

Enhancing Requirements Engineering in Agile Methodologies by Agent-Oriented Goal Models

Two empirical case studies

Tanel Tenso, Alexander Horst Norta, Hannes Rootsi, Kuldar Taveter, Irina Vorontsova

Department of Software Science
Tallinn University of Technology
Tallinn, Estonia

tanel.tenso@ttu.ee, kuldar.taveter@ttu.ee, alexander.norta@ttu.ee, hannes.rootso@gmail.com, ivorontsova@gmail.com

Abstract— The use of agile methods in software engineering is a standard practice and user stories are established artefacts used for breaking complex system requirements into smaller subsets. However, user stories do not suffice for understanding the big picture of system requirements. While there are methods that try to solve this problem, they lack visual tool support and are too heavy for smaller projects. We have earlier proposed a novel agile agent-oriented modelling (AAOM) method for filling this gap. The AAOM method comprises a visual approach to requirements engineering in agile projects that is based on goal models originating in agent-oriented modelling and connects goals intuitively to user stories. The purpose of the study reported in this article was evaluating the AAOM method for requirements engineering in two real-life case studies. The qualitative evaluation explores the applicability of AAOM for requirements engineering in agile software development processes.

Keywords- *Agile software engineering, goal model, user story, case study.*

I. INTRODUCTION

Agile software engineering is a group of software development methodologies that adhere to the agile manifesto [1]. Agile software engineering methodologies are currently the most widely used methodologies for developing software systems [2]. For example, Ramesh et al [3] describe methodologies of agile software engineering as time-boxed, iterative and incremental. Another characteristic of agile software engineering is a frequent delivery of usable software and collaboration with customers. Additionally, agile software engineering supports self-organizing cross-functional teamwork. These factors play a role in the ability of quick responding to changes.

Requirements engineering (RE) is a process of formulating, documenting and managing requirements for software and comprises requirements identification, analysis, documentation and validation [4]. Since RE is the initial phase in software development, errors detected late in requirements are very costly [5] and produce incorrect software that does not satisfy customer needs.

In agile software development projects [3], RE continues through the lifetime of a system. Several variations of performing RE within agile methodologies or doing agile RE exist depending on the use of one or another agile software

engineering methodology, such as Scrum [6], XP [7], Lean [8], or Kanban [9]. Common for different agile software engineering methodologies is the lack of intuitive alignment between engineered requirements and visual system development support [3]. To overcome this challenge, there are methods for organizing central artefacts for RE in agile software engineering – user stories – into larger structures. For example, Cohn [10] suggests Epics – bigger user stories grouping smaller ones. Epics covering different levels of abstraction are also used in Scaled Agile Framework (SAFe) [11]. Large-Scale Scrum (LeSS) [12] focuses on splitting requirements into smaller Product Backlog Items. Disciplined Agile Delivery (DAD) [13] sums up many agile practices, introducing for RE the term Portfolio, which is a hierarchical list of work items. Scrum of Scrums [14] includes team-level planning and requirements tracing between teams. Lean approach to agile requirements [15] divides requirements according to the level of detail into team, program, and portfolio levels. However, according to the recent study [16], one of the key challenges in agile RE is still not to lose sight of the big picture during the implementation of complex requirements. To address this gap, we define a novel requirements engineering method, namely the agile agent-oriented modelling (AAOM) method [17-18] explained in Section II. In this article, we evaluate the applicability and usefulness of AAOM, by employing a case study research methodology [24, 35], which we apply in two different projects.

The rest of this article is structured as follows. Section II provides a short overview of the method proposed by us for RE in agile projects. Research questions, selection and setup of case studies, data collection and analysis, and validity procedures are described in Section III. In Section IV, we present results and findings from the analysis of the collected data. Finally, Section V concludes the article, discusses limitations, and presents open issues for future work.

II. SHORT OVERVIEW OF THE AAOM METHOD

To facilitate stakeholders having the big picture in sight, we have earlier proposed a novel method for RE called agile agent-oriented modelling (AAOM) [17-18]. AAOM is based on agent-oriented modelling (AOM) [19], which is a holistic methodology for designing socio-technical systems consisting of humans and technical components. RE by means of the

AOM methodology has been elaborated in [20]. We have chosen AOM as the basis for AAOM because the types of models put forward by AOM are intuitively understandable for stakeholders, including non-technical practitioners [19-20]. AAOM focuses on a specific model type out of a larger set of model types put forward by AOM – goal models. A goal model is a hierarchy of functional goals, where to the functional goals are attached quality goals, also known as non-functional goals, that apply to them and the roles that are required for achieving the respective functional goals. Each sub-goal represents a certain aspect of achieving its higher-level goal [19].

