A Modeling Language for Agile Requirements Engineering

Eva-Maria Schön *eva.schoen@iwt2.org*
University of Seville
Seville, Spain

Jörg Thomaschewski *joerg.thomaschewski@hs-emden-leer.de*
University of Applied Sciences Emden/Leer
Emden, Germany

María José Escalona *mjescalona@us.es*
University of Seville
Seville, Spain

Nora Koch *nora.koch@iwt2.org*
University of Seville
Seville, Spain

Abstract

Agile methodologies have an impact on how organizations carry out Requirements Engineering (RE). In this context, organizations use different kind of agile techniques like artifacts, meetings, methods or roles, but there is a lack of specific guidelines for agile RE. The aim of this paper is to present a modeling language for supporting organizational aspects of agile RE. It allows the visualization of agile RE concepts and their relationships, which can be used to define guidelines for a specific organization, project or domain. The modeling language for agile RE is used in projects in industry and our experiences reveal that it supports organizations in detecting problems and visualizing internal conflicts during the agile requirements phase, among other benefits.

Keywords: Agile Software Development, Requirements Engineering, Human-Centered Design, Metamodel, Profile

1. Introduction

Industry is incorporating Agile Software Development (ASD) in order to enhance the ability to manage changing priorities as well as reducing time to market [1]. In this context, organizations often use hybrid development models in order to adapt existing agile methodologies like Scrum [2], Kanban [3] or Extreme Programming [4] to their needs. Hybrid models consisting of an integration of agile methodologies and Human-Centered Design (HCD) [5] are used to increase the value delivery of an organization [6], [7]. Due to the integration of different agile methodologies new ways for Requirements Engineering (RE) have to be found. These hybrid methodologies are in alignment with existing knowledge in the field.

Our study on key challenges in agile RE [8] shows that companies are facing up different kind of problems in terms of agile RE. We have identified six key problems, these are: *functional or technical dependencies to other teams, losing sight of the big picture, understanding of agile values of the stakeholder, refine requirements in collaboration with users, involve stakeholders iteratively,* and *continuous management of requirements*. The study´s panel, who help to identify the agile RE problems, was composed of people from 19 different organizations. The organizations are heterogeneous in terms of size (freelancer up to concern), industry (e.g. e-commerce, consulting, cyber security, finance, and publishing) and business model (service provider and product manufacturer). This heterogeneity has led to

different perspectives in identifying the problems and has the advantage that the results of the study can be applied to many organizations. The agile RE problems can be handled by means of using best practices known from ASD. The detection of agile RE problems is a crucial task in terms of improving existing agile RE approaches. In light of this, we created the agile RE metamodel [9], which allows us to analyze an organizational environment as well as ease the detection of agile RE problems and conflicts.

The results of our systematic literature review (SLR) confirmed that guidelines for choosing appropriate agile techniques to carry out agile RE are missing [10]. This has also an impact on finding appropriate solutions for solving agile RE problems. To this end, we identified agile RE patterns [9] that provide practitioners with assistance for choosing appropriate agile techniques, such as user stories, impact maps or sprint reviews so as to solve their problems. We identified in sum 41 agile RE patterns like *Minimum Viable Product (MVP), definition of ready and definition of done, and refinement meeting*. For the complete list see [9].

In this paper, we are contributing a modeling language for supporting organizational aspects of agile RE. This modeling language provides visualization techniques for information systems engineering. Then, we show how the different components (agile RE metamodel, agile RE problems and agile RE patterns) work together. The modeling language can be used as an additional tool to support existing practices such as kaizen, retrospectives, or Scrum Master. The modeling language is defined as an UML profile of our RE metamodel [9] and lets practitioners and researchers build a domain specific model for agile RE by means of analyzing their organizational environment. The visual representation of the domain specific model supports the detection of agile RE problems and simplifies the selection of agile techniques.

This paper is structured as follows: Section 2 summarizes the state of the art of agile RE. Then, section 3 outlines the solution by describing the modeling language of agile RE. Section 4 shows how the modeling language is applied to industry. Subsequently, section 5 discusses on results and limitations and section 6 finalizes this work by means of conclusions together with an outlook on future work.

2. State of the Art of Agile Requirements Engineering

There is a shift of cultural values in agile environments, compared to environments that apply plan-based process models like waterfall models [11]. Agile values stated by the Manifesto for ASD [12] embrace change in the mindset and behavior of people. The responsibility for requirements management is no longer in one role; instead, the entire product development team is responsible for the continuous management of requirements. Moreover, in agile environments cross-functional collaboration and communication are more valued than comprehensive documentation of requirements.

