Continuous Requirements Engineering in Sociotechnical Systems: Challenges and Solutions

Marite Kirikova
Department of Artifical Intelligence and Systems Engineering, Riga Technical University
Riga, Latvia
marite.kirikova@rtu.lv

***Abstract*—Continuous requirements engineering in sociotechnical systems faces the challenges that originate from diverse and fast changes in systems contexts, project-based issues, and the multi-systems nature of sociotechnical systems. The interplay of these challenges and reported suggested treatments point to the necessity for flexible frameworks and new ways of knowledge management in systems development projects that concern sociotechnical systems.**

Keywords—sociotechnical systems, requirements engineering, continuous engineering, knowledge management

I. Introduction

Requirements engineering, as a discipline, has been known for already more than three decades. It differs from mere requirements management or requirements elicitation by using methods and artefacts that provide a system-based view of the requirements and their contexts. With the advent of agile and continuous delivery in sociotechnical systems development, agility and continuity are also expected in the field of requirements engineering [1], [2]. However, many problems in this area have been reported and discussed when handling requirements in agile and DevOps environments [3], [4]. And these are not the only challenges related to requirements being faced by today's sociotechnical systems development projects. The broader scope and sources of these challenges are discussed in this paper with the purpose of characterizing the variations of continuity that can be expected in requirements engineering; and pointing to some possible solutions that are emerging when looking at treatments applied in order to meet the reported challenges.

Section II ponders over the necessity of continuity in requirements engineering in sociotechnical systems. Section III discusses challenges and treatments in agile projects. Section IV concerns the challenges that stem from the multi-systems nature of sociotechnical systems. Section V suggests some solutions for meeting the challenges discussed in sections III and IV. Section VI concludes the paper with the emphasis on the need for new forms of knowledge management in continuous requirements engineering in sociotechnical systems.

II. Why Continuity in Requirements Engineering

A. *Specifics of Sociotechnical Systems*

"A sociotechnical system is one that considers requirements spanning hardware, software, personal, and community aspects". [5] Therefore in sociotechnical contexts it is necessary to be concerned about the different subjects and objects of requirements at different levels of abstraction and decomposition. The diversity of objects and subjects is accompanied by their different speeds of action, life cycles, and mutual relationships. All of the aforementioned aspects are sources of possible changes in requirements that can happen in both predictable and unpredictable situations and time points. To embrace this diverse and fast changing environment of requirements, the continuous handling of such requirements, based on a good understanding of systems involved, can be derived as a logical means for requirements engineering. Some of the reasons for continuity in requirements engineering of sociotechnical systems are shown inf Fig. 1.

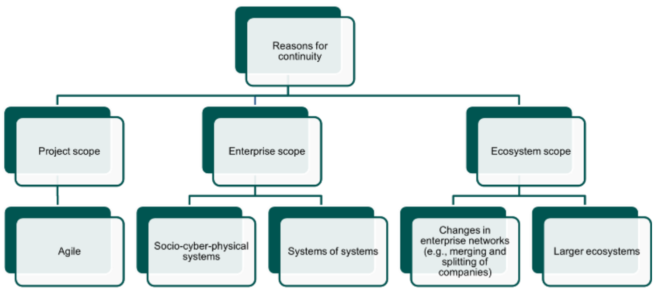

Fig.1. Reasons for continuity in requirements engineering.

The reasons are structured in three groups. The first group of reasons derives from the nature of projects that are performed for developing the constituents of sociotechnical systems [3]. Agile approaches work with the artifacts (e.g., user stories) that differ from the ones used in traditional requirements engineering. As a result, several challenges are reported by researchers and practitioners [3]. [4].

Another group of challenges stems from the variable nature of systems to be considered in sociotechnical contexts. It is not only that social and technical aspects are to be looked at in a systems-based way; today the elements of artificial intelligence are often embedded in physical systems and, therefore, their features also must be respected by developers.