General idea of the AAOM method is that the requirements for the software system to be created are first represented as a hierarchy of functional goals along with attached to them quality goals and roles. The goal model is then elaborated into lower-level goals until the leaf-level of the goal tree is achieved. The leafs are then further elaborated into user stories, which are simple artefacts for representing requirements in agile software engineering [21-22]. A user story is a written sentence that describes functionality from the perspective of a system user. The AAOM method makes use of the following format of user stories, which has been adapted from Cohn [21]:

As a <user performing a certain role>, I need <to perform action> to support <achieving a certain goal>. (1)

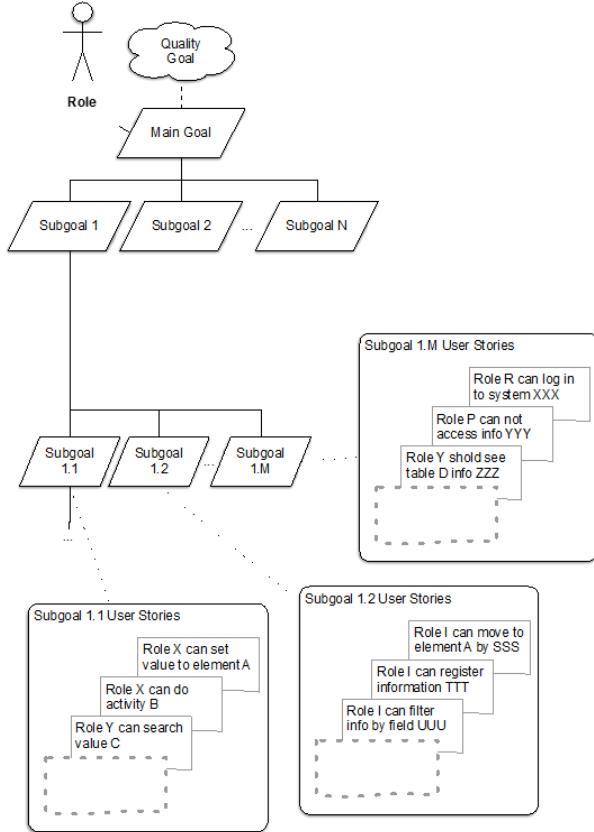


Figure 1. Overview of the AAOM method

Roles and goals included by the user story format (1) are the ones represented by the goal model. An overview of the AAOM method is depicted in Figure 1. As is shown in Figure

1, functional goals are depicted as parallelograms and quality goals are represented as clouds attached to the corresponding functional goals. A quality goal attached to a functional goal applies to the given functional goal and *all* its sub-goals in the goal tree. Goal models also contain roles required for achieving the goals, which are attached as stick men to the corresponding functional goals. User stories are attached to the leaf-level sub-goals.

Figure 2 from [17] depicts how the activities of AAOM fit into the agile software engineering lifecycle.

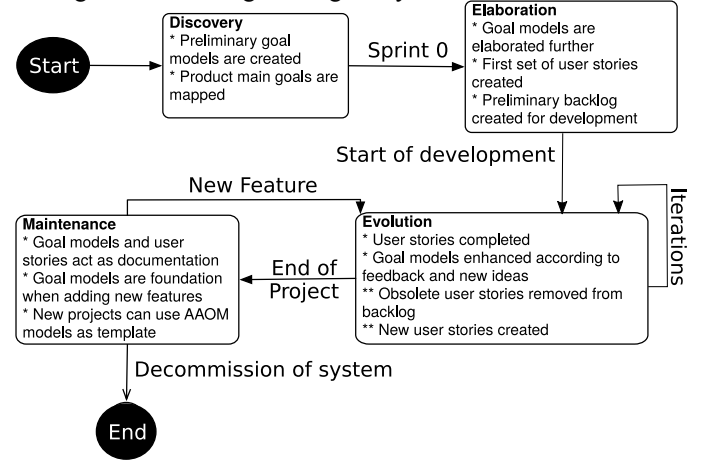


Figure 2. Activities of AAOM in agile software engineering lifecycle

III. RESEARCH DESIGN

As the study reported in this article is based on the research methods described in [22-23], there is no need to repeat here the basics of case study research in software engineering. Therefore, we describe in this section only how the case study method was adapted to the study reported herein.

A. Research Questions

The purpose of the study reported in this article is to evaluate the utility of AAOM for engineering requirements in agile software engineering. Accordingly, the overall research question for the study is: How does AAOM help to improve activities of agile software engineering? This question can be elaborated into the following three sub-questions:

RQ1: What are the benefits of using AAOM?

RQ2: What effect do project setup and tooling have on the usage of AAOM?

RQ3: What aspects of the usage of AAOM need further refinement?

The research questions RQ1-RQ3 establish the basis for selecting appropriate case studies and methods for data collection and analysis. For guiding data collection and analysis, the research questions RQ1-RQ3 are elaborated into more detailed research sub-questions. To find answers to the research sub-questions, interview questions are composed. For example, an interview question corresponding to the research sub-question “Is a visual approach suitable for requirements engineering?” is “What practices and activities of goal modelling are clear and what practices and activities need clarification?” Each research sub-question corresponds to a theme of qualitative case study research. The emerging and identification of themes is treated in Section III-D. The

research sub-questions along with the corresponding themes are represented in Table I. The four final rows in Table I represent the themes that emerged only during conducting the interviews. For these themes, additional research sub-questions were formulated that are represented in Table I.

