sources using this term in different collections of scientific papers. However, while the term is not so often used, still the ideas with respect to different issues of continuous requirements engineering are discussed in papers on enterprise architecture, business process management, and adaptive information systems. The paper lists different challenges in continuous requirements engineering (1) by looking from the perspective of enterprise architecture and knowledge involved in requirements engineering and (2) by considering ideal linkages between knowledge, enterprise architecture, business processes, and development projects. It also refers to the current work under the query "continuous requirements engineering" and briefly discusses how it helps to meet various challenges of continuous requirements engineering.

Keywords: continuous engineering, requirements engineering, enterprise architecture, knowledge perspective

1 Introduction

Flexibility and ability to change in a fast manner has become a necessity in the nowadays networked business environment. Taking into consideration that the changes are needed not only in the products or services that enterprises provide to bring value for their environment, but also in organizational processes and structures, it is quite straight forward that requirements engineering has to become a continuous activity instead of being just a part of temporary information systems projects.

When looking at the term "continuous requirements engineering", there is a limited number of sources using this term in different collections of scientific papers. However, while the term is not so often used, still the ideas with respect to different issues of continuous requirements engineering are discussed in papers on enterprise architecture, business process management and adaptive information systems [1]. Most of research work has concentrated on product requirements, i.e., on how to identify what changes are needed in the information systems development product (or manufacturing products). Less attention has been paid to the relationship between the needs for changes in information systems and needs of changes in enterprise architectures (EA). One may argue that EA development frameworks include changes in requirements and information systems [2]. While theoretically it is so, still in practice the requirements are rarely handled systemically before "testing" their quality at the implementation stage.

The purpose of this paper is to discuss continuous information systems requirements engineering from the EA and knowledge perspectives. The research method applied here is simple. The information system is considered as the supporter of business processes that produce products or services; and business processes, in turn, are considered as a part of EA. The situation between snapshots of two states of EA (As-Is and To-Be) is taken as an object of analysis with the purpose to envision "ideal" linkage between different EA models and knowledge of various stakeholders for continuous requirements engineering.

The paper is structured as follows. Section 2 introduces the ideal linkage between models and knowledge of stakeholders, based on EA concept. Section 3 discusses how contemporary approaches to requirements engineering support this linkage. Section 4 concludes the paper with the list of topics that would be relevant for further research in continuous requirements engineering.

2 EA, Knowledge, and Continuous Requirements Engineering

The purpose of this section is to provide a context for the discussion on continuous requirements engineering. We mainly will speak about information systems requirements (or requirements for information technology solutions). However, the issues discussed here are relevant also for other domains of requirements engineering.

At a high level of abstraction and following the top-down systems development thinking, we can assume that any changes in enterprise architecture (EA) might be propagated to business processes and, when architecture gap is identified, implemented in information systems development projects (see Fig. 1).

We can also assume that all of the phenomena reflected in Fig. 1 (As-Is EA and Business Process, To-Be EA and Business Processes, the EA Gap and the project) are at least to some extent documented. What is essential to pay attention to here is that in all cases part of the knowledge are still tacit - reflected only in the brains of people involved into EA, business process, or systems development. Moreover, explicit knowledge can be reflected in different ways - it might be reflected in either documents or models that allow consideration of an artifact as a whole only; or it can be reflected in the models that allow to consider the artifact as a whole and also its elements at several levels of detail. For instance, there might be a possibility to view just a document with the picture of a business process in a particular notation; or there can be a possibility to view the business process model and add the link to one of its activities or resources. Thus with respect to each of above-mentioned phenomena we can say that there is tacit, and explicit knowledge that refers to it. And, with respect to the explicit knowledge, we can distinguish between the knowledge that is reflected in some knowledge holders in general and the knowledge that is reflected in systems engineering tools, in particular. *In the ideal case*, with respect to each artifact, tacit knowledge includes explicit knowledge and, in turn, explicit knowledge includes knowledge reflected in systems engineering tools (see triangles in Fig.1). Taking into consideration the complexity and the size of EA artifacts, it is not realistic that all of them would be reflected in tacit knowledge of stakeholders and developers, and not all items in the tools are related explicitly to all other relevant artifacts outside the tool. Thus the *challenge* here is *how to establish right equilibrium between different types of knowledge to be able to ensure relative consistency between different artifacts; and how to maintain the consistency when one knowledge type changes*.