In the literature, we can find some reviews and mapping studies, dealing with agile RE. For instance, Inayat et al. [13] investigated agile RE challenges and practices. They aimed to understand how traditional RE problems are resolved using agile RE. In summary, they provided 17 commonly used practices and also practical challenges that agile teams had to face. Soares et al. [14] analyzed difficulties while working with requirements in an agile environment, particularly, causes that can lead to documentation debt (e.g. missing, inadequate and incomplete requirements).

Beside these literature reviews, we conducted a systematic literature review [10] in order to analyze the state of the art of agile RE with strong focus on stakeholder and user involvement. In particular, we investigated what approaches exist to involve stakeholder in the process, which methodologies are commonly used to present the user perspective and how requirements management is carried out. In sum, 27 papers were included in the study [10] and were analyzed according to our predefined research protocol.

The results of that analysis revealed that the research field of agile RE is very close to current work practices in companies, since most of the included studies report results from case studies (19/27 = 70%). Then, we observed that agile RE is a complex research field with a lot of different cross-functional influences, for instance, from the fields of HCD, ASD, or RE.

However, we identified some gaps in existing literature. We learned that building a shared understanding concerning the user perspective is not very well established in agile environments, although building a shared understanding is very important in terms of requirements management in an agile environment [15], [16]. Moreover, we were not able to find a common process model for stakeholder and user involvement. Nevertheless, it is known that organizations usually deal with those shortcomings by integrating additional methodologies like HCD [17], Design Thinking [18], Contextual Inquiry [19] or Participatory Design [20], [21].

Furthermore, we found studies presenting process models for agile RE ([22], [20], [17], [23], [24]). All these process models have in common that they utilize different types of artifacts, meetings, methods and roles (referred to as agile techniques) for the management of requirements in an agile environment. In this context, the related work lacks in providing guidelines for choosing an appropriate set of agile techniques that help solve a specific agile RE problem [8].

To overcome this gap, we have contributed a modeling language for supporting organizational aspects of agile RE. It provides a visualization technique, which allows us to model the organizational environment, in which agile techniques are applied. The model provides an intuitive representation for the analysis and selection of best solutions to carry out an effective agile RE management.

3. Modeling Language for Agile Requirements Engineering

This section presents the modeling language for supporting organizational aspects of agile RE by means of introducing a profile for the agile RE metamodel [9]. The modeling language enables the analysis of the organizational environment in terms of agile RE in a systematic manner, since it identifies dependencies among people working in such organizational environment as well as evaluates the impact that applying agile methodologies can have.

The aim of our modeling language is to provide a toolkit to model agile RE concepts and relationships in a real world context. For that purpose, we have created a modeling language by means of an UML profile, which can be used by CASE (Computer-Aided Software Engineering) tools.

3.1. Creating a Profile for the Agile RE Metamodel

Based on the analysis of the state of the art of agile RE, an agile RE metamodel [9] was defined by means of MOF (Meta Object Family). This metamodel enables us to create a common language for agile RE and provides an overview of generic concepts in the field. Moreover, it allows analyzing an organizational environment in terms of how an agile RE process is carried out to develop a product.

We used the UML notation for building our profile (see Fig. 1). Then, we followed the process of creating a profile described by Garcia-Garcia [25] and utilized the tool Enterprise Architect[1] (EA) with the Model Driven Generation (MDG) Technology Builder. The result is an add-in for EA, which can be used for creating domain specific models in the area of agile RE.

3.2. The Agile RE Profile

Fig. 1 presents the profile for agile RE using UML notation. It contains metaclasses from UML as well as the stereotypes defined for the agile RE profile, whereas the tagged values are the attributes of the agile RE profile.

Each stereotype is defined as an extension of an UML metaclass. The stereotypes *Impact* and *Methodology* are UML <<Enumeration>>s. *OrganizationalEnvironment*, *AgileREProblem*, *AgileREPattern*, *Domain*, *System* and *ContextOfUse* are UML <<Class>>es.

[1] www.sparxsystems.eu/start/home/

Stakeholder, *User* and *AgileTeam* are UML<<Actor>>s. Besides, the stereotype *AgileREActivity* is defined as a UML <<Activity>>.

Next, we will briefly explain each element of the agile RE profile focused on the agile RE metamodel. A full description of the metamodel can be found in Schön [9].