One more aspect generating challenges is the different types of relationships between the systems. This is a well-recognized problem which needs to be solved. In 2019 it was deemed necessary for a standard [6], facilitating the handling of these relationships, to be released. The same aspect also concerns the roles of sociotechnical systems in their environment, where frequent changes in company relationships occur; due to their merging with other companies or acquiring new ones and then integrating them in their structures. These changes influence the ecosystemic balance in the environment that must be considered in requirements engineering so as to avoid breaking or disturbing value networks that are essential for well-being of sociotechnical systems, their development, and adjustment.

The discussed features of projects and sociotechnical systems impact knowledge content and processes in continuous requirements engineering.

B. *Knowledge Content and Processes in Continuous Requirements Engineering*

The challenges and their treatments in continuous requirements engineering can be considered from two perspectives, namely, the perspective of knowledge content and the perspective of requirements engineering processes or activities. Regarding the knowledge content, two types of knowledge are essential in requirements engineering – tacit and explicit. The main factor in traditional requirements engineering is explicit knowledge expressed in forms of models and requirements specifications [7]. In agile approaches, tacit knowledge plays an essential role, as knowledge representation and acquisitions formats are less complex and less consistent; often leaving knowledge integration results undocumented [8]. Both perspectives will be considered in the two following sections that focus on some of the continuous requirements engineering challenges and treatments.

III. Challenges and Treatments in Agile Projects

Requirements engineering challenges in agile projects have been under the watch of researchers for several years [3]. Recently, two surveys were published about this topic [3], [4]. These surveys are used in this section to discuss the challenges and treatments from the perspectives of knowledge content, its distribution [9], and requirements engineering activities. The discussion follows the structuring of challenges in (partly overlapping) groups proposed in [3], also adding to each group some of the issues discussed in [4].

Group1: *Build and maintain shared understanding of customer value* [3], *direct communication with stakeholders* [4], *less preliminary planning and focus, no initial team involvement* [4], *tacit knowledge* [4]. As agile approaches rely upon tacit knowledge, it is challenging to understand customer value without direct communication. On the other hand, direct communication is time-consuming and the benefit from it is perceived mainly by those involved in the communication. So, the essential questions here are: when who should communicate with whom and how acquired knowledge can and should be further distributed.

Group 2: *Support change and evolution* [3], c*hanging requirements* [4]. While well-defined requirements management procedures are available in conventional requirements management tools, handling changes in vaguely defined requirements procedures is a new problem. Here, the main questions are how to identify the change, how to see its impact on other requirements, and how to know when, to whom, and how the changes should be communicated.

Group 3: *Build and maintain shared understanding about system* [3], *lack of documentation* [4]. To meet this challenge, systems thinking, and the appropriate amount of documentation are seen as possible treatments. When considering the number of views, possible levels of decomposition and abstraction, and information/knowledge dependencies and their flows between developers and stakeholders, continuous maintaining of a valid shared body of knowledge seems to be a task of a very high complexity.

Group 4: *Representation of requirements knowledge* [3]: *manage levels of decomposition, consistency, quality of requirements*, etc; *missing, ambiguous, and conflicting requirements* [4], *negligence of non-functional requirements, inability of customer in telling user stories* [4]. Whereas Group 3 concerned overall knowledge about the sociotechnical system, this group of challenges directly concerns the knowledge about requirements. More specifically, requirements knowledge may be missing, may be represented inappropriately or not at the right level of decomposition. It seems that the problems with requirements knowledge do not differ from those of Group 3, therefore the same tools might be used for handling both of these groups of challenges.

Group 5: *Process aspects* [3], such as *prioritization, managing completeness, consistency*, and *quality of requirements*; also, *requirements prioritization* in [4]. Some approaches, such as staged frameworks [4], clear hierarchy of teams, backlog combinations [3], and taxonomies are proposed for handling these problems, while acknowledging that there are no agreed upon means for managing complexity of requirements.

Group 6: *Organizational aspects* [3] *such as bridging plan driven and agile, planning validation and verification based on requirements, allocating time for invention and planning, and seeing the impact on infrastructure.* System-level awareness, iterations in planning and innovation, actively managed boundary objects are some of the suggested treatments [3].