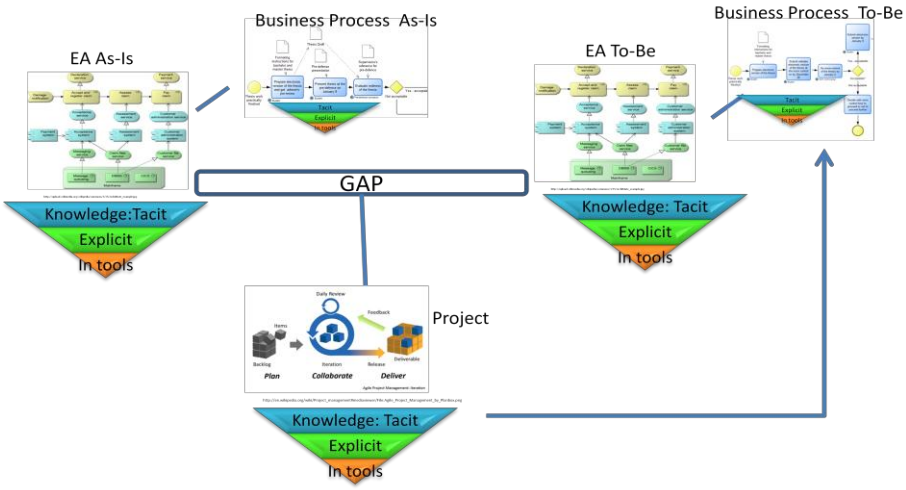

Fig. 1. A simplified view on change propagation

So far we looked from the top-down point of view. However, in systems development, bottom-up, meet in the middle, iterative, incremental, and many other approaches are considered. It shows that the changes in requirements can concern EA, business process, and project at any time form different perspectives (see Fig. 2).

Such dense possibilities for changes and agility requested in meeting the changes, *in the ideal case,* would require documentation of all potential and real changes in the tool used for systems engineering in order to follow the basic principles of engineering, such as "manage phased life-cycle plan", "perform continuous validation", "maintain disciplined product control", etc [3]. While the impact between the software or hardware development (acquisition) project, business processes, and EA is clear in general, the ability to manage specific relationships in each moment of time still face the following *challenges to know*:

- *What is the landscape (or the ecosystem of models and physical objects and agents) that has to be taken into consideration?*
- *What is the level of granularity at which the change has to be considered?*
- *Is it possible to distinguish between permanent changes and temporal changes?*
- *Is it possible to distinguish between high impact changes and low impact changes?*

When speaking about relationships between EA and business processes, *in the ideal case*, the EA would reflect data elements that refer to all data states in the business process models as well as all performers of processes with respect to all possible process instances. However, such business process notations as BPMN [4], scarcely reflect data processing issues, and also the possibility to reflect performers by BPMN is pretty limited. Thus there is the *next challenge, namely incompatibility between EA requirements and business process requirements representation possibilities*.

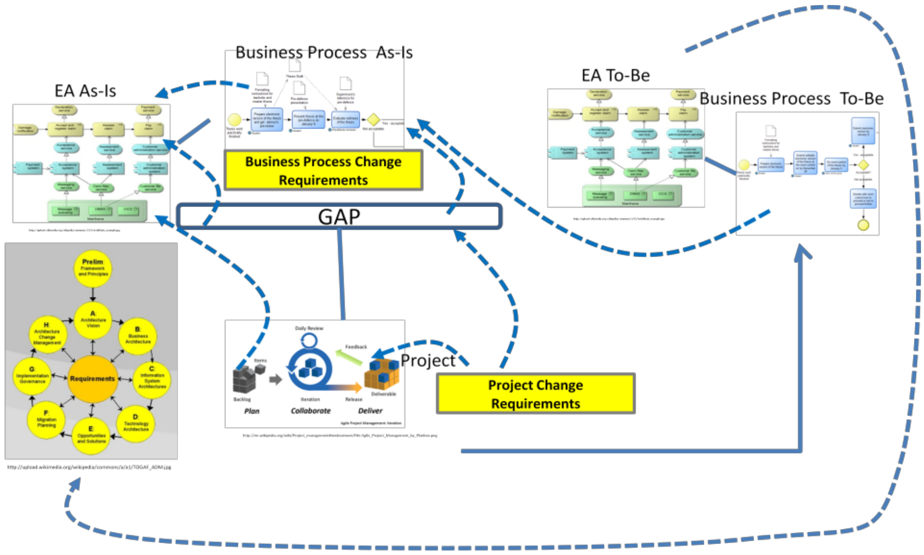

Fig. 2. Closer to reality in change propagation

With respect to the software requirements and business process requirements the situation is similar to the one discussed in the previous paragraph. The information in project requirements is considered mainly as something to be acquired from databases or data warehouses and the information circulation in the business processes rarely is taken into consideration. But *in the ideal case* it would be essential to see both - the way how data (information) historically is modified in each step of the process and where and how it rests during the lifecycle of objects that data reflects. Thus the one more *challenge* in continuous requirements engineering is *limited possibilities to see the relationship between information circulation in business processes and information life cycle in information storage systems*.

