TALLINN UNIVERSITY OF TECHNOLOGY

Faculty of Information Technology

Department of Informatics

Chair of Software Engineering

EVALUATING AGILE AGENT-ORIENTED MODELLING ADVANTAGES BASED ON CUSTOMER FACTORING SYSTEM DEVELOPMENT

Master Thesis

Author: Hannes Rootsi

Student Code: 124465 IABMM

Supervisor: Tanel Tenso, Alexander Norta

Tallinn

2015

Author declaration

I hereby declare that this thesis is based on my own work. All ideas, major views and data from different sources by other authors were only used as reference and/or for research purposes. The thesis has not been submitted for any degree or examination in any other university.

Hannes Rootsi

Date Signature

Abstract

This thesis evaluates agile agent-oriented modelling (AAOM) for requirements engineering (RE) in agile software development processes. As the author of this thesis and case study participants see gaps in currently used modelling methods, AAOM was experimentally used in a finance-industry development project.

Rounds of interviews are conducted with development team to find out if AAOM poses a benefit over alternatives or not. A case study research on financial-industry case is conducted in order to evaluate this. Research subjects are customer and the team developing software. Interviews are transcribed and coded. Nodes and themes resulting from coding are analyzed and results are reported in this thesis.

The conclusion is that AAOM has many benefits over alternatives. It improves understandability and mutual communication. It also helps to create a big picture of the developed software. Visual representation of AAOM is suitable and understandable to all case subjects. However proper tooling is needed in order to help automatically visualize and manage goals. Additionally some new ideas were brought out by research subjects about applying AAOM for product management goals, not only IT goals. Also case study participants brought out that AAOM could be used to structure already existing system documentation.

As another similar research has been conducted in a different industry with positive results recently, it concludes that AAOM improves development process and can be used by different sizes of teams.

The thesis is in english and contains 45 pages of text, 6 chapters, 4 figures, 10 tables.

Annotatsioon

See lõputöö hindab agiilset agent-orienteeritud modelleerimist nõuete kogumiseks agiilses tarkvara arenduses. Kuna selle töö autor ja uurimuses osalejad näevad puudujääke hetkel kasutatavates modelleerimise metoodikates, kasutati eksperimentaalselt agiilset agent-orienteeritud modelleerimist finantssektori tarkvara arenduses.

Arendusmeeskonnaga viiakse läbi intervjuud eesmärgiga selgitada välja, kas AAOM loob alternatiivsete lahenduste ees eeliseid. Tulemuse hindamiseks kasutatakse juhtumiuuringul põhinevat meetodit finantssektori ettevõtte näitel. Juhtumisuuringu subjektid on tellija ning arendusmeeskond. Intervjuud transkribeeritakse ning seejärel kodeeritakse. Saadud koodid ja teemad analüüsitakse ning esitatakse tulemustest raport.

Lõputöö järeldus on, et AAOM omab palju eeliseid oma alternatiivide ees. See parendab arusaadavust ja ühist suhtlust ning aitab luua suurt pilti arendatavast tarkvarast. Visuaalne kujutamine AAOM's on sobilik ja arusaadav kõigile uurimuses osalejatele. Siiski on vajalik abivahendi olemasolu visualiseerimise lihtsustamiseks. Uurimuse käigus toodi osalejate poolt välja uusi mõtteid, et AAOM'i saab kasutada ka tootejuhtimiseks, mitte ainult IT eesmärkide modelleerimise jaoks. Lisaks pakuti välja, et AAOM'i abil on võimalik ka olemasolevat dokumentatsiooni struktureerida.

Kuna hiljuti on teise valdkonna näitel koostatud sarnase sisu ja tulemustega lõputöö, võib järeldada, et AAOM parandab arendusprotsessi ning seda on võimalik kasutada erineva suurusega arendusmeeskondade poolt.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 45 leheküljel, 6 peatükki, 4 joonist, 10 tabelit.

Glossary

AAOM – Agile Agent-oriented modelling

RE – Requirements Engineering

LOC – Lines of code

COF – Consumer financing

Contents

Figures

Tables

1 Introduction

1.1 Requirements Engineering

This thesis is written to validate agile agent-oriented modelling [1] (from here on AAOM) for the requirements engineering (from here on RE) methodology [2] [3]. Effective RE is an important part of the software development process and it consists of gathering requirements from customers, documentation, analyzing and presenting to developers, and so on [4]. As customers often do not have an information technology background, requirements created by them can often be inaccurate, scattered and/or too general for development team [4]. According to [4], RE helps to focus the needs of customers and to make them comprehensible for the development team. AAOM helps with performing RE more effectively thanks to the support of visual representation and collaborative modelling [3].

1.2 AAOM

As mentioned previously, AAOM helps with RE process [3]. In AAOM, a software development team with customer collaboratively create a goal model of hierarchically structured functional and non-functional goals [3]. An important part of it is a visual representation of created models that help a team to better understand the big picture and relations between more general goals and specific technical aspects of a system [5]. The model can already contain roles, so it is possible to understand which role should be able to achieve specific goals [3].

1.3 Requirements engineering in agile

As name suggests, modelling in AAOM is agile. This means that agile practices are applied in RE processes. Agile practices mainly consist of good face-to-face communication between all team members, being prone to changes (also changes to requirements based on customer input), iterative development (and therefore iterative RE) and constantly participating in collaborative meetings during development [6]. One of the most important artifacts of AAOM, user stories (described in chapter 2.2) are also part of Agile methodologies [6].

1.4 Problem statement

In our modern world, fast-changing technology and increased competition create the need for different kinds of software, which means that the demand for system development is ever-increasing [4]. However, as computer science is a relatively new industry, it is not as much regulated as for example the construction industry is. The author sees failure of communication between the client and the development team as one of the biggest problems in the development process. It is not rare that a customer explains one thing, the development team lead understands something else and programmers write something entirely different from both customer and team lead understandings. This sequence of misunderstanding has even been cartooned countless times, as seen in Figure 1 Inadequate communication. The customer explains a totally different thing that is really needed. The team lead understands another thing and the programmer writes a completely different product.