Methodology. An important aspect in agile RE is the applied methodology (e.g. Scrum, Kanban, XP or HCD). Each methodology comes with its own requirements and has an impact on how RE is carried out. For instance, there is a difference between time-boxed approaches like Scrum, or flow-driven approaches like Kanban. Requirements are managed in a different manner and therefore, the methodology needs to be considered while modeling agile RE.

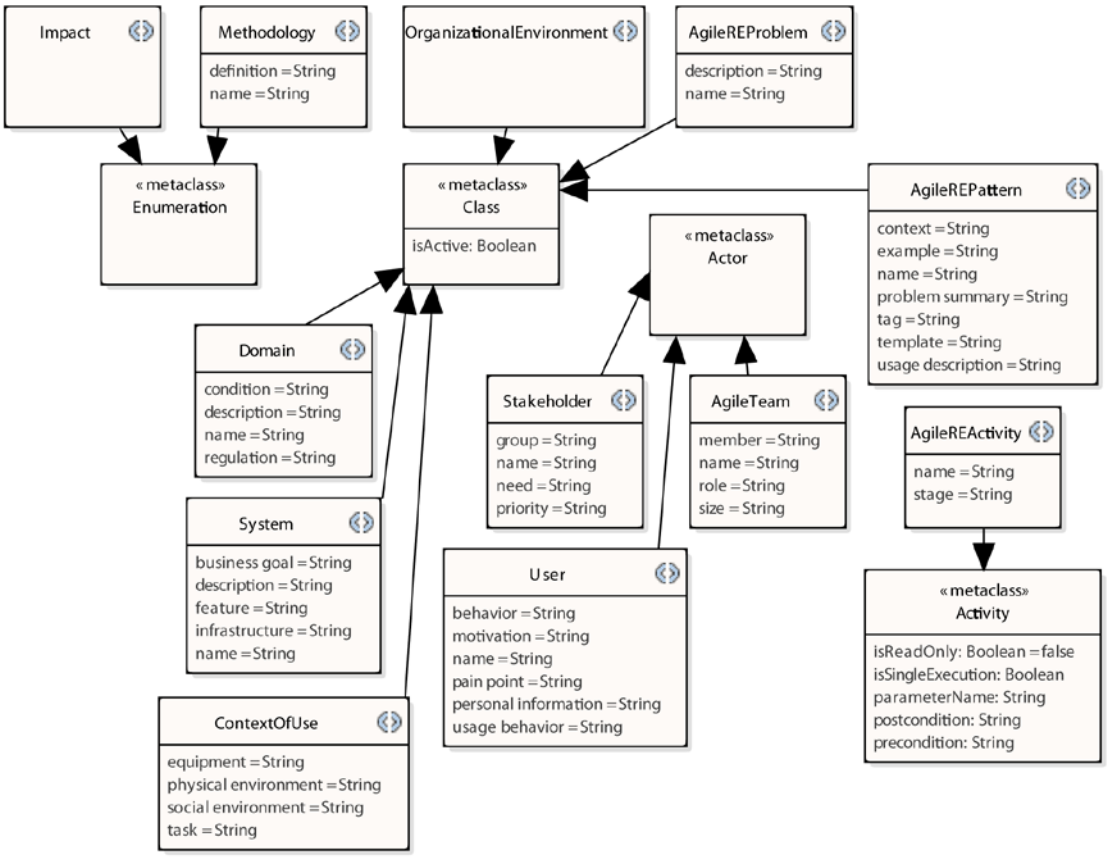

Fig. 1 Profile for agile RE metamodel

Impact. Due to its iterative and incremental character, agile methodologies influence the way requirements are managed in an organizational environment. Impact is filled during runtime and describes how requirements are managed. For instance in Scrum, requirements are managed by a Product Backlog, whereas in Kanban, requirements are managed by a Kanban board.

OrganizationalEnvironment. It describes the surroundings or conditions in which the product development takes place. The organizational environment is changing because of the different people involved in the process (user, stakeholder or agile team).

AgileREProblem. This modeling element describes problems that occur in an agile environment in terms of RE. Agile RE problems appear during system development in an agile context. Examples of agile RE problems can be *"continuous management of requirements"* or *"losing sight of the big picture during the implementation of complex requirements"* (see [8]).

Domain. The requirements for each system differ due to the diverse domains in which the system is used. The concept *Domain* becomes more important in the era of Industry 4.0. As an example, it can be highlighted that there are a lot of different IoT (Internet of Things) platforms on the market that need to be customized for a specific domain (e.g. automotive, utilities or wind energy).