TABLE I. RESEARCH SUB-QUESTIONS AND THEMES

Research Question	Research Sub-Question	Theme
RQ1	What are the benefits of using the method?	Benefits
	How to improve communication between participants?	Collaborative modelling
	How does AAOM competitively compare to other methods?	Method comparison
	Is a visual approach suitable for requirements' engineering?	Visual representation
RQ2	How much time does it take to follow AAOM practices?	Elaboration sessions
	How long does it take to sketch system requirements with AAOM?	Time taken for modelling
	What is the effect of tooling on AAOM implementation?	Tools usage
RQ3	Do participants understand the method and its details?	Method clarification
Emerged	What negative impressions exist about a project setup?	Drawbacks
	What are the expectations about the benefits of AAOM?	Expectations
	What are the suitability issues of AAOM in information technology projects?	Modelling suitability
	What proposals exist for method improvements, or for project-setup improvement?	New ideas

B. Selection and Setup of Case Studies

The first case study is a software development project on an information system for consumer financing undertaken by one of the major banks in Estonia. In the project, a fully functional submodule of a large banking information system was planned, designed, implemented, and launched. This real-life case study was chosen because at that time the project was just about to start, which enabled monitoring the whole software development process from the beginning to the end. Another reason for choosing this case study was that the case study subjects had participated in a similar project preceding this one and therefore had a fresh experience from a software engineering project making use of other modelling methodologies for grouping user stories mentioned in Section I. As most of RE is usually done at the beginning of a system development lifecycle [4], this project suited perfectly well. It was possible to collaboratively create a goal model and observe the evolution of the goal model and affiliated user stories and the development of the information system. In this project, the working software was released in cycles each lasting for three weeks and the project was finished in three iterations lasting in total for nine weeks.

Four people – one analyst, one customer representative, and two developers – participated in the project. The statistics on demographics was collected on the experience by the participants in their respective roles in software engineering: the analyst was experienced, the customer representative had

an average experience, and both developers also had an average experience. Different roles helped us to evaluate the application of AAOM by different team members, while experience increased the reliability of the gathered information. Low, average, and high levels of experience in both case studies respectively mean that a person has acted less than a year, between one and two years, and more than two years in a specific role. All the subjects participated in the modelling process, RE, development, and collaborative meetings. The data was collected from the participants by interviews.

The second case study is developing a Lost & Found (L&F) mobile app, which belongs to the problem domain of circular economy. The business idea of the mobile app is to reunite lost objects of any type with their rightful owners. Instead of having to rely on lost and found offices at police stations, airports, cinemas, and so on, the L&F app is a simple and quick mobile solution to report findings by using the capabilities of a smartphone – a phone camera allows taking an instant photo of a lost item, while simultaneously tracing and storing the location of shooting the photo. The app is also beneficial for people who have lost something – the L&F app can be used for announcing the loss and receiving notifications when someone has found and registered with the app an item with a similar description. In the project, an app at the level of “proof-of-concept” prototype was planned, designed, and implemented.

The L&F app development project followed an agile software development cycle, relying on specific techniques of the Scrum methodology [6], such as planning with user stories, backlog management, and iterative development. Three development iterations took place in which the progress was visualized on a task board and meetings with customers took place at the end of each iteration.

Four subjects – one person playing the roles of both analyst and system developer and three customer representatives – participated in the project. The statistics on demographics was collected on the experience by the participants in their respective roles in software engineering: the analyst was inexperienced, the developer was experienced, and one customer representative was experienced, while the other two customer representatives were inexperienced.

For either case study, we set up as a single case with holistic design [24]. The unit of analysis was the application of AAOM for iterative requirements elicitation and representation. The research team consisted of two researchers working in co-operation, providing peer-review to each other. The procedures included taking part in all meetings between case subjects that were modelling, demonstration, and retrospective sessions. The researchers acted as silent participants taking notes about the usage of AAOM throughout different meetings. Based on the research questions and meeting notes, the research team devised interview questions and conducted interview sessions for gathering qualitative data. The analysis of the gathered data by researchers provided answers to the research questions.

C. Data Collection

Procedures for data collection were selected according to the research sub-questions represented in Table I. First, the members of the research team took part in all meetings between subjects of the case studies, which included modelling, demonstration, and retrospective sessions. All the sessions

were videorecorded. For videorecording, we asked for and received a consent by the participants. Secondly, we carried out interviews with the stakeholders and collected answers to the research sub-questions defined in Table I.