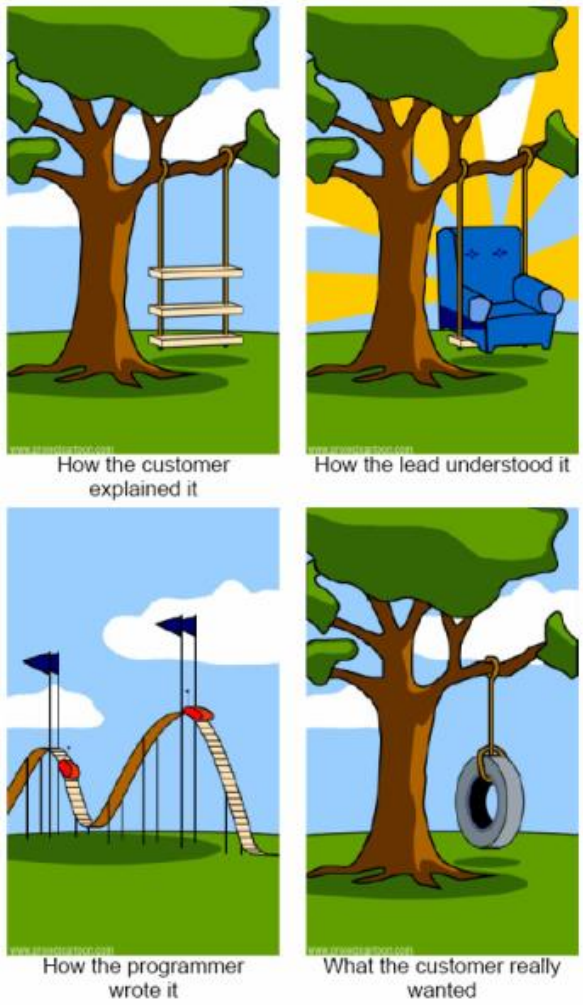


Figure 1 Inadequate communication

The problem is not always inadequate communication but also the fact that developers do not always fully understand the true objectives of written software [7]. In other words, developers do not see the big picture of the system and that is a recipe for failure. Also the reason for low customer satisfaction may be the fact that a system specification written by analysts is not structured and too messy to understand and track later, brought out by the subjects of current research.

The opposite of endless and messy documentation is having very little or no documentation at all. It is encouraged by the Agile Manifesto [6] itself that states the following "Working software over comprehensive documentation". Many teams take this too seriously and think of requirements engineering as old-fashioned, overhead and waste [2].

As software development is expensive, these problems result in delays in delivery and therefore additional expenses for the customer. If the contract is signed with constraints about a finished product and the developer has not met these constraints, the developer may have to pay agreed agreement fee. Also, because the project deadline is pushed and the customer has taken into account the planned product launch and current economical state, the financial damage could potentially be multiple times higher than the cost of software in the first place.

In agile software development RE can vary according to the framework used while it usually consists of face-to-face communication, prototyping, writing user stories, writing specification into documents and updating documentation during development iterations [2]. However, the author of this thesis thinks that none of the existing approaches to RE is equally satisfactory for all of the roles in a modern software development process.

1.5 Research Questions

The main question addressed by this thesis is "How does AAOM create an advantage in a software development process over alternative methods?". As it is a relatively general question, it is divided into more specific sub-questions addressing HOW this can be done.

The first sub-question is "How to achieve a better understandability and mutual communication with AAOM to create an advantage in a software development process over alternative methods?". Based on [4] the main project success factor is user involvement and collaborative modelling improves that. During collaborative modelling, the whole team meats and

communicates with each other and together improve goal models. [4] also brings out reasons for project failure and one of the main reasons is lack of user involvement and also lack of resources. This research question is answered in Chapter 3.

As visual representation improves understandability [5], the next research question is: "How can visual representation help AAOM create and advantage in a software development process over alternative methods?". As there are many different roles in the software development process (customer, project lead, developer, tester, analyst, etc.) it is necessary that modelling suits their needs. Because all these roles have different backgrounds, a simple and understandable representation of goals must be used. The author of this thesis claims that visual representation is intelligible for developers with a technical background as well as for customers with no technical background as the primary goal of data visualization is to communicate information clearly and efficiently to the users [5]. This research question is answered in Chapter 4.

As tools are needed for modelling, the next questions addresses tooling. "How does tooling affect AAOM and help it to create an advantage in a software development process over alternative methods?". Tools are needed to visualize goal models and also to update and change them. However, using new software implies a learning curve that results in overhead for the software engineering process. Additionally, tools are expensive resulting in unnecessarily high cost of developed software. These two shortcomings are mitigated by using tools with better usability [8] and using community supported freeware [9] [10]. In current thesis investigation for advantages of tooling and possible tools is performed. This research question is answered in Chapter 5.

1.6 Context

To address the shortcomings described in Chapter 1.4, AAOM for RE is experimentally used in a consumer factoring (from here on COF) project development [11]. Factoring is a finance transaction where a business sells its account receivables to the factor so the business can get money in front. In other words, customer pays for the product in installments to the factor (in this case a financial-industry establishment) and factor pays the whole amount for the product to the business. The process of offering consumer factoring to clients consists of receiving a financing application, scoring customers' creditworthiness and signing contracts. The

development team consists of 4 members: an analyst, two software engineers and a quality assurance engineer. The customer is from the same company but from a different department. The same team had previously developed a COF system for private customers but the need of offering financing to business clients results in a need for additional development. During this research, a goal model is created collaboratively during a meeting with customer, analyst, author and supervisor of this thesis.

Agile software development methodologies such as Kanban [12], Lean [13] and Test-driven development (TDD) [14] are used in the project. Kanban is a method for managing development process with an emphasis on just-in-time delivery [15]. Development tasks are visualized on a white-board. Each participant in software development can have a pre-defined amount of work in progress (WIP) [12]. Lean software development is a collection of agile principles for more effective software development [13]. Lean principles consist of eliminating waste, amplifying learning, deciding as late as possible, delivering as fast as possible, empowering the team, building quality in and seeing the big picture [13]. TDD is a software development process where tests are written before the production code itself [14]. When tests are written and failing (as there is no production code yet), then the real production code is developed so that tests pass [14]. Thus, the combination of Kanban (for managing development process), Lean (for making the process more effective thanks to following Lean principles) and TDD (software development process) is used in the COF project. The working software is released in 3 weeks cycles and the project is finished in 3 iterations (9 weeks). COF for private customers was previously developed using different RE method. Requirements were gathered from the customer in free text in a text document. They were analyzed and more detailed specification was written. Development tasks were then created and linked to specification text. However, the team sees the need for more effective communication and understanding between team members and therefore, AAOM is experimentally used for additional development.

1.7 Structure

As case study research method is used for this thesis, linear-analytic structure is used as proposed in [16]. It suggests a structure for the correct way of reporting case study results. However it is adjusted to better suit thesis' mandatory structure.

The thesis is divided into introduction in Chapter 1, research background in Chapter 2, answers to the research questions in Chapters 3, 4, and 5 and conclusion in Chapter 6. Chapter 1 consists of introduction to RE in Chapter 1.1, AAOM in Chapter 1.2 and RE in agile in Chapter 1.3. As using AAOM in a software development process is the topic of this thesis, then description of them is needed. Thesis also consists of problem statement (gap detection) in Chapter 1.4 and the research questions in Chapter 1.5. Problem statement highlights shortcomings in current processes and research questions are formulated based on them.

Next, the research background in Chapter 2 is introduced. Firstly, earlier studies in Chapter 2.1 are described and then the theory of AAOM in Chapter 2.2 is explained. Also, case study method and implementation is more thoroughly described. Case study Chapter 2.3 consists of method introduction in Chapter 2.3.1. In Chapter 2.3.2 case and subject selection are described. It is explained why this specific project is chosen and who are the subjects participating in case project development. Then data collection procedure in Chapter 2.3.3 is described. It explains how and what kind of data is collected. Finally, in Chapter 2.3.4 data analysis procedure using composed formula is explained.