Stakeholder. It refers to individuals or organizations having a right, share, claim or interest in a system or in the characteristics it should have in order to meet their needs and expectations. This can be for instance management, sales, marketing or customer. Continuous involvement of stakeholders is very important to ASD in order to develop a system that fulfills the expectations and needs of both customers and users.

AgileTeam. It groups those people who are responsible for system development, including roles like developer, User Experience (UX) designer, tester, Agile Coach, Scrum Master and Product Owner.

AgileREPattern. It is composed of a recurring problem and a solution description. In particular, an agile RE pattern [9] consists of an agile RE problem and one or more agile techniques that support solving the problem. For instance, the agile RE problem *"continuous management of requirements"* can be solved by the agile RE patterns *"continuous refinement meetings with stakeholders"* or *"sprint review meetings"*.

System. It is a combination of hardware, software and/or services that describe the product.

User. It is a person who interacts with the system. S/he comprises a specialized form of a stakeholder and is valued by an additional stereotype in the agile RE profile. The user is in the center of product development within a value-driven organization where HCD plays an important role.

AgileREActivity. It is an action carried out in terms of RE in an agile environment. Agile RE activities describe actions that occur in relation to requirements management. They can be used to categorize agile RE patterns.

ContextOfUse. The user is in a context of use during the usage of the system. This context of use is defined by [5] and comprises: users, tasks, equipment (hardware, software and materials) and the suitable physical and social environments to use the system.

4. Application of the Modeling Language in Industry

In this section, we demonstrate how the modeling language for supporting organizational aspects of agile RE can be used in industry. For this purpose, we have created a domain specific model for agile RE by means of using the profile for the agile RE metamodel (see Fig. 1). The domain specific model covers the analysis of the as-is situation regarding agile RE in the organization, as well as recommendations to improve such situation. Therefore, agile RE problems [8] are detected and appropriate agile RE patterns [9] to solve these problems are suggested.

We have already used the modeling language in consultancy projects in industry. The next subsections will describe a specific case in order to exemplify the application of the modeling language.

4.1. Background of the Project

This case deals with a project that was carried out in a medium-sized IT company, located in Germany, specialized in e-commerce, mobile apps and Software as a Service (SAAS) tools. The project team consisted of twelve members (one team leader, one project manager, two visual designers, two User Experience experts and six developers). The aim was the relaunch of an internet-based newspaper portal in a period of six months in 2013/2014 [6].

4.2. Domain Specific Model for Agile RE by means of a Modeling Language

We conducted an analysis of the organizational environment with the aim to improve the existing agile RE approach within the IT company. To this end, we used our modeling language for creating a visual representation of the organizational environment, as Fig. 2 shows. This figure represents the domain specific model for agile RE in a Kanban-based environment. For illustrating the instance, we used our profile created in EA.

As it is noticed, the applied methodology integrates Kanban and HCD. The flow-driven approach of Kanban has an impact on the way requirements are handled. The domain where the system is used can be classified as e-commerce in the publishing sector.

In light of this, we can state that the system to be developed is an internet-based newspaper portal. As the analysis of the organizational environment is concerned, we observe a lot of different stakeholders, who are involved in the development process of the system. The stakeholder groups comprise managers, investors or shareholders, among other roles.

In the scope of the agile RE metamodel, the user is a special type of a stakeholder, that means, an experienced web user who is interested in news. Printed media is too slow for her/him and s/he appreciates consuming videos and additional interactive content. With regard to the context of use that involves the user, we realize that his/her main task consists in retrieving news, thus s/he tries to be informed of contemporary issues. Therefore, the first touch point of users is the home page where they can browse through the teasers and select the interesting ones for further reading. They use equipment consisting in a mobile platform (tablet or smartphone) in most of the cases. Both the physical environment and the social environment of users depend on the situation in which they approach the system. For instance, they can use it either in public transport that may be loud and crowded or in their private gardens, where the sun may be shining on the display.

As mentioned before, the agile team consists of twelve members: one team leader, one project manager, two visual designers, two UX experts and six developers. All the aforementioned information describes the organizational environment regarding the project setting.

After analyzing the organizational environment, we detected several problems concerning agile RE. One of them was *continuous requirements management*. We observed that developers handle their tasks by means of a Kanban board. This supports visualizing the workflow and organizing their tasks. In comparison, we observed that conceptual tasks (e.g. user research, specify user requirements or usability testing) were not visible in the same manner, although they were taking place.