In the first case study, the first round of interviews was performed right after the completion of the goal model and attached to it user stories but just before starting the development. The second round of interviews was performed right after finishing the first iteration. By that time, the customer had received the first minimal viable product and the developers had got used to creating user stories based on the goal model. Because a goal model mostly affects the work of the analyst, the second round of interviews was performed only with the analyst right after the first iteration. The third and last round of interviews was conducted right after finishing the product in three iterations. In the second case study, we conducted only one set of interviews right after completing the three iterations.

Before each interview was conducted, we informed the participants that the interview would be transcribed and returned to the interviewee for verifying that the ideas expressed by him/her in the interview were correct. The participants were also informed on how the gathered data would be used after conducting the interviews and each participant had an option to either agree or disagree with publishing the data originating in the interviews conducted with him/her.

For conducting the interviews, we chose a *semi-structured format* [24-25]. We planned the interview questions in such a way that their order was of no importance and was changeable during the interview, depending on the discussion flow and the answers by the interviewees to the preceding questions. Semi-structured interviews can also provide additional insight beyond interview questions. The research sub-questions and the corresponding themes stated in Table I guided the preparation of the interview questions. Different sets of questions targeted each respective role – customer, analyst, developer – without offering predefined answers. Thus, interviewees could not answer “yes” or “no” but instead had to express their own opinions. Interviews adhered to an hourglass model [24], where an interview begins with broad questions and continues with more specific questions. At the end of an interview, again broad questions are presented.

The interview structure was similar in all interviews and the interviewees were informed about the interview structure during the process. Each interview session lasted roughly for one and a half hours and started with an introduction, followed by presenting role-specific questions. The interviews were audio recorded as MP4 files for subsequent post-interview analysis. In case an interviewee responded to a question only briefly, the researchers asked additional questions on the same topic to gather more insight. Interviews were recorded anonymously so that instead of real names the role names – customer, analyst, developer – were used.

We also gathered work artefacts, such as goal models, user stories, and source code. Since a dedicated development toolkit for AAOM is still under development¹, we employed for the case studies reported in this article a set of freely

available tools. The first tool – Draw.io² – is an online diagramming tool to draw goal models for the case studies. Another tool employed is Trello³ – a collaboration tool to organize project tasks on boards. Trello visualizes tasks in the form of user stories like “post-it” notes in status columns to observe the progress during software development – tasks to do, pending, in progress, completed, and so on. Finally, Bit Bucket⁴ is a free source code hosting service and at the same time a simple wiki and issue manager for a project.

The work artefacts – goal models, user stories, and source code – were used as complementary evidence to confirm or deny statements by the interviewees. Goal models and user stories were relevant for investigating their evolution during the project, as the history of their changes was recorded. Source code was maintained in a version control system that allowed to observe the lines of code (LOC), changes in LOC, time between LOC changes, and links to the corresponding goal model. However, in the research reported in this article, the interviews played the most important role and generated the most relevant output.

The details of conducting the interviews, including the interview questions, are available in the online source [34] for the first case study and in the online source [33] for the second case study.

D. Analysis Procedure

As was stated in Section III-C, different types of data were collected within the case studies reported in this article. We focused on the analysis of the interviews, as they yielded the answers to the main research question about the utility of AAOM and the research sub-questions represented in Table I. The second most important source of information was made up by the goal models created and elaborated during the software engineering process under study. However, due to the page limitations, we will not present the goal model analysis [26] in this article.

For analysing the interviews, they were first transcribed and the results were coded following the guidelines given in [27-28, 36]. Shortly, *codes* are meaningful keywords or phrases extracted from the interviews. Next, codes were grouped by themes, as is prescribed in [24]. A *theme* is an outcome of coding, categorization, and analytic reflection [28]. Some themes are based on research questions, while others were identified during coding. The analysis resulted in 12 themes so that each code belongs to the theme from Table I. Each theme corresponds to the research sub-question shown in Table I. For coding, we used the tool for qualitative data analysis NVivo⁵.

We devised a simple formula to compare the validity of codes against each other for analysing and evaluating aspects of the utility of AAOM. The formula expresses *code value* in terms of three variables: references, sources, and role experience. The first variable *references* indicates how many times a code has been mentioned in interviews. The second variable *sources* shows how many different interviewees have mentioned the given code. Finally, the third variable *role experience* expresses an interviewee’s experience in his/her

¹ <http://www.tud.ttu.ee/im/Msury.Mahunnah/AOM4STS/>

² <https://www.draw.io>

³ <https://trello.com>

⁴ <https://bitbucket.org>

⁵ <http://www.qsrinternational.com/>

role. Each variable has a numeric value. The formula to calculate code value based on the values of the above-mentioned variables is as follows:

$$\text{code value} = (\text{references} * \text{sources}) + \text{role experience}. \quad (2)$$

The higher the values of the variables and the resulting code value are, the higher is the reliability of the corresponding code. For example, if a code has been mentioned in 8 sources where 3 of them are the interviews with experienced subjects and 5 sources are the interviews with subjects with average experience, and the code has been mentioned 10 times, the final evaluation score of the code is calculated as follows:

$$8 (\text{sources}) * 10 (\text{references}) + (3 * 3 + 5 * 2) = 8 * 10 + 19 = 99.$$

Additionally, we attached to each code the attributes of polarity and type. Polarity of a code denotes the emotionality of the code. Polarity can be positive, neutral, or negative. A code's type indicates whether the code is a recommendation or a statement by the interviewee.