Then all 3 research questions in Chapters 3, 4 and 5 are answered. All the research questions describe the question itself and how they are split up into more specific sub-questions. Then each sub-question is analyzed in the "Interview nodes analysis" chapter and then given summaries in the "Analysis summary" chapter. The first research questions' answer is in Chapter 3 and it addresses how to create better understandability and communication with AAOM. The second research question is answered in Chapter 4 and it addresses how visual representation of goal models help AAOM to achieve its goals. The last research question addressing tooling in AAOM is in Chapter 5. All previously mentioned chapters answering main research questions are connected to each other by conclusion and introduction.

After the research questions are given their answers, this thesis is concluded in Chapter 6. Conclusion consists of summary of findings in Chapter 6.1, relation to existing evidence in Chapter 6.2, future expectations in Chapter 6.3 and limitations and future work in Chapter 6.4. In the summary of findings, answers to main- and sub-questions are summarized. Relation to existing evidence chapter compares findings of this work with already existing evidence. Future expectations then describes research subjects' expectations for AAOM and limitations and future work chapter describes work needed to be done in the future based on research results.

2 Research Background

2.1 Earlier studies

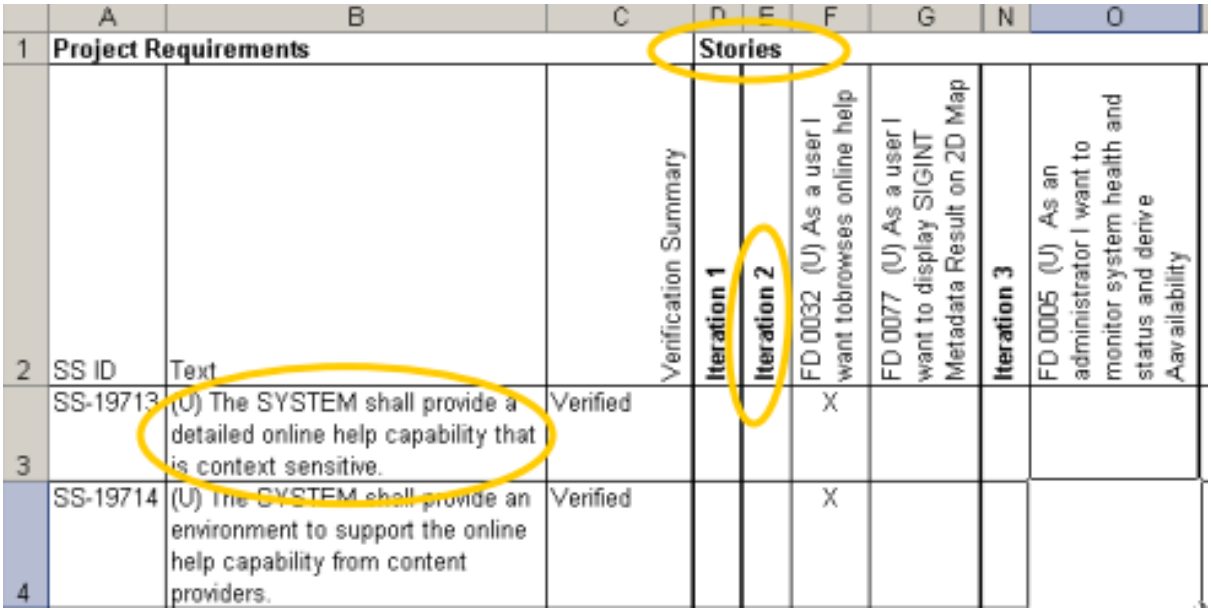
A lot of research work exists about goal based RE. AAOM combines different aspects of them into a new methodology. This chapter describes and introduces earlier studies on agile goal modelling.

An article [17] on Goal-oriented RE suggests goals classification based on their type, for example performance, accuracy and security goals. To be clear, these types of goals are sub-types of functional and non-functional goals, which are also used in AAOM (functional goals and quality goals respectively).

[17] also proposes that goals can be soft and hard goals. The result of completing soft goals can't be measured clearly. The result of hard goals' however can. A good way to specify the difference of links between goals and roles is to give them types. Thanks to different types of links it is possible to express which goals have to be carried out to complete higher goals and which are optional. Suggested link types are "AND" and "OR" links where "AND" link describes the need to complete sub-goals and OR link describes the possibility to complete some (not all) sub-goals.

A good way to extract a goal hierarchy from customer is to ask WHY and HOW questions [17]. Asking WHY questions about a goal reveals higher lever goals and asking HOW questions reveals lower level goals. This approach is also used in the industry case study.

Another source [18] recommends the same structure for requirements and user stories in an agile software development but does not offer any visual aid for created links a hierarchy of requirements and user stories (see Figure 2 Requirements representation).



	A	B	C	D	E	F	G	N	O
1	**Project Requirements**			**Stories**					
2	SS ID	Text	Verification Summary	**Iteration 1**	**Iteration 2**	FD 0032 (U) As a user I want tobrowses online help	FD 0077 (U) As a user I want to display SIGINT Metadata Result on 2D Map	**Iteration 3**	FD 0005 (U) As an administrator I want to monitor system health and status and derive Aavailability
3	SS-19713	(U) The SYSTEM shall provide a detailed online help capability that is context sensitive.	Verified			X			
4	SS-19714	(U) The SYSTEM shall provide an environment to support the online help capability from content providers.	Verified			X			

Figure 2 Requirements representation [18]

[18] also recommends that user stories are created in such detail that they need to be split up into tasks, which is different from AAOM. However, the main principles are the same – requirements and user stories can change after each iteration, requirement analysis is carried out in the release planning phase, high level requirements can be mapped to user stories and they are either completed or not and therefore affecting the status of higher level requirements (partial, completed, etc.).

In general the main point of all studies is the same: goals are hierarchically linked and therefore structured, models change in iterations, user stories are used as the most specific layer and goal models are created in the analysis phase. Differences are whether goals are visually represented or not. Also if links are given types and goals are classified. In AAOM, visual representation is used, links do not have types and goals are classified as functional and non-functional.

2.2 Theory of AAOM

AAOM consist of user stories, roles, links, quality goals and functional goals. This chapter explains different artifacts of goal model.

A user story is an artifact in agile software development and is a very high-level definition of a requirement. It should in theory contain just enough information so that the developers produce

estimates of implementing it [19]. User stories are written in a way that they describe what and why a user in a specific role needs to do. Mike Cohn suggests using the format: “As a (role) I want (something) so that (benefit)” [20]. For example: ”A client manager has to be able to receive financing applications so he could score them”. In the example, a client manager is the role, receiving financing applications is the activity and being able to score the applications is the benefit. As user stories are simple enough to understand and write, it is recommended that stakeholders write them, not developers and that is what AAOM RE is trying to promote.