Common treatments that are suggested in several groups of challenges are the following:

- Holistic view on sociotechnical systems and requirements
- Respecting and introducing hierarchies
- Establishing and maintaining the traceability between knowledge items
- Well-organized communication

These aspects suggest that a well-defined and, at the same time, flexible system or structure of knowledge is expected to lie behind the methods and tools of continuous requirements engineering in agile settings. Systems aspects, from a different perspective, are discussed further in the next section.

IV. Enterprises and Ecosystems: a Systems-based View

When looking from an enterprise and ecosystem perspective on continuous requirements engineering, two issues become the main sources of challenge: (1) the diversity of system types (social, software, hardware, physical, biological, etc.) and (2) diversity of relationships between the systems. The challenges stemming from the diversity of the systems will be discussed in the context of socio-cyber-physical systems [10]. The challenges regarding the diversity of relationships between systems will be discussed based on research in the Systems of Systems area [11], [12], [13].

A. *Diversity of Systems in Sociotechnical Contexts*

In requirements engineering, the diversity of systems requires consideration of a large amount of data, information, and knowledge flows, that are, for instance, produced by systems of different natures [10]:

- Mechanical hardware components
- Computing hardware components that can be stand-alone (computers) or those embedded in mechanical hardware (smart devices)

- Software components that are installed on computing hardware
- Human components that can form different social structure components

Data and knowledge can be part of computing software, human or social components. Data (as a components) are used for communication; and data, information, and knowledge are exchanged between all other components. Data can be processed, transformed into information, and saved as knowledge in different ways. Thus, the diversity of systems causes a hard to manage diversity of data, information, and knowledge that needs to be handled in continuous requirements engineering. Besides these problems (and partly because of them), cyber-physical contexts yield similar problems to those discussed with respect to agile environments (levels of decomposition of knowledge, different lengths of activities, lack of tools for knowledge handling, etc.) [14].

Additionally, such issues as openness of systems, the necessity to consider their real time behavior in several contexts, and their natural and artificial intelligence-based adaptability, contribute to the complexity of analyzing, representing, and planning for requirements engineering in sociotechnical systems.

B. *Diversity of Relationships between Systems.*

Sociotechnical systems form various types of systems of systems; for instance, directed, acknowledged, collaborative, and virtual systems [11]. The patterns of relationships are different in each of these cases and must be, first, discovered, second, represented and respected, and then thirdly, the changes in these relationships must be perceived and understood so as to align the requirements with those relationships between the systems that are actually in place at a given point of time. Understanding of relationships between systems is based on consideration of the following features attributed to systems of systems [11], [12], [13]:

- *Independence*, which shows that the systems which form a larger system can operate and are managed separately [12]. It is possible to distinguish between operational independence, managerial independence [13], and evolutionary independence [11].
- *Distribution*. The systems that form larger systems can be dispersed and, also, communicate over larger distances [12]. For physical systems, such as smart cars or traffic lights, geographical distribution is essential [13]. However, when considering social systems and software, the topological distribution can be considered with respect to social distances, code threading, and other aspects.
- *Emergence* which is defined as the behavior of a larger system that exceeds the behavior of the systems that are parts of it (its constituent systems [6]) [12]. It is possible to distinguish between three types of emergence behaviors [11], [15]: simple emergence behavior that occurs in relatively simple systems and can be predicted; weak emergence behavior which is the expected emergence behavior that is desired or allowed for in the system structure, but cannot be predicted from the knowledge of the characteristics of the individual constituent systems; and strong emergence behavior which is unexpected emergence behavior that becomes evident only during system failure and cannot be attributed to any particular constituent system(s). This feature is the least researched in requirements engineering and one of the most challenging issues in continuous requirements engineering.
- *Evolution*, as systems of systems are in continual development and can never be considered fully completed.

One of the forms of evolution is merging of companies or acquisition of one company by another company. These cases are especially challenging because the identities and roles of constituent systems change, knowledge loss is possible, and many iterations are needed to reach consensus with respect to the changes in social and information technology related aspects. In these systems it is important to acquire, accumulate, share, and process knowledge, not only about current and future states of the system(s), but also to acquire, accumulate, share, and process knowledge about the implementation of changes [16]. In the next section a possible integration of these knowledge management activities with a continuous requirements engineering framework will be shown.