The variable *role experience* is extremely useful for evaluating positive or negative statements. If an experienced person praises or disapproves AAOM or its aspects then this is more significant than in case of a person with no relevant experience.

The details of coding are available in the online source [34] for the first case study and in the online source [33] for the second case study.

E. Validity Procedure

To assess the validity of our research, we used criteria proposed by Shenton [29]: credibility, transferability, dependability, and confirmability.

To increase credibility, we employed the following strategies:

- We used a well-established body of knowledge on conducting case studies, mainly [23-24]. To set up and receive useful data from interviews, we used guidelines by Robson [25] and for best practices to analyse gathered data, we considered Saldana [28].
- We conducted interviews with all the participants of both projects covering all different roles. This is considered as a form of triangulation since we captured the viewpoints of all informants [30].
- We used changes in LOC, time between LOC changes, and the corresponding goal models and user stories as anecdotal data sources only to subjectively evaluate some statements received from interviewees. Unfortunately, we were not able to triangulate via these data sources, because there are no appropriate methods for relating goal models and user stories to LOC.
- Before starting the case studies, researchers studied AAOM as the unit of analysis and the contexts where it was applied, which were respectively the problem domains of consumer financing and circular economy.
- To help ensure honesty by subjects, we informed them that the use of the collected data was going to be anonymous and that their voice was going to be

recorded. All interviewees agreed with the recording and with the anonymous usage of the data for evaluating AAOM. The interviewees reviewed the transcribed documents to assure a valid transferral of ideas.

- Three researchers participated in the case study and provided constant peer reviewing to each other. With that setup and constant debriefing among each other, the vision by the researchers was broader than it would have been when working alone.
- The researchers participating in the case study have relevant related research backgrounds. Thus, the credibility is assured by the extensive research experience.

Achieving the second criterion – *transferability* – could be demonstrated to a considerable extent because the study addressed with a similar setup two case studies from two different problem domains and with different participants but with a partially overlapping set of researchers [29]. As it is explained in Section IV, the same research results largely reoccur in both case studies. In addition, the study reported in this article is helpful for other researchers to compare their results with ours [29]. One more factor assuring transferability was the fact that agile teams are limited to the size between 3 and 9 [31], which was also the team size range in our case studies.

To address *dependability*, Shenton [29] recommends a case study report that includes sections devoted to the research design and its implementation, operational details of data collection, and reflective appraisal of the project. In this article, the research design is described in Section III and the details of data collection are presented in Section III-C. We do not evaluate the effectiveness of processes undertaken in the study reported in this article. This will be evaluated as an important part of the future work.

According to Jensen [32], *confirmability* is an accurate criterion for verifying if the researchers have understood a phenomenon from the perspective of the research participants and if they have understood the meanings that the participants have given to their experiences. To address confirmability, we carefully documented the application of the data collection and analysis procedures for further use and cross-checked the two case studies with each other.

The research reported in this article contains a threat to researcher bias because one researcher is the inventor of AAOM – a method under investigation. We are aware of this threat and avoid it with the same means as for credibility and confirmability – by providing detailed descriptions. According to Shenton [29], in a qualitative study, researcher bias is inevitable. We also acknowledge the limitation of one person performing two roles in the second case study, by this possibly removing one friction point between the analyst and developers. This could have been mitigated by assigning another person to that role for which we lacked a sufficient project budget.

IV. RESULTS

We found the answer to the main research question of the study reported in this article – how does AAOM help to improve activities of agile software engineering? – by

answering the research sub-questions RQ1-RQ3 specified in Section III-A and the research sub-questions represented in Table I. We additionally considered any non-expected data found during the interviews. All themes and codes discussed in this section are presented in more detail in the online source [34] for the first case study and in the online source [33] for the second case study.

A. Benefits of using AAOM (RQ1)

To find answers to RQ1, we used four themes that address direct benefits of AAOM perceived by the participants. We evaluated the results by means of the formula (1) devised in Section III-D.

Most codes analysed under the first theme *Benefits* are positive statements about using AAOM. Participants of both case studies express that AAOM provides a secure feeling for project direction and improves mutual communication within a project. Participants of the first and second case study respectively find that AAOM makes extracting information from customer better and helps to discover new angles. Participants of the first case study also claim that AAOM helps them to understand the value delivered and estimate the work ahead and the system was developed according to the goal model. One participant – developer – of the first case study claims that a user story is oversized for software development. On one hand, the developer is right because in agile software engineering, user stories are often elaborated into more fine-grained units – tasks. On the other hand, since this is claimed only once by a developer with average experience, the statement can also be caused by inadequate training in agile software engineering.