In AAOM RE, user stories are a part of goal models and are interpreted as the lowest level goals. User stories need to be large enough for development task. In agile software development, a good practice is to keep tasks small enough to be developed in 1-3 days (in the industry case a maximum length of 3 days per task is used) [21] [22]. This in turn means that user stories must be detailed enough to be developed in 1-3 days. As mentioned, user stories are the lowest level in goal model and therefore they are sub-goals of their parent goals. A parent goal cannot be carried out before all the sub-goals are completed. This applies for the whole model up until the top-most level goals which are the most general. If the very highest level goal is completed, i.e. also meaning all of its sub-goals are completed, then the whole project is completed too.

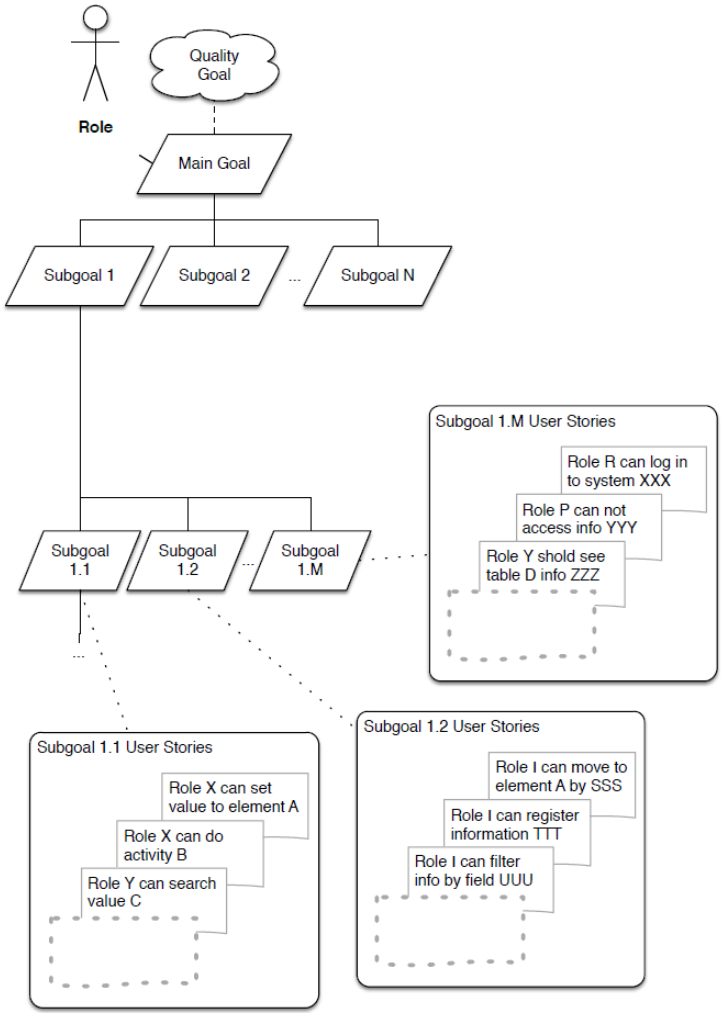


Figure 3 Example goal model [3]

Quality goals in AAOM (clouds in Figure 3 Example goal model) represent soft goals or non-functional requirements. Non-functional requirements [23] specify a criteria by what it is possible to judge the operation of a program. This is in contrast with functional goals which specify behaviors and functions of system.

Roles are capacities or positions that are needed to achieve goals [3]. They help to realize constraints of different roles in system in early stages of analysis. If a role needs to achieve a goal then it also needs to achieve all of its sub-goals.

The notation for representing a goal model is simple. It consists of 5 different symbols as seen in Figure 4 Goal Model notations .

Symbol	Meaning
	Goal
	Quality goal
	Role
	Relationship between goals
	Relationship between goals and quality goals

Figure 4 Goal Model notations [3]

Functional goals are represented as parallelograms and non-functional goals (quality goals) as clouds. Roles are represented as stickmen. Goals and roles are connected with a continuous line and quality goals are connected with dashed line.

2.3 Case study

2.3.1 Description of Case study method

The case study method is used in this thesis. A case study in software development is an empirical inquiry that draws on multiple sources of evidence to investigate instance(s) of a contemporary software engineering phenomenon within its real-life context, especially when the boundary between phenomenon and context cannot be clearly specified [16]. In industry case multiple sources of evidence are used (see chapter 2.3.3) to find out if AAOM is better in real-life context (Consumer Financing system).

There are criticisms about the case study method that it is not accurate enough and it is impossible to generalize based on it. However using the correct research methodology practices it is possible. It is also important to reconsider received knowledge for generalization [16] [24].

Case studies consist of different stages including case study design, data collection, analyzing data and reporting. This thesis is composed keeping these stages in mind and using reporting structure suggested in [16].

2.3.2 Case And subject selection

The case of this thesis is software development project (Consumer financing) with a need to gather requirements. It is used to study how agent-oriented requirements engineering method works in real life context. This case is chosen mainly because it is a starting project. This makes it possible to monitor the whole software development process from the beginning to the end. Another reason this case is chosen is the fact that case study subjects participated in a similar project before starting this one, so they had experience developing a project with other modelling methodologies mentioned in Context 1.6. As requirements engineering is usually done in the beginning of system development lifecycle [4], this project suits perfectly. It is possible to collaboratively create a goal model and observe the completion of the system while doing interviews with case subjects.

Subjects chosen for this case are customer, analyst and developers. They are chosen because they are most closely related to completing the project. All participate in modelling process, requirements engineering, development and collaborative meetings. Also data is collected from them. Subjects have different experience in software development process: analyst is

experienced, customer has average experience and both developers have also average experience.

2.3.3 Data collection procedure

Primary data collected are interviews. Interviews are conducted in 3 rounds. First round of interviews are done right after model completion but just before implementation. Second round of interviews is done after finishing first iteration (3 weeks). By then customer has received first part of ready product and developers have got used to developing user stories based on goal model. Third and last round of interviews is conducted right after finishing the product in three iterations (9 weeks).

Each round of interviews is conducted with all team members (analyst and developers) and customer. Interviews are hour-glass like, more generic questions at the beginning and in the end and more detailed questions in the middle. Interviews are recorded and no names are used. Instead, role names are used. Interviews are then transcribed and coded in NVIVO [25].

LOC is also gathered. Thanks to the use of version control system, it is possible to get the exact state and amount of LOC at any given moment. As code commits are connected to goals through JIRA tasks, it is also possible to get the exact state of code by goals.

History of model changes is also collected. First version of model is created on blackboard with analyst, customer and author and supervisor of this thesis. Second version of goal model, based on first version on blackboard is saved in the project task management tool JIRA which makes collecting data of model changes easy. Tasks in JIRA represent goals in goal model and are linked to each other through linking mechanism provided by the tool. When goals change, a team member changes corresponding JIRA tasks. Tool provides also rudimentary functionality for visual representation of created goal structure via tasks and links.