V. Knowledge Management Enhanced Continuous Requirements Engineering

The challenges of continuous requirements engineering in sociotechnical systems show that the tasks of requirements engineering, in this context, require strong support in terms of knowledge management both (1) in terms of the content of knowledge to by acquired, accumulated, processed, and shared and (2) in terms of activities and processes performed during continuous requirements engineering. For instance, agile and plan driven approaches are expected to be combined [3]. One of the frameworks that allows accommodation of different life cycles of systems development is the FREEDOM framework [17]. This assumes fractal organization of knowledge regarding target systems, its context, and the systems development processes. This framework can also accommodate different enterprise architecture representations regarding current and future states of sociotechnical systems [18]; and can be supported by knowledge management methods and tools (Fig. 2).

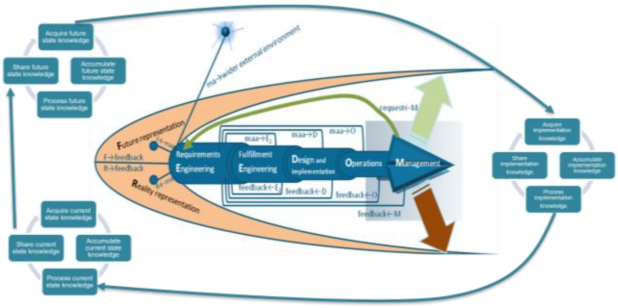

Fig.2. FREEDOM framework and knowledge management.

Fractality in representing knowledge about the sociotechnical system(s) and development processes might be a solution for some of the challenges discussed in the previous sections. This could allow traceability between hierarchically well-organized pieces of knowledge that are structured according to organizational units, processes, and development projects [19]. It could define separate knowledge distribution processes for development teams and help with handling consistency between requirements at different levels of granularity, and of different forms of representation. Thus, it is a potential means for achieving the following suggested treatments of the requirements engineering challenges in sociotechnical systems that were discussed in Section III:

- Holistic view on sociotechnical systems and requirements can be achieved as fractal knowledge representation provides the possibility of considering knowledge items simultaneously via "part of" and classification relationships between knowledge components.
- Respecting and introducing hierarchies is possible as (1) hierarchies are a natural form of representation in a fractal system, and (2) a multifractal approach (having different hierarchies for parameters that scale the system) can be applied so as to have both the knowledge component hierarchies according to the sociotechnical systems configurations and the knowledge component hierarchies according to the development team configurations.
- Establishing and maintaining the traceability between knowledge items is possible as the fractal system allows for preserving different types of relationships between items belonging to different fractals.
- Well-organized communication could be achieved by combining fractal knowledge representation with the compliant knowledge distribution methods [9].

The above proposal highlights just some of the possibilities that will further be elaborated and tested in continuous engineering settings where sociotechnical systems are concerned. For instance, it is not clear whether fractal representations of knowledge can help in the handling of the strong emergence behavior challenge, which is the least researched challenge in continuous requirements engineering in sociotechnical systems.

VI. Conclusion

Analysis of challenges in different requirements engineering projects has revealed the necessity for appropriate knowledge management methods and tools that could ensure knowledge reuse, timely and adequate knowledge acquisition and distribution; and establishing and representing well recognizable relationships between knowledge components used in systems development. This necessity emerges behind almost all of the challenges discussed in this paper. Whilst, the origins of the challenges are different, they are all rooted in the lack of (1) tool-supported systems-based representation of knowledge, as the basis for continuous requirements engineering, and (2) a simple means to handle this knowledge and to use it. Fractal approaches may be a solution for knowledge representation as they allow development of complex representations as simple combinations of pre-defined and emerging components.

References

[1] E.-M. Schön, J. Thomaschewski, and M. J. Escalona, "Agile requirements engineering: a systematic literature review," *Computer Standards & Interfaces*, vol. 49, pp. 79–91, 2017.