The next theme *Collaborative Modelling* relates to the hypothesis that AAOM improves communication between customers and the development team by working together on requirements elicitation. The highest ranked codes are all positive recommendations, which confirms the expectations set for AAOM. The participants of both case studies confirm that AAOM improves the understandability and facilitates involving participants and having everyone on the same page. Participants of the second case study further emphasize that AAOM helps to pinpoint problems.

The theme *Method Comparison* gathers the experience by the participants with other methods like AAOM. The participants of the first case study claim that using AAOM is better than making notes. Unfortunately, the experience with similar methods is low among participants of the second case study. Consequently, the overall comparison with other methods under this theme is not sufficient and does not yield reliable results.

Finally, the only code with a high evaluation score under the theme *Visual Representation* for either case study expresses that visual representation by a goal model is suitable.

B. Effects of the Project Setup and Tooling on AAOM (RQ2)

The first theme corresponding to the research question RQ2 addresses explaining the setup of elaboration sessions and finding the effects of applying the AAOM method. The next theme is related to temporal measures, explaining the time required to create and manage AAOM models. The final theme is concerned with the effects of using software-based tools for the application of AAOM.

Which practices are favoured and which ones need improvement was evaluated by the formula (1) devised in Section III-D. The theme *Elaboration Sessions* appears only in the interviews for the first case study. This theme addresses the content, duration, and suitability of sessions for applying AAOM. The evaluation results suggest that the session length, which was one and a half hours, was selected correctly for the first case study.

The theme *Modelling Time Usage* is concerned with the interview questions related to the time spent on modelling activities carried out using the AAOM method. Participants of both case studies state that AAOM enables moving fast from idea to development. Participants of the first case study additionally claim that with AAOM, time is used effectively. For the first case study, two conflicting statements were made by the analyst about refining goal models as being fast and slow, respectively. For the second case study, in this respect there was just one positive statement by the analyst confirming that refining goal models was fast. It is also noteworthy that for the second case study, one neutral statement was made expressing that moderate time had to be spent until an idea had been formulated as a collection of user stories. A few contradictory results can be explained by the fact that this theme was measured in terms of the subjective feelings by the participants about the time spent on modelling activities.

The theme *Tools Usage* addresses the importance of software-based tools in applying AAOM. Participants of the first case study strongly state that dedicated tool support for AAOM is needed. Similarly, participants of the second case study state that using freely available tools is provisionally satisfactory, while having an integrated tool suit would reduce the amount of work needed for RE by AAOM. Available commercial tools were considered as better by participants of both case studies. We can conclude for this theme that there is an interest to use for AAOM new tools that are better tailored for this method compared to the free tools chosen for the case studies.

C. Further Needs for Refining AAOM (RQ3)

The theme *Method Clarification* corresponds to the research question RQ3 that addresses which practices of AAOM are clear and which ones need further explanation or redefining.

On the positive side, positive statements dominate claiming that the usage of goal models and user stories is clear. The usage of quality goals in the first case study is clear for the customer and analyst and somewhat unclear for the developers. We will further investigate these findings in our future research. Finally, the participants of the second case study claim that the sequence of activities prescribed by AAOM and the usage of roles are clear.

On the other hand, there are also a few contradicting negative statements about the same aspects. First, we gathered contradictory information when trying to find out if creating user stories for the goals at the lowest level of the goal tree was clear. While participants of the first case study claim that proceeding from goals to user stories is clear, participants of the second case study express that proceeding from goals to user stories is unclear. Subjects of the second case study further claim that the link between quality goals and user stories is unclear. Our hypothesis for explaining these contradictions is

that they are related to the level of experience, because the interviews show that proceeding from goals to user stories and creating the link between quality goals and user stories is much clearer among the participants of the first case study, who are generally more experienced. Analysing these contradictions deeper is another topic for our future research work.

D. Emerged Themes and Results

The emerged codes result from the coding procedure that is explained in Section III-D.

The theme *Drawbacks* appears only in the interviews for the first case study. Under this theme, the participants express their opinions about the project setup that is not directly related to the AAOM method but still influences its usage. The participants also express that experienced participants are needed to fully benefit from AAOM. In other words, AAOM is not necessarily intuitively fully understandable for all participants in RE. To increase understandability, a solution could be to write a better user guide for the AAOM method, as is also suggested by the participants. Furthermore, the analyst is identified as playing a vital role in the usage of AAOM by having the biggest responsibility in modelling activities. As a final statement worth noting here, a doubt is expressed about the suitability of the method in smaller projects due to the possible method overhead.