2.3.4 Analysis procedure

As interviews are the most important data gathered, they are analysed most thoroughly. Interviews are conducted in three rounds. All of them are recorded and have an hourglass like structure, where generic questions about research are asked first, then more detailed questions and finally again more general questions. Each interview consists of 4 - 15 questions depending

on round and subject. First round of interviews is conducted right after the modelling process and before the start of development. As there are 4 subjects in this case study, first round consists of 4 interviews. 2 interviews are conducted with developers, 1 with analyst and 1 with customer. Because goal model affects the work of analyst most, second round is done only with analyst. This interview is conducted right after the first iteration, 3 weeks from the start of project. Third round of interviews is conducted at the end of project, 3 iterations and 9 weeks after creating the goal model. 5 interviews are written and 4 are conducted orally. Interviews conducted orally need to be transcribed. After transcription written interviews are coded. Interview analysis begin with axial and open coding [26]. Some categories are based on research questions, others are identified during coding. Groups of data are identified as nodes. 57 of 100 created nodes are actually used in analysis. They are then grouped in 11 themes. A theme is an outcome of coding, categorization, and analytic reflection, not something that is, in itself, coded [27]. They consist of:

- Benefits (12 nodes)
- Modelling suitability (6 nodes)
- Method clarification (10 nodes)
- Collaborative modelling (5 nodes)
- Expectations (7 nodes)
- Visual representation (1 node)
- Tools usage (4 nodes)
- Time taken for modelling activities (4 nodes)
- New ideas (2 nodes)
- Drawbacks (3 nodes)
- Method comparison (2 nodes)

There are 11 different groups of nodes and each of them can exist in only one group. Nodes are given weights as described in Chapter 3.5. This helps better understand the proportion and relevance of nodes and themes.

Polarity and type are also added to nodes. Polarities can be “Positive”, “Neutral” and “Negative”. Types are “Statement” and “Suggestion”. Referenced by shows short representation of subjects (Chapter 2.3.2). C is customer, A is analyst and D is developer.

$$sources \times references + experience = score$$

Previous algorithm is used to compare nodes. It is devised by the author and supervisor of this thesis. Sources are the number of interviews consisting specific node. For example if 5 interviews of 9 mention the same node, then sources of it is 5. References is the count a node is mentioned over all interviews. For example if it is used 3 times in one interview and 2 times in other, then references of that node is 5 and sources 2. As each subject of study have different experience in field, it seems correct to give them different scales. 3 subjects of 4 have average experience and one is experienced. Average experience gives a score of 2 per each source and experienced gives a score of 3 per each source. So if experienced and with average experience subjects mention the same node once or more times in one interview (source), then the experience of that node is 3 + 2 = 5. For example if a node is mentioned in 8 sources where 3 of them are the interviews of experienced subject and 5 are the interviews with average experience subjects and node is mentioned 10 times all together, then the final score of it is:

$$8\ (sources) \times 10\ (references) + (3 \times 3 + 5 \times 2) = 8 \times 10 + 19 = \mathbf{99}$$

Maximum number of sources for one node is 8 and maximum number of references is 10 (mentioned 10 times in 8 interviews). Least mentioned interviews have only 1 reference in 1 source by a subject with average experience, which means it has a minimum score of 3. Lowest score for a theme is 10.

2.4 Conclusion

In this chapter RE is reviewed and AAOM is fitted for RE as method of our choice. To evaluate the suitability of AAOM, the Case study research method is used. A scientific framework is set up using this approach. Case study phases are planning, data collection and analysis. For any of them scientific methods can be applied. For planning phase case study protocol is used, for data collection interviews are conducted and for data analyses coding is performed. To report and conceptualize analysis a formula is devised.

3 Understandability and communication

3.1 Introduction

This chapter answers the research question “How to achieve a better understandability and mutual communication with AAOM to create an advantage in a software development process over alternative methods?”. It is divided into more specific sub-questions “What benefits do better understandability create?”, “What benefits do collaborative modelling create?” and “What amount of time is spent on modelling activities?”. Interviews are conducted to find answers to these questions. Nodes connected to these sub-questions are grouped in themes, which consist of “Benefits”, “Collaborative modelling” and “Time taken for modelling activities”. Each theme addresses one sub-question. Nodes are given scores and they are analyzed in the next chapters.

3.2 Benefits

The first sub-question is “What benefits do better understandability create?” and it is about different benefits created by better understandability in AAOM. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 3.2.2.

3.2.1 Interview nodes analysis

The theme “Benefits” addresses this research question and consists of the next nodes (ordered by the count of references).

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Secure feeling for project direction	8	10	Positive	Statement	c.a.d.d.
Extracting information from customer better	3	4	Positive	Statement	a.

Understood the value delivered	2	4	Positive	Statement	d.d.
Mutual communication	3	3	Positive	Statement	a.
System was developed according to goal model	3	3	Positive	Statement	c.d.d.
Estimate work ahead	2	2	Positive	Statement	d.d.
User story size good for development task	2	2	Positive	Statement	a.d.
Helps to clarify existing documentation and system	2	2	Positive	Statement	a.d.
Constructive modelling	1	1	Positive	Statement	c.
Intuitively understandable	1	1	Positive	Statement	a.
Roles help better understand the customer	1	1	Positive	Statement	a.
Goal model shows positive results	1	1	Positive	Statement	a.
User story too big for development task	1	1	Negative	Statement	d.

Table 1: Nodes for theme "Benefits"

All the nodes in Table 1: Nodes for theme "Benefits" are positive statements supporting hypothesis. It consists of the number of sources and references for each node, node's polarity and type and also which roles are referring it. The biggest benefit according to nodes is the fact that team members have a secure feeling for project direction and goal model helps them to see the big picture of system. Each subject mentions this for at least once during the interviews. Analyst claims in each interview that extracting information from customer is better than before and also goal models help with the mutual communication. Developers say that goal model helps them to better understand the value delivered, which also helps them to create software without the need to ask as many questions as before. It additionally gives an opportunity to estimate work ahead. Almost all subjects think that system is developed according to created model, meaning the overhead of miscommunication is significantly lower (as model was created collaboratively).

3.2.2 Analysis summary

The biggest benefit found according to case study's participants is "Secure feeling for project direction". It is safe to say that using AAOM helps subjects of development lifecycle to see the big picture of project and feel secure for its direction. All research subjects mention it, 3 out of 4 even more than once. Two other benefits noted are "Extracting information from customer better" and "Mutual communication". According to the case study participants' claims AAOM also helps to understand the value delivered and also almost all subjects say that system is developed according to model. One participant claims user story to be oversized for development task. Since it is referred only once by a developer with average experience, additional training in agile may be needed. Further investigation on this topic is needed. Nodes with lower scores are "Goal model shows positive results", "Constructive modelling" and "Intuitively understandable". Although all of these benefits are mentioned by the team, the number of references is too low to conclude anything from them. The reason for this may be the fact that time from modelling process to finishing the project was too short to feel any noticeable results from models and as team was relatively experienced, understandability was explicit there and not worth mentioning.

3.3 Collaborative modelling

This chapter is for the sub-question "What benefits do collaborative modelling create?" and it is about benefits created by collaborative modelling in AAOM. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 3.3.2.