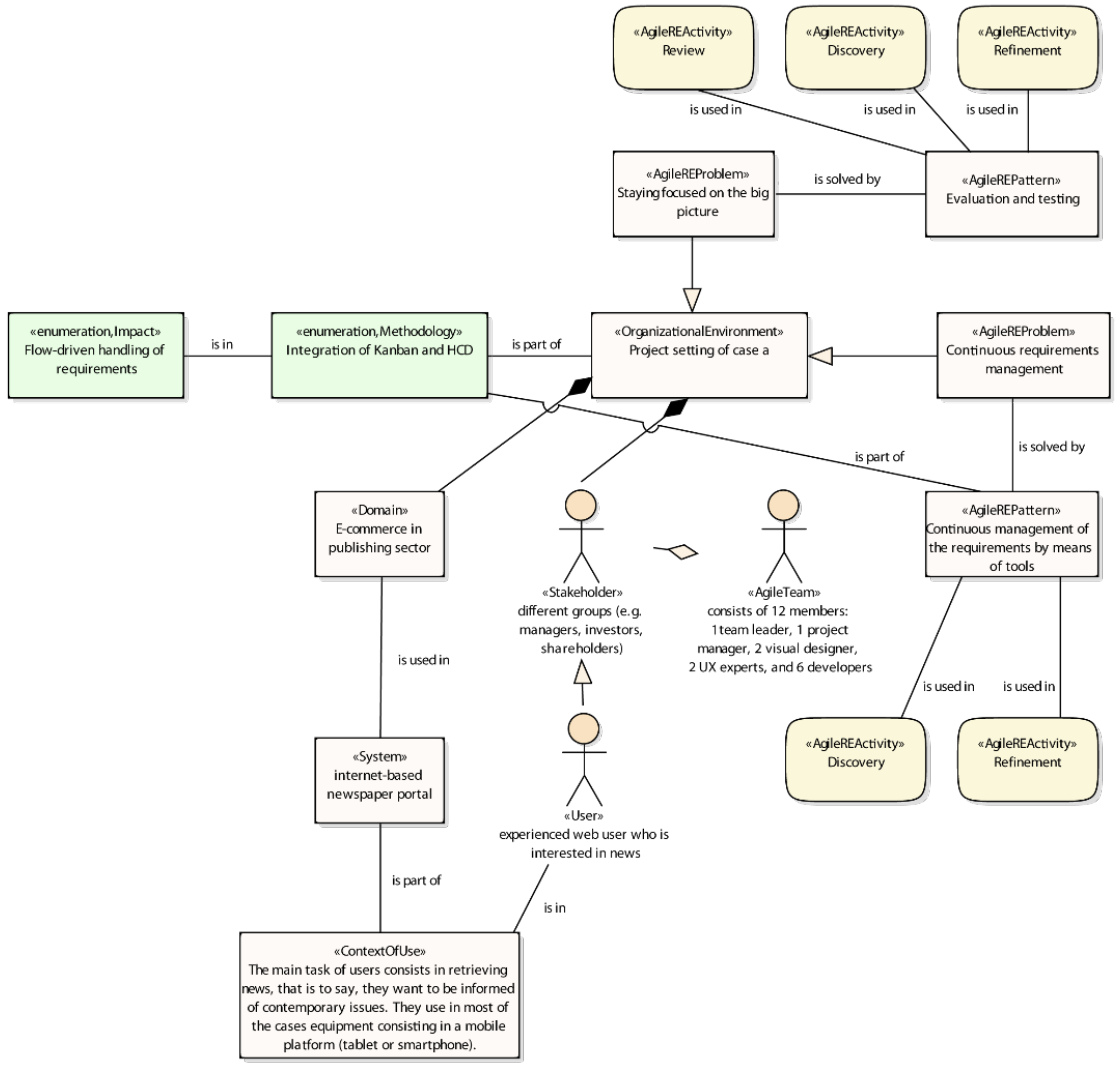

Fig. 2 Domain specific model for agile RE in a Kanban-based environment

Due to the context, we decided to solve this problem by means of using the agile RE pattern *continuous management of the requirements by means of tools*. This pattern can be categorized by the agile RE activities *discovery* and *refinement*, since both activities happen in terms of requirements management while applying the pattern. The product development team benefits from the application of the pattern, since this allows visualizing the workflow of one requirement from discovery to release.