The theme *Expectations* shows the positive impression by the participants of both case studies. Participants of the first case study express that AAOM supports common expectations by stakeholders. The customer and analyst of the first case study claim that AAOM enhances customer's ability to adapt to an already delivered product. Developers of the first case study state that using AAOM involves less unexpected changes. The latter statement is made by both developers of the first case study because they are the ones who need to implement changes. For the second case study, the most highly rated expectations are introducing updates to the formats of goal models and user stories, and having working results and extensible implementations. A negative statement is that more participants are needed to attain the goals pertaining to the project setup phase as compared to the implementation phase.

The results captured under the theme *Modelling Suitability* support the hypothesis claiming that the AAOM method helps to focus on elicitation of objectives and organizing thoughts to express expectations by a customer. Under this theme, participants of the first case study claim that goal modelling is targeted at an analyst and customer rather than a developer, the goal models did not change during development, and the participants would use goal models in other projects. Participants of both case studies claim that goal modelling is good for organizing thoughts. Participants of the second case study further claim that goal modelling clarifies what needs to be done.

The theme *New Ideas* appears only in the interviews for the first case study. Statements gathered under this theme provide novel suggestions to use AAOM also for business modelling rather than just in information technology projects and to use goal models also for documenting systems.

V. DISCUSSION AND CONCLUSIONS

In this article, we evaluated AAOM as the novel method for agile RE in two real-life agile software engineering projects. The first project was devoted to developing an information system for consumer financing by a bank. The second project was developing from scratch a mobile app for circular economy. A case-study based qualitative research was applied for the evaluation. Interviews with the participants of both projects provided the most important input for evaluating the AAOM method. Interviews were coded and analysed to find answers to the research questions and to identify new knowledge outside the scope of the research questions.

According to the most notable evaluation results, the *benefits* of using the AAOM method are *a secure feeling for project direction, mutual communication, improving information extraction from a customer, and discovering new angles in requirements for the project*. The methodology supports *collaborative modelling by involving participants, having everyone on the same page, improving understandability, and pinpointing problems*. The *visual representation* provided by goal models is *intuitively comprehensible*. Furthermore, time spent on AAOM activities is found to be adequate and the overhead during that process is marginally low. For projects on a small budget, free tools with manual integration suffice while there is a desire for integrated tool support tailored for AAOM. The usage of goal models and user stories in AAOM is clear.

Unclear procedures that need better guidance or redefining, pertain to the way of finding the lowest level of goal models from which user stories are to be created, and relating quality goals to user stories. One of the discovered factors outside the scope of the research questions is that before using AAOM effectively, longer experience in software engineering is needed or additional guidance should be available. Finally, the analyst has been identified as the central role in applying AAOM.

The study reported in this article also has several limitations. The first limitation is the lack of interview-based feedback with respect to other methodologies of agile software engineering and methods of RE. This could reveal how well the AAOM method is compatible with other agile software engineering methodologies and RE methods. The second limitation is related to the research bias towards supporting factors in favour of AAOM usage because one researcher is the inventor of AAOM. The third limitation is that the information collected from the second case study is biased towards the opinions by the customer because in the second case study three participants out of four performed the customer role. Also, in the second case study only two of the four participants had experience in participating in software engineering projects.

Further studies and investigations must apply the AAOM method in diverse domains for demonstrating its universal applicability. Based on the feedback from participants of the case studies reported in this article, an improvement of the AAOM method must include better explanation of the function of quality goals and particularly how they can be more clearly related to user stories. Since the analyst plays an important role in applying the AAOM method, we also need a deeper understanding of this essential role. Finally, in the future we

will also evaluate the effectiveness of processes undertaken in the study reported in this article.

ACKNOWLEDGMENT

We thank LHV Bank for allowing us to conduct research using their resources in the first case study, Mari-Ann Vellerand and Hando Rand for donating their time to participate in the second case study and Pekka Abrahamsson and Dietmar Pfahl for providing constructive feedback on the setup of the case studies.