3.3.1 Interview nodes analysis

Theme addressing this question is "Collaborative modelling". It consists of the following nodes (ordered by the number of references).

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Improving understandability	2	4	Positive	Statement	c.a.

Involving participants	3	4	Positive	Statement	c.a.
Having everyone on the same page	2	2	Positive	Statement	a.
Few feelings about collaboration	1	1	Neutral	Statement	c.
Pinpointing problems	1	1	Positive	Statement	a.

Table 2: Nodes for theme "Collaborative modelling"

These nodes in Table 2: Nodes for theme "Collaborative modelling" focus on the collaborative modelling and benefits created by it. The table consists of the count of sources and references, node's polarity and type and also which roles reference it. Nodes in this table are only referenced by the customer and the analyst. Both of them think that collaborative modelling improves understandability and involves participants. Analyst also say that it helps to keep everyone on the same page and also to pinpoint problems. Customer however mentions once that she has few feelings about collaboration.

3.3.2 Analysis summary

The theme "Collaborative modelling" focuses on what subjects think about modelling and what benefits it creates. These nodes directly point out benefits answering this research question. Biggest benefit from collaborative modelling in this theme is "Involving participants" and this is mentioned by the analyst in 2 interviews and by the customer in 1 interview. Involving participants greatly improves mutual communication. "Improving understandability" is another benefit brought out by analyst and customer in 1 interview both 2 times. These benefits answer this research question. A node with lowest score in this theme is "Few feelings about collaboration" which is mentioned once by customer. As its score is so low, it is not possible to really conclude anything from that.

3.4 Amount of time spent on modelling activities

This chapter is for the sub-question "What amount of time is spent on modelling activities?" and it is about time usage in AAOM. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 3.4.2.

3.4.1 Interview nodes analysis

The theme "Time taken for modelling activities" directly addresses this sub-question and contains the following nodes (ordered by the number of references).

Node	Sources	References	Polarity	Type	Ref. by
Time used effectively	2	2	Positive	Statement	c.a.
From idea to user story fast	2	2	Positive	Statement	c.a.
Before AOM took lot of time	1	1	Positive	Statement	a.
Refining goal models fast	1	1	Positive	Statement	a.
Refining goal model slow	1	1	Negative	Statement	a.
Initial user stories take time	1	1	Negative	Statement	a.

Table 3: Nodes for theme "Time taken for modelling activities"

This Table 3: Nodes for theme "Time taken for modelling activities" consists of 5 nodes. The table consists of the number of sources and references, polarity and type and also which roles reference the node. As the customer and the analyst are more involved in modelling activities, only they reference nodes in this theme. Both think that time is used effectively and from idea to user story is fast. The analyst also say that modelling activities took more time before.

3.4.2 Analysis summary

Two nodes with the highest score are "Time used effectively" and "From idea to user story fast". Both are referenced by the analyst and the customer once, meaning it is possible to say that AAOM does not create overhead. On the contrary, subjects claim that time is used effectively and in AAOM it is possible to go from idea to user story fast. However, initial user stories can take more time as this method is new to participants. Another nodes in this theme are "Before using AAOM, modelling took lot of time", "Refining goal model fast", "Initial user stories take time" and "Refining goal model slow". All of them are mentioned once by one case

study participant and this greatly shows, that it is not possible to conclude from one reference, as two of mentioned nodes are contradictory.

3.5 Conclusion

As all the sub-questions find positive answers, it is possible to conclude that better understandability and mutual communication are benefits created by using AAOM. Additionally, it can be concluded that using AAOM reduces time taken for modelling activities. Created benefits include project members' secure feeling for the project direction, understanding the value delivered, mutual communication, more precise extracting information from customer and improving understandability thanks to collaborative modelling. The time for modelling and creating visual representation was used effectively and time spent from idea to user story was reduced. However, as found out from interviews, understandability is better largely thanks to visual representation, which is formulated into next research question.

4 Visual representation

4.1 Introduction

Visual representation is one of the most important parts of AAOM. This is why a research question "How to create visual representation with AAOM for achieving an advantage in a software development process over alternative methods?" is dedicated to evaluate this. It is divided into more specific sub-questions "Where does visual representation fit in?", "Who finds visual AAOM suitable?" and "What practices and activities of goal modelling are clear and what needs clarification?". Interview questions are composed to find answers to these questions. Themes addressing them are formed from nodes. These themes consist of "Visual representation", "Modelling suitability" and "Method clarification". Nodes are given scores and they are analyzed in the next chapter.

4.2 Visual representation suitability

The first sub-question is "Where does visual representation fit in?" and it is about visual representation suitability in software development process. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 4.2.2.

4.2.1 Interview nodes analysis

The theme addressing this question is "Visual representation" and it consists of one node.

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Visual representation is suitable in general	5	10	Positive	Statement	c.a.

Table 4: Nodes for theme "Visual representation"

This node in Table 4: Nodes for theme "Visual representation" is referenced by the customer and the analyst 10 times in 5 interviews, which is impressive. The table consists of the number of sources and references, polarity and type and also which roles reference the node. Another

reason this node has such a high score is the fact that the experienced analyst mentions this in all 3 interviews, which means thanks to his experience the score is higher.

4.2.2 Analysis summary

The theme "Visual representation" consists of only one node "Visual representation – suitable in general". This node is really strong as it is referenced 10 times in 5 interviews by both customer and analyst. Based on the high score of it, it is possible to answer this research question with the following answer: visual representation generally fits in development process.

4.3 Modelling suitability

The second sub-question is "Who finds visual AAOM suitable?" and it is about visual modelling suitability for participants in software development process. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 4.3.2.

4.3.1 Interview nodes analysis

The theme addressing this research question is "Modelling suitability" and it consists of next nodes (ordered by the number of references):

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Model didn't change	3	8	Neutral	Statement	c.a.
Modelling more for analyst and client, not dev	6	6	Negative	Statement	c.a.d.d.
Would use goal models in other projects	4	4	Positive	Statement	c.a.d.d.
Clarifies what needs to be done	2	3	Positive	Statement	a.
Didn't watch model much during development	2	2	Neutral	Statement	c.
Modelling more for analyst, not client	1	1	Neutral	Statement	c.
Analyst has the most responsibility	1	1	Negative	Statement	c.

Table 5: Nodes for theme "Modelling suitability"

These nodes in Table 5: Nodes for theme “Modelling suitability” focus on how modelling suits into the development process. The table consists of the number of references and sources, node’s polarity and type and also which roles reference it. All participants of project would use goal models in other projects, which is really satisfying. Case study participants also think that modelling is more for the analyst and the client, not the developers. However, the customer specifies that modelling is more for the analyst and not the customer.