At the end of this section it is necessary to point to one more issue that becomes a *challenge* in nowadays requirements engineering, namely, *the form and the role of requirements specification is not clear*. Development and reading of requirements specification is a very time-consuming task. So in many agile projects its use is as much as possible limited. On the other hand, the product of requirements engineering is requirements. So the product of continuous requirements would be continuously changing requirements which still adhere to all quality characteristics of the requirements specifications. Thus the question on *what is the ideal form of the product of continuous requirements engineering* is still open.

3 Related Work on Query "Continuous Requirements Engineering"

In this section, in the context of challenges in continuous requirements engineering discussed in Section 2, we will look at some approaches which were available on the query "continuous requirements engineering" in Springer, IEEE, and Elsevier collections of scientific papers. The total number of publications found on this query was eleven, two of them were not relevant in the context of this paper. Table 1 represents, which challenges, at least partly, are addressed in which of nine papers discussed below.

As mentioned in the introduction, a lots of work relevant for continuous requirements engineering is done in the area of self-adaptive systems. In [1] a modeling language, called Adaptive RML, for the representation of early requirements for self-adaptive systems is proposed The language has graphical primitives and formalization. This work can be considered as the step forward to meet the challenge of unclear role and form of the product of continuous requirements engineering.

In [5] different types of changes in requirements are discussed and continuous adaptive requirements engineering framework CARE is proposed, which leverages so called Techne - a new generation requirements modeling language with goals, preferences, and inconsistency handling. The work in [1] also is closely related to Techne. The following types of changes are considered in [5]:

- Changes that are anticipated at the design time
- Changes in the environment (with the request that the system must monitor each change)
- Changes that are unanticipated by the designer for which there is a possibility for addition or simple refinement in the system
- Changes that are unanticipated by the designer for which there is no possible addition or simple refinement in the system (system evolution instead of adaptation is needed)

In [6], in the context of adaptive systems, the main attention is given to requirements handling at run time. In [7] a special attention is paid to end-user requirements in the context of self-adaptive systems. The work on adaptive systems well shows the dynamics in continuous requirements engineering. However the requirements engineering there is focused on one specific adaptive system only. When looking from EA perspective, many different systems and services shall be considered, so the landscape or ecosystems challenges still are not met in contemporary work on continuous requirements engineering in the context of adaptive systems. Nevertheless the work represented in [5]-[7] to some extent addresses Challenge 1 as well as Challenges 3-6 represented in Table 1.

Table 1. Challenges partly addressed by current work in Continuous Requirements Engineering

N	*Challenge*	*Sources*
1	How to establish right equilibrium between different types of knowledge to be able to ensure relative consistency between different artifacts, and how to maintain the consistency when one knowledge type changes	[5], [7], [8], [9], [14]
2	It is not clear how to know what is the landscape (or the ecosystem of models and physical objects and agents) that has to be taken into consideration for currently handling the change	[13]
3	It is not clear how to know what is the level of granularity at which the change has to be considered	[5]
4	It is not clear how to distinguish between permanent changes and temporal changes	[6]
5	It is not clear how to distinguish between high impact changes and low impact changes	[5]
6	Incompatibility between EA requirements and business process requirements representation possibilities	[5]
7	Limited possibilities to see the relationship between information circulation in business processes and information life cycle in information storage systems	
8	The form and the role of requirements specification is not clear	[10], [11], [12]
9	It is not clear what is the ideal form of the product of continuous requirements engineering	[1], [10], [11], [12]

In [10], [11], and [12] four requirements abstraction levels are proposed and verified in different case studies. The following requirements abstraction levels are proposed:

- Product level requirements (goals)
- Feature level (features)
- Functional level (functions/actions)
- Component level (details, consists of)

The paper [10] presents also distribution of requirements sources concerning each requirement type. Additionally, the paper presents a number of requirements attributes and the list of states of requirements and requirements state transition model. The results of this research including their validation ([11], [12]) might be helpful for meeting Challenge 8 and Challenge 9.

Requirements in the context of EA are discussed in [13]. In [14] monitoring of requirements is discussed in larger context than in [9]. Thus the work in [13] to some extent contributes to meeting Challenge 2 and work in [14] addresses Challenge 1.