Another agile RE problem, detected along the product development has to do with *staying focused on the big picture*. This problem occurs because working in a Kanban system make people focus on small tasks. This can cause the problem *of losing sight of the big picture during the implementation of complex requirements.* Hence, it is hard to design a positive UX for the user. This problem was handled by applying the agile RE pattern evaluation and testing [9]. This pattern is categorized by the activities *review*, *discovery* and *refinement.* The regular release evaluation helped the product development team concentrate on the big picture and allows carrying out usability and UX testing [26], [27], [28] continuously. Therefore, a work in progress (WIP) limit is introduced to the last column (*Done*, see Fig. 3) of a Kanban board. The release evaluation should start, once the WIP limit is reached.

4.3. Example of Agile RE Pattern

In the following paragraphs, we will present the applied agile RE pattern *continuous management of requirements by means of tools*, as an example.

Table 1. Agile RE pattern continuous management of requirements by means of tools

Pattern name	Continuous management of requirements by means of tools
Context	Established RE approaches fit into sequential approaches to software development. On the contrary, ASD is used to enhance the ability to deal with changing requirements over the course of time. Agile techniques for continuous requirements management need to be implemented in order to ensure traceability of requirements.
Tag	Discovery, refinement, methods
Problem summary	In ASD, continuous management of requirements is a problem since not all requirements are fixed at the beginning and they may change over the course of the project. • providing a positive UX to the user. • carrying out a release evaluation continuously. • not interrupting the workflow due to scheduling testing activities and organization. • reducing costs for long-term UX testing.
Usage description	The workflow of the system development is visualized by means of Kanban boards for different types of tasks (e.g. UX design, development or operation). One requirement can be tracked along the workflow and its evolution is managed through the whole development process (see Fig. 3). Organizing the management of requirements by means of Kanban boards implies a continuous management and tracking of changes.
Example	Fig. 3 presents the interaction of two Kanban boards. We used a third Kanban board for operation that was placed on the right hand side of the development board. The Kanban boards represent the workflow from design through development of the internet-based newspaper portal. One task from the design board might be split into more than one task on the development or the operation board. The aim of this procedure is to obtain continuous flow within the board and among the boards. The project team use tool support by means of JIRA from Atlassian, so as to work with multiple Kanban boards. The different Kanban boards were displayed on several screens on the wall.
Template	See Fig. 3

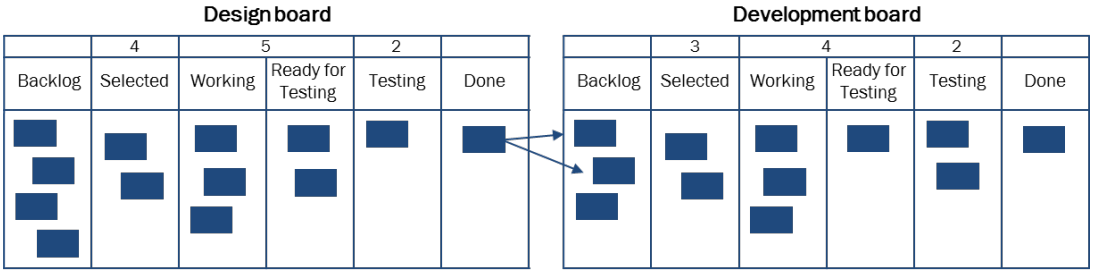

Fig. 3 Example of multiple Kanban boards

It is worth mentioning that more than one of the presented agile RE problems occurred along the development of the project. For instance, "*losing sight of the big picture during the implementation of complex requirements*" is categorized as a problem since the development with Kanban is focused on small tasks. Therefore, HCD activities are integrated by means of a release evaluation [6].

5. Discussion

Due to the application of the modeling language in several projects we learned more about its benefits and limitations. Below, we would like to outline the positive impact that the application of the modeling language has on companies as well as discuss on the limitations we have observed so far.

5.1. Benefits of the Modeling Language for Agile RE

The modeling language for supporting organizational aspects of agile RE provides visualization techniques for analyzing an organizational environment. These visualization techniques speed up analysis due to mechanisms, what allows us to switch the viewpoints in terms of humans, products, processes and projects [10]. The modeling language supports practitioners in terms of organizational development with focus on agile RE. The approach does not provide a detail description regarding the documentation of requirements.

Humans. Visualization of dependencies across stakeholders, users and agile teams, lets us understand how people currently collaborate. This knowledge supports the detection of agile RE problems. Once a problem is detected, we can search for appropriate solutions by enriching the detected agile RE problem with contextual information, which we receive from the visualized domain specific model of agile RE (see Fig. 2). With regard to humans, we can select agile RE patterns to improve the collaboration among them.