4.3.2 Analysis summary

In the “Modelling suitability” theme the node with highest score is “Modelling more for analyst and client, not developer” thanks to the developers, who both mention this in 2 interviews. This lets us know that in this project and team setup developers do not feel the need to participate in modelling process. It is interesting however that both the analyst and the customer also mention this in 1 interview. It is also brought out by the customer that the analyst has the most responsibility. This is a negative statement as too much work should not fall on a single team member’s shoulders. The node with the most references (but lower score) is “Model didn’t change” and it is mentioned by the analyst and the customer. It seems that model is initially created so well and business requirements do not change, so there is no need to change or restructure it. Really positive is that the node “Would use goal models in other projects” also has a high score and all participants mention this once meaning all the team feel this way. As the subjects are quite experienced (see 2.3.2), it can be inferred that goal model suits all of them. However, the customer and the developer both reference the node “Didn’t watch model much during development” once. It is rather negative than positive, but the reason might be the fact, that model didn’t change and all subjects just remembered it well enough and did not need to watch it much during development. When all team members think that modelling is more for the analyst and the client, not the developer, then the customer additionally says that it is more for the analyst, not the client. This means that all the team members think that modelling should be a job for the analyst and the customer, but the customer herself rather prefers that it should stay on the shoulders of the analyst. The reason for this could be the fact that the analyst is the most experienced of them all and the customer is just slightly afraid to take responsibility. In this case, it depends on the project setup, who should focus more on modelling and who should not.

4.4 Method clarification

The third sub-question is "What practices and activities of goal modelling are clear and what needs additional clarification?". Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 4.4.2.

4.4.1 Interview nodes analysis

This question is addressed by the theme "Method clarification" and it consists of the following nodes (ordered by the number of references).

Node	Sources	References	Polarity	Type	Ref. by
Goal model clear	4	5	Positive	Statement	c.a.d.
User story concept clear	4	5	Positive	Statement	c.a.d.
From goals to user stories clear	3	4	Positive	Statement	c.a.d.
Usage of quality goals clear	2	3	Positive	Statement	c.a.
Usage of quality goals unclear	2	2	Negative	Statement	d.d.
From goals to user stories unclear	1	1	Negative	Statement	d.
Usage of roles clear	1	1	Positive	Statement	c.
User story concept unclear	1	1	Negative	Statement	d.
Quality goals link to user stories unclear	1	1	Negative	Statement	d.
Didn't feel the need for quality goals	1	1	Neutral	Statement	a.

Table 6: Nodes for theme "Method clarification"

These nodes in Table 6: Nodes for theme "Method clarification" focus on visual clarification of goal model. The table consists of the number of references and sources, node's polarity and type and also which roles reference it. All the subjects except for one developer find goal model and user stories clear. They also find the sequence form goals to user stories clear. Quality goals are clear for only the customer and the analyst and unclear for the developers. The customer understands the use of roles the best.

4.4.2 Analysis summary

Nodes with highest scores in this theme indicate that creating goal models and user stories are clear for the team. This means that participants are included in modelling process and they understand it. The mechanism of goals to user stories is clear to all team members except for one developer. This can possibly mean that the experience of that one developer is lower than other team members'. An interesting fact to bring out here is that the customer and the analyst both say that usage of quality goals is clear, as where the both developers claim the usage of quality goals unclear. This suits with the highest score node "Modelling more for analyst and client, not developer" in theme "Modelling suitability". It is clear, that both developers think that modelling process should be in analysis phase and should not consider them much. However the analyst says that he does not feel the need for quality goals. That may be because they have not used non-functional requirements before and therefore do not feel the shortage of them.

4.5 Conclusion

It can be concluded from all previous nodes that visual representation is suitable for all roles and it generally fits in software development process. Visual representation of goal model is clear for the team. Also the fact that the customer understands different parts of goal model and therefore the model itself means that goal model helps non-technical people to better understand the system developed. Although the whole team considers visual AAOM suitable, we infer that participants rather prefer that modelling is done by the analyst and the customer. As manual goal model drawing can create overhead due to model changes and it is difficult to manage different versions of drawn models, a tool is needed. The research question about tooling is in next chapter.

5 Tooling

5.1 Introduction

As visual representation is one of the most important parts of AAOM, a visualization tool is needed. A research question "How to use tooling with AAOM and to create an advantage in a software development process over alternative methods?" is dedicated for tooling and how does it influence modelling. It is divided into two more specific sub-questions: "What are tooling shortcomings for AAOM?" and "What kind of method can be compared to AAOM?". These sub-questions are addressed by two themes of nodes: "Tools usage" and "Method comparison". Nodes are given scores and they are analyzed in the next chapter.

5.2 Tools usage

The first sub-question is "What are tooling shortcomings for AAOM?". It addresses different shortcomings of AAOM. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 5.2.2.

5.2.1 Interview nodes analysis

The theme answering this research question is "Tools usage" and it consists of the following nodes (ordered by the number of references):

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Dedicated tool support needed	5	6	Negative	Suggestion	c.a.d.d.
Need more integration	1	1	Negative	Suggestion	a.
Commercial better	1	1	Negative	Statement	a.
Good enough for starters	1	1	Neutral	Statement	c.

Table 7: Nodes for theme "Tools usage"

These nodes in Table 7: Nodes for theme “Tools usage” address shortcomings created by tools usage. The table consists of the number of sources and references, node’s polarity and type and which roles reference it. All subjects feel the need for dedicated tool support, which is understandable. The analyst thinks that commercial tools are better and using tools creates the need for more integration. The customer however says that used tools are good enough for starters.

5.2.2 Analysis summary

This theme directly addresses this research question. It contains a node called “Dedicated tool support is needed”. This is mentioned by all subjects and it is possible to conclude, that as this method is relatively new, it lacks dedicated tools support and really needs it. This is also the main shortcoming of tooling in AAOM. Another nodes are mentioned only once and consisted of “Need more integration”, “Commercial better” and “Good enough for starters”. It is possible to conclude that the biggest shortcoming of AAOM is the lack of dedicated tool support.

5.3 Method comparison

The second sub-question is “What kind of method can be compared to AAOM?” and it addresses method comparison to alternatives. Nodes from interview analysis are described next. Additionally, analysis is summarized in Chapter 5.3.2.

5.3.1 Interview nodes analysis

The theme answering this research question is “Method comparison” and it consists of two nodes (ordered by the number of references):

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Better than making notes	2	2	Positive	Statement	c.a.
Similar to what was done earlier	1	1	Neutral	Statement	c.

Table 8: Nodes for theme "Method comparison"

Nodes in Table 8: Nodes for theme "Method comparison" address how participants compare AAOM to other methods. The table consists of the number of sources and references, node's polarity and type and which roles reference it. Both, the customer and the analyst say that method is better than making notes. The customer also thinks that method is similar to what was done earlier.

5.3.2 Analysis summary

Previous interview nodes analysis suggests weakly that AAOM suits into larger projects and also smaller ones, where making notes is more common. The second node is only referenced once and is called "Similar to what was done earlier". But in general comparison to other methods (except for making notes) based on interviews cannot be made, due to little information gathered.