REFERENCES

- [1] Fowler, Martin, and Jim Highsmith. "The agile manifesto." *Software Development* 9, no. 8 (2001): 28-35.
- [2] "9th annual state of agile survey," 2015. Last accessed on 15 June 2017 at <http://www.versionone.com/pdf/state-of-agile-development-survey-ninth.pdf>
- [3] Cao, Lan, and Balasubramaniam Ramesh. "Agile requirements engineering practices: An empirical study." *IEEE Software* 25, no. 1 (2008), pp. 60-67.
- [4] Hull, Elizabeth, Ken Jackson, and Jeremy Dick. *Requirements engineering*. Springer Science & Business Media, 2010.
- [5] Carlson, Dick, and Philip Matuzic. "Practical agile requirements engineering." In *Proceedings of the 13th Annual Systems Engineering Conference*, 2010.
- [6] Schwaber, Ken. *Agile project management with Scrum*. Microsoft Press, 2004.
- [7] Beck, Kent. *Extreme programming explained: Embrace change*. Addison-Wesley Professional, 2000.
- [8] Poppendieck, Mary, and Tom Poppendieck. *Implementing lean software development: From concept to cash*. Pearson Education, 2007.
- [9] Kniberg, Henrik, and Mattias Skarin. *Kanban and Scrum –making the most of both*. Lulu.com, 2010.
- [10] Cohn, Mike. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [11] Leffingwell, Dean. *Scaling software agility: best practices for large enterprises*. Pearson Education, 2007.
- [12] Larman, Craig, and Bas Vodde. *Practices for scaling lean & agile development: Large, multisite, and offshore product development with large-scale scrum*. Pearson Education, 2010.
- [13] Ambler, Scott W., and Mark Lines. *Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise*. IBM Press, 2012.
- [14] Sutherland, Jeff, Anton Viktorov, Jack Blount, and Nikolai Puntikov. "Distributed scrum: Agile project management with outsourced development teams." In *System Sciences*, 2007. HICSS 2007. 40th Annual Hawaii International Conference on, pp. 274a-274a. IEEE, 2007.
- [15] Leffingwell, Dean. *Agile software requirements: lean requirements practices for teams, programs, and the enterprise*. Addison-Wesley Professional, 2010.
- [16] Schön, Eva-Maria, Dominique Winter, María José Escalona, and Jörg Thomaschewski. "Key Challenges in Agile Requirements Engineering." In *International Conference on Agile Software Development*, pp. 37-51. Springer, 2017.
- [17] Tenso, Tanel, and Kuldar Taveter. "Requirements engineering with agent-oriented models." In *ENASE 2013 – Proceedings of the 8th International Conference on Evaluation of Novel Approaches to Software Engineering: 8th International Conference on Evaluation of Novel Approaches to Software Engineering*, ENASE 2013; Angers; France; 4 July 2013 through 6 July 2013. SciTePress, 2013, pp. 254–259.
- [18] Tenso, Tanel, Alexander Norta, and Irina Vorontsova. "Evaluating a Novel Agile Requirements Engineering Method: A Case Study." *ENASE 2016 – Proceedings of the 11th International Conference on Evaluation of Novel Approaches to Software Engineering*, Rome, Italy, 27-28 April 2016. SciTePress, 2016, pp. 156–163.
- [19] Sterling, Leon, and Kuldar Taveter. *The art of agent-oriented modeling*. MIT Press, 2009.
- [20] Miller, Tim, Bin Lu, Leon Sterling, Ghassan Beydoun, and Kuldar Taveter. "Requirements elicitation and specification using the agent paradigm: the case study of an aircraft turnaround simulator." *IEEE Transactions on Software Engineering* 40, no. 10 (2014): 1007-1024.
- [21] Cohn, Mike. *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [22] Paetsch, Frauke, Armin Eberlein, and Frank Maurer. "Requirements engineering and agile software development." In *Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2003. WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on, pp. 308-313. IEEE, 2003.
- [23] Runeson, Per, and Martin Höst. "Guidelines for conducting and reporting case study research in software engineering." *Empirical software engineering* 14, no. 2 (2009): 131.
- [24] Runeson, Per, Martin Host, Austen Rainer, and Bjorn Regnell. *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.
- [25] Robson, Colin, and Kieran McCartan. *Real world research*. John Wiley & Sons, 2016.
- [26] Horkoff, Jennifer, and Eric Yu. "Analyzing goal models: different approaches and how to choose among them." In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pp. 675-682. ACM, 2011.
- [27] Taylor, C., and G. R. Gibbs. "How and what to code?," 2010. Last accessed on 16 June 2017 at http://onlineqda.hud.ac.uk/Intro_QDA/how_what_to_code.php.
- [28] Saldaña, Johnny. *The coding manual for qualitative researchers*. Sage Publications, 2016.
- [29] Shenton, Andrew K. "Strategies for ensuring trustworthiness in qualitative research projects." *Education for Information* 22, no. 2 (2004): 63-75.
- [30] Van Maanen, John. "The fact of fiction in organizational ethnography." *Administrative Science Quarterly* (1979): 539-550.
- [31] Cockburn, Alistair. *Agile software development: the cooperative game*. Pearson Education, 2006.
- [32] Jensen, Devon. "Confirmability." *The Sage encyclopaedia of qualitative research methods* 1 (2008): 112. Sage.
- [33] Rootsi, Hannes. "Evaluating agile agent-oriented modelling advantages based on customer factoring system development." M.Sc. thesis, Tallinn University of Technology, 2015. Last accessed on 16 June 2017 at <https://digi.lib.ttu.ee/i/?2933>.
- [34] Tenso, Tanel, Alex Norta, and Irina Vorontsova. "Evaluating novel agile requirements engineering method: A case study." Technical report, Tallinn University of Technology, 2016. Last accessed on 16 June 2017 at <https://digi.lib.ttu.ee/i/?4064>.
- [35] Yin, Robert K. *Case study research: Design and methods*. Sage Publications, 2014.
- [36] Strauss, A. and J. Corbin. *Basics of qualitative research: Grounded theory procedures and techniques*. Sage Publications, 1990.