Product. The domain specific model, which we receive as a result of the analysis of the organizational environment, allows us to perceive information concerning the product at first sight due to the metaclasses *Domain*, *System*, and *ContextOfUse* (see Fig. 2). This makes us understand the kind of product that is developed as well as the way its users utilize it. For instance, the product development techniques vary for different types of products. In relation to embedded systems in the domain of manufacturing systems engineering, regulatory requirements are more important since the health of humans might be affected. On the contrary, in the domain of e-commerce, requirements concerning the market are more important since those are changing quickly.

Process. With regard to applied processes, the domain specific model (see Fig. 2) visualizes the kind of methodology that is used for product development. Additionally, the metaclass *Impact* gives information about the management of requirements. There is a difference between flow-driven approaches like Kanban or time-boxed approaches like Scrum. Requirements in Scrum are managed by a Product Backlog [2]. Requirements with the highest priority are pulled into the upcoming Sprint. In contrast, Kanban is based on the principle of visualizing the work to be done together with the workflow. There are no restrictions linked to the moment to pull a particular requirement. This eases short-term decisions involving changing priorities.

Project. All metaclasses of the domain specific model provide details for describing a project. The comparison of two domain specific models shows that the organizational environment where the product development takes place differs from each other. This outlines the uniqueness of a project. Nevertheless, the modeling language can also be applied to organizations, where the product development is carried out without using projects.

Summarizing the general benefits of the modeling language for agile RE, we can state that organizations can use the modeling language for identifying internal conflicts, which slow down the value delivery. To this end, the modeling language for agile RE entails increasing the value delivery as well as advancing an agile transition of organizations. Practitioners in the industrial environment need only parts of the modeling language without applying the whole approach. In the beginning people working in the organizational environment will usually be coached regarding new techniques and approaches.

5.2. Limitations

As Fig. 2 shows, the information provided by the metaclasses is not very detailed because we have not accompanied this project for a longer time period. The more time we have for

analyzing the organizational environment, the more we will learn about it and the more detailed information we will be able to collect. Nevertheless, applying our approach has been very beneficial for the project, since we have detected several agile RE problems and we have been able to handle them short-term by means of applying appropriate agile RE patterns.

One further limitation of our approach is caused by the tool support. Our modeling language is available as add-in for EA at this point. However, since we have used an UML notation for the modeling language, visualization techniques can be used without the add-in or implementing the UML profile in other CASE tools.

6. Conclusion and Future Work

This paper has presented a modeling language for supporting organizational aspects of agile RE. Besides, we have shown how industry benefits from applying the modeling language. The modeling language for agile RE contributes visualization techniques for information systems engineering. It allows visualizing domain specific models for agile RE. The visualizing techniques improve the agile way of working since they foster collaboration among people. Practitioners can use the modeling language as a tool for improving their RE and their ability to reflect on problems. Visualization of applied domain specific models enables organizations to analyze their organizational environment in terms of existing agile RE problems. Moreover, it eases the selection of agile RE patterns, which are used for solving agile RE problems.

Future research may specifically measure the outcome of the application of the modeling language. Currently, we are using it in our projects in industry. In this context, we are focused on identifying metrics in order to measure the impact on an objective level. First experiences indicate that the application of the modeling language reduces the effort of orientation and knowledge transfer when new people become part of a running project.

Acknowledgements

This research has been supported by the Pololas project (TIN2016-76956-C3-2-R) of the Spanish Ministry of Economy and Competitiveness.

References

1. VersionOne Inc.: 11th Annual State of Agile Report. (2017).
2. Schwaber, K.: Agile Project Management with Scrum. Microsoft (2004).
3. Anderson, D.J.: Kanban - Successful Evolutionary Change for your Technology Business. Blue Hole Press. (2010).
4. Beck, K.: Extreme Programming Explained: Embrace Change. Addison-Wesley (2000).
5. International Organization for Standardization: ISO 9241-210:2010 - Ergonomics of human-system interaction - Part 210: Human-centred design for interactive systems. (2010).
6. Schön, E.-M., Winter, D., Uhlenbrok, J., Escalona, M.J., Thomaschewski, J.: Enterprise Experience into the Integration of Human-Centered Design and Kanban. In: Proceedings of the 11th International Joint Conference on Software Technologies (ICSOFT 2016). pp. 133–140. , Lisbon, Portugal (2016).
7. Pfeiffer, T., Hellmers, J., Schön, E.-M., Thomaschewski, J.: Empowering User Interfaces for Industrie 4.0. Proc. IEEE. 104, 986–996 (2016).
8. Schön, E.-M., Winter, D., Escalona, M.J., Thomaschewski, J.: Key Challenges in Agile Requirements Engineering. In: Baumeister, H., Lichter, H., and Riebisch, M. (eds.) XP 2017, LNBIP 283. pp. 37–51 (2017).
9. Schön, E.-M.: A Framework for Modeling and Improving Agile Requirements Engineering, PhD thesis, University of Seville (2017).
10. Schön, E.-M., Thomaschewski, J., Escalona, M.J.: Agile Requirements Engineering: A Systematic Literature Review. Comput. Stand. Interfaces. 49, 79–91 (2017).