Table 1 shows that the related work found on query "continuous requirements engineering" can provide some solutions for the most of challenges discussed in Section 2. However, none of them fully meets any of these challenges. Handling of tacit and explicit knowledge has been addressed by several approaches, including usage of business analytics and monitoring techniques, which is important contribution in engineering of requirements. However, only two approaches (the one in [1] and the one in [10]-[12]) addresses the product of continuous requirements engineering (Challenge 8 and Challenge 9). Scope of artifacts to be addressed (Challenge 2), granularity of knowledge (Challenge 3), persistence (Challenge 4), and impact of changes (Challenge 5) as well as incompatibility between EA requirements and business process requirements representation possibilities (Challenge 6) are also scarcely addressed. Challenge 7 - limited possibilities to see the relationship between information circulation in business processes and information life cycle in information storage systems, is not addressed at all. Thus additional research is needed to address all challenges on continuous requirements engineering.

The spectrum of related work discussed in this section is limited to the research papers available on the query "continuous requirements engineering". The papers that are not positioned under keyword "continuous requirements engineering", but still are related to any of challenges presented in Table 1, have to be investigated in further research as there could be found solutions that can contribute in meeting the challenges of continuous requirements engineering.

Conclusions

The paper presents different challenges in continuous requirements engineering from enterprise architecture and knowledge perspectives based on requirements engineering issues in enterprise architecture, business processes, and systems development projects. It also refers to related work under the query "continuous requirements engineering" and briefly discusses how this work can contribute for meeting identified challenges.

The paper reveals nine challenges in continuous requirements engineering. It shows that the related work quite largely addresses one challenge, seven challenges are addressed by at least one work, but one challenge, namely, limited possibilities to see the relationship between information circulation in business processes and information life cycle in information storage systems, is not addressed at all.

Further research will include search and analysis of related works in broader scope of areas for each presented challenge of continuous requirements engineering, as well as development of continuous requirements engineering methods that integrate and extend existing solutions and adhere to the main principles of engineering.

Acknowledgment

This work is supported in part by the Latvian National research program SOPHIS under grant agreement Nr.10-4/VPP-4/11.

References

1. Qureshi, N. A., Jureta, I. J., Perini, A.: Towards a Requirements Modeling Language for Self-Adaptive Systems. Requirements Engineering: Foundation for Software Quality. LNCS, vol. 7195. pp. 263–279 (2012)
2. TOGAF® 9.1: Part II: Architecture Development Method (ADM) . Introduction to the ADM, available at http://pubs.opengroup.org/architecture/togaf9-doc/arch/chap05.html, (1999-2011)
3. Boehm, B. W. Basic principles of software engineering. Journal of Systems and Software 3, pp. 3-24 (1983)
4. Business Process Model and Notation (2011), available at http://www.omg.org/spec/BPMN/2.0/
5. Qureshi, N.A., Perini, A., Ernst, N.A., Mylopoulos, J.: Towards a continuous requirements engineering framework for self-adaptive systems. 2010 First International Workshop on Requirements@Run.Time. pp. 9–16. IEEE (2010)
6. Cheng, B. H. C., Eder, K. I., Gogolla, M., Grunske, L., Litoiu, M., Müller, H. A., Pelliccione, P., Perini, A., Qureshi, N. A., Rumpe, B., Schneider, D., Trollmann, F., Villegas, N.M.: Using Models at Runtime to Address Assurance for Self-Adaptive Systems. Models@run.time. LNCS, vol. 8378. pp. 101–136 (2014)
7. Qureshi, N. A., Seyff, N., Perini, A.: Satisfying User Needs at the Right Time and in the Right Place: A Research Preview. Requirements Engineering: Foundation for Software Quality. LNCS, vol. 6606. pp. 94–99 (2011)
8. Natt och Dag, J., Regnell, B., Carlshamre, P., Andersson, M., Karlsson, J.: A Feasibility Study of Automated Natural Language Requirements Analysis in Market-Driven Development. Requir. Eng. 7, pp. 20–33 (2002)
9. Knauss, E., Damian, D., Cleland-Huang, J., Helms, R.: Patterns of continuous requirements clarification. Requir. Eng. (2014)
10. Gorschek, T., Wohlin, C.: Requirements Abstraction Model. Requir. Eng. 11, pp. 79–101 (2006)
11. Gorsichek at al. A controlled evaluation of requirements abstraction model. Information and Software Technology 49, pp. 790-805, Elsevier, (2007)
12. Gorschek, T., Garre, P., Larsson, S.B.M., Wohlin, C.: Industry evaluation of the Requirements Abstraction Model. Requir. Eng. 12, pp. 163–190 (2007)
13. Pimentel, J., Castro, J., Santos, E., Finkelstein, A.: Towards Requirements and Architecture Co-evolution. Advanced Information Systems Engineering Workshops. LNCS, vol. 112. pp. 159–170 (2012)
14. Robinson, W.N.: Implementing Rule-Based Monitors within a Framework for Continuous Requirements Monitoring. Proceedings of the 38th Annual Hawaii International Conference on System Sciences. p. 188a–188a. IEEE (2005)