[2] S. Mosser and J.-M. Bruel, "Requirements engineering in the DevOps era," *2021 IEEE 29th International Requirements Engineering Conference (RE)*, pp. 510–511, 2021.

[3] R. Kasauli, E. Knauss, J. Horkoff, G. Liebel, F. Gomes de Oliveira Neto, "Requirements engineering challenges and practices in large-scale agile system development," *The Journal of Systems & Software*, vol. 172, article 110851, 2021.

[4] A. Rasheed, B. Zafar, T. Shehryar, N. A. Aslam, M. Sajid, N. Ali, S. H. Dar, and S. Khalid, "Requirement engineering challenges in agile software development," *Mathematical Problems in Engineering*, vol. 2021, article 6696695, 18 pages, 2021.

[5] Socio-technical systems, Interactive design. Available: https://www.interaction-design.org/literature/topics/socio-technical-systems

[6] ISO/IEC/IEEE 21839:2019, Systems and software engineering – System of systems (SoS) considerations in life cycle stages of a system. Available: https://www.iso.org/standard/71955.html

[7] J. A. Bubenko and M. Kirikova, "Improving the quality of requirements specifications by enterprise modelling," *Perspectives on Business Modelling*, Springer, Berlin, Heidelberg, pp. 243–268, 1999.

[8] V. Gervasi, R. Gacitua, M. Rouncefield, P. Sawyer, L. Kof, L. Ma, P. Piwek, A. de Roeck, A. Willis, H. Yang, and B. Nuseibeh, "Unpacking tacit knowledge for requirements engineering," *Managing Requirements Knowledge*, Springer, Berlin, Heidelberg, pp. 23–47, 2013.

[9] M. Kirikova and J Grundspenkis, "Using knowledge distribution in requirements engineering," *Knowledge-Based Systems*, pp. 149–184, 2000.

[10] K. Lace and M. Kirikova, "Required changes in requirements engineering approaches for socio-cyber-physical systems," *24th Joint International Conference on Requirements Engineering: Foundation for Software Quality Workshops, Doctoral Symposium, REFSQ-JP 2018*; Utrecht; Netherlands; 19 March 2018, CEUR Workshop Proceedings, vol. 2075, 2018.

[11] C. Ncube and S. L. Lim, "On systems of systems engineering: a requirements engineering perspective and research agenda," *2018 IEEE 26th International Requirements Engineering Conference (RE)*, pp. 112–123, 2018.

[12] S. Hallerstede, F. O. Hansen, J. Holt, R. Lauritsen, L. Lorenzen, and J. Peleska, "Technical challenges of SoS requirements engineering," *2012 7th International Conference on System of Systems Engineering (SoSE)*, pp. 573–578, 2012.

[13] F. L. Duarte, A. Félix de Castro, and P. G. G. Queiroz, "Reap-SoS: a requirement engineering approach for system of systems," *Computer Science and Information Technology*, April 2018.

[14] T. Rzazade, Continuous requirements engineering for cyberphysical systems, Master Thesis, Riga Technical University, Riga, Latvia, 2021.

[15] S. E. Page. *Understanding Complexity*. The Great Courses. Chantilly, VA, USA: The Teaching Company, 2009.

[16] K. Lace and M. Kirikova, "Post-merger integration specific requirements engineering model," *Artificial Intelligence in Business Informatics*, LNBIP, vol. 430, pp. 115–129, 2021.

[17] M. Kirikova, "Continuous requirements engineering in FREEDOM framework: a position paper," *Joint Proceedings of REFSQ-2016 Workshops, Doctoral Symposium, Research Method Track, and Poster Track co-located with the 22nd International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2016)*, March 14–17, 2016, Gothenburg, Sweden. CEUR Workshop Proceedings, vol. 1564, 2016.

[18] M. Kirikova, "Variable Contents of Enterprise Models," *Procedia Computer Science*, vol. 104, pp. 89–96, 2017.

[19] M. Kirikova, "Towards flexible information architecture for fractal information systems," *2009 International Conference on Information, Process, and Knowledge Management*, pp. 135–140, 2009.