11. Royce, W.W.: Managing the development of large software systems. In: IEEE WESCON. pp. 1–9 (1970).
12. Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., Marick, B., Martin, R., Mellor, S., Schwaber, K., Sutherland, J., Thomas, D.: Manifesto for Agile Software Development, http://www.agilemanifesto.org/ (2001).
13. Inayat, I., Salim, S.S., Marczak, S., Daneva, M., Shamshirband, S.: A systematic literature review on agile requirements engineering practices and challenges. Comput. Human Behav. 51, 915–929 (2015).
14. Soares, H.F., Alves, N.S.R., Mendes, T.S., Mendonca, M., Spinola, R.O.: Investigating the Link between User Stories and Documentation Debt on Software Projects. In: 2015 12th International Conference on Information Technology - New Generations. pp. 385–390. IEEE (2015).
15. Abdullah, N.N.B., Honiden, S., Sharp, H., Nuseibeh, B., Notkin, D.: Communication patterns of agile requirements engineering. In: Proceedings of the 1st Workshop on Agile Requirements Engineering - AREW ’11. pp. 1–4. ACM Press, New York, New York, USA (2011).
16. Buchan, J.: An Empirical Cognitive Model of the Development of Shared Understanding of Requirements. In: Communications in Computer and Information Science. pp. 165–179 (2014).
17. Maguire, M.: Using human factors standards to support user experience and agile design. In: Proceedings - International Conference, UAHCI 2013, Held as Part of HCI International 2013, Las Vegas, NV, USA. pp. 185–194 (2013).
18. Adikari, S., McDonald, C., Campbell, J.: Reframed contexts: Design thinking for agile user experience design. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (2013).
19. Obendorf, H., Finck, M.: Scenario-based usability engineering techniques in agile development processes. In: Proceeding of the twenty-sixth annual CHI conference extended abstracts on Human factors in computing systems - CHI ’08. p. 2159. ACM Press, New York, New York, USA (2008).
20. Kautz, K.: Participatory Design Activities and Agile Software Development. In: IFIP WG 8.2/8.6 International Working Conference. pp. 303–316 (2010).
21. Näkki, P., Koskela, K., Pikkarainen, M.: Practical model for user-driven innovation in agile software development. In: Proceedings of the 2011 17th International Conference on Concurrent Enterprising (ICE 2011). pp. 1–8. IEEE (2011).
22. Memmel, T., Gundelsweiler, F., Reiterer, H.: Agile Human-Centered Software Engineering. Proc. 21st Br. HCI Gr. Annu. Conf. People Comput. HCI...but not as we know it - Vol. 1. 167–175 (2007).
23. Rivero, J.M., Grigera, J., Rossi, G., Robles Luna, E., Montero, F., Gaedke, M.: Mockup-Driven Development: Providing agile support for Model-Driven Web Engineering. Inf. Softw. Technol. 56, 670–687 (2014).
24. Olsson, H.H., Bosch, J.: Towards Continuous Customer Validation: A Conceptual Model for Combining Qualitative Customer Feedback with Quantitative Customer Observation. LNBIP. 210, 154–166 (2015).
25. García García, J.A.: Una propuesta para el uso del paradigma guiado por modelos (MDE) para la definición y ejecución de procesos de negocio, PhD thesis, University of Seville (2015).
26. Hartson, R., Pyla, P.S.: The UX Book: Process and Guidelines for Ensuring a Quality User Experience. Morgan Kaufman Publ Inc, Amsterdam, and Boston (2012).
27. Grigera, J., Garrido, A., Rivero, J.M., Rossi, G.: Automatic detection of usability smells in web applications. Int. J. Hum. Comput. Stud. 97, 129–148 (2017).
28. Otaduy, I., Diaz, O.: User acceptance testing for Agile-developed web-based applications: Empowering customers through wikis and mind maps. J. Syst. Softw. 133, 212–229 (2017).