5.4 Conclusion

In conclusion, dedicated tool support is definitely needed and creation of it is mentioned in Chapter 6.4. This is expected as there is no good way to connect visual representation of goal model with the real development task. Another reason for the need of tool support is that goal model should update itself automatically based on development tasks updates so there is no need for duplicate work. However, due to the lack of information received from the case study participants, it is not possible to make conclusions about comparison to other methods. This needs additional work that does not fit into this thesis' scope.

6 Conclusions

6.1 Summary of findings

In this thesis a case study research is carried out to find out how AAOM creates an advantage over alternative methods. Case and case subjects are chosen and data is collected in the form of interviews. Conducted interviews are transcribed, coded and then analyzed. Acquired findings are reported in this thesis.

The first research question, "How to achieve a better understandability and mutual communication with AAOM to create an advantage in a software development process over alternative methods?", finds the following answer. AAOM creates an advantage in a software development process thanks to better understanding, collaborative modelling and also more effective time using in modelling activities. Benefits of better understanding contain participants' secure feeling for project direction, better information extracting from customer and understanding the value delivered. Additional benefits from collaborative modelling include improved understandability, participants' involvement and also having everyone on the same page. More effective time usage in AAOM also answers this research question. All these benefits from better understandability, mutual communication and more effective time usage help AAOM to create advantage over alternatives.

The second research question, "How to create visual representation with AAOM for achieving an advantage in a software development process over alternative methods?" is answered next. Research concludes that visual representation of goal model suits in general and that it can be used for larger as well as for smaller projects. Another conclusion made is that visual AAOM is suitable to all software development process participants, however in this case project setup team preferred that modelling is done by analyst and customer. Modelling process suits participants because it clarifies what needs to be done. Based on the interview analysis it can be concluded that visual representation of AAOM is clear to all participants, with one exception, explained further on. Additionally, the modelling makes the creation of user stories from goals understandable. We can conclude that using goal models with User Stories in order to create competitive visual representation for RE is suitable.

The third research question, "How to use tooling with AAOM and to create an advantage in a software development process over alternative methods?" is answered as follows. Because AAOM draws a lot on visualization, proper tools are needed to simplify creating and updating

goal models. From interview analysis it is concluded that dedicated tool support is strongly needed, since currently there is none. This thesis did not give a clear answer about comparison with competitive methods. Only remark was that AAOM is better than making notes. Researching AAOM positioning among other methods needs however additional work that is out of this thesis' scope.

Finally, an answer to the main research question can be formulated. AAOM creates an advantage in software development process over alternative methods through better understandability and mutual communication, more effective time usage, generally suitable and clear visual representation of AAOM. Additional value can be added with proper tooling.

To get additional confirmation for previously described results, answers to this research questions are compared to existing evidence in next chapter.

6.2 Relation to existing evidence

Another thesis [28] is written on similar subject but a different system is developed. That work concludes that AAOM suits software development. Visual approach helps better understand the connection between different goals and user stories. In other words it helps to understand the big picture, which corresponds with the result of this thesis. The strongest node in this interview analysis is "Secure feeling for project direction" which suits with the other work. Both thesis also conclude that AAOM encourages collaboration and therefore helps with mutual communication. AAOM also helps team members to participate more in software analysis phase.

In general both thesis have similar research questions. The study [28] has research question "How suitable is AAOM for RE in agile software development?" and as mentioned in Chapter 1.5 the main research question of this thesis is similar. Both works successfully answer these similar questions meaning there is a strong relation to existing evidence.

6.3 Future expectations

During interviews some future expectations are brought out that do not answer any research questions. They are grouped in theme "Expectations" and consist of the following.

Node	Sources	References	Polarity	Type	Ref. by
Common expectations between stakeholders	3	3	Positive	Statement	c.a.d.
Less unexpected changes	2	2	Positive	Statement	d.d.
Customer ability to adapt with already delivered product	2	2	Positive	Statement	c.a.
Can use existing model for future developments	1	1	Positive	Statement	c.
Time estimation correct	1	1	Positive	Statement	d.
Use agile methodologies	1	1	Neutral	Statement	d.
Get customer feedback	1	1	Positive	Statement	d.

Table 9: Nodes for theme "Expectations"

These 6 nodes in Table 9: Nodes for theme "Expectations" each represent participants' expectations to goal model. The table consists of the number of sources and references, polarity and type and also which roles reference the node. Strongest expectation in this theme is "Common expectations between stakeholders" and it is referenced by all subjects except one developer. Another strong nodes are "Customer ability to adapt with already delivered product" and "Less unexpected changes". Both developers reference "Less unexpected changes" as they are the ones who need to implement them. Customer and analyst however expect customer's ability to adapt with already delivered product. This should help customer to better understand the system. Other future expectations in this theme are "Can use existing model for future developments", "Time estimation correct", "Use agile methodologies" and "Get customer feedback". As they are all referenced only once, it is not possible to conclude anything relatively important based on them.

6.4 Limitations and future work

As mentioned in chapter 6.1 the answers to research questions are positive. However there are some drawbacks highlighted by the subjects. For example "Analyst has the most responsibility". It is important that too much weight would not fall on one persons' shoulders as that would potentially create more errors. A solution for this problem is always to involve the whole team in modelling process, which is also done in financial-industry case. A strong team lead should try to make sure that everyone collaborates in modelling process and no one feels being the most responsible. It is understandable that not all team members have the same experience and therefore some might have difficulties understanding parts of AAOM. And that brings us to the future work.

Future work consists of creating instructions for team members about AAOM. For example some might have problems understanding quality goals, functional goals or user stories. Mike Cohn in [29] brings out how it is possible to understand non-functional goals or quality goals or system constraints through creating user stories. And that is something that should be also done in AAOM. Mike Cohn also helps to understand that User Stories helps to shift the focus from writing about requirements to talking about them [30].

There are some nodes brought out by subjects that do not answer any research questions particularly but rather bring out new ideas. These following nodes are grouped in theme "New ideas".

Node	**Sources**	**References**	**Polarity**	**Type**	**Ref. by**
Models can be used for system documentation	1	3	Positive	Suggestion	c.
Use AAOM for business development, not only IT	2	2	Positive	Suggestion	a.d.

Table 10: Nodes for theme "New ideas"

As seen from Table 10: Nodes for theme "New ideas" customer brings out an idea about structuring the existing documentation in team collaboration software Confluence [31] where it is currently being held. However, it consists of pages with the system specification and is not

structured. As JIRA is also Atlassian's product, it could be possible to link JIRA tasks (goals) with Confluence documentation and therefore hierarchically structure it.

Another idea brought out by participants is that AAOM could also be used for product development. Product development also consists of a number of sub-goals needed to achieve the main goal. It can consist of IT goals, marketing goals, product development goals, etc. To do that, a goal management system like Jira needs to be used and also, dedicated tool to visualize goals is needed.

"Dedicated tools support needed" is one of the strongest node and is referenced by all subjects. It is also the biggest perspective of future work. A software to visually represent goals on top of Atlassian JIRA with possibility to modify model is currently being analyzed. This tool could also help with documentation structuring, as it would be possible to link specification against a specific goal. Creation and aspects of this tool would be another masters' thesis worth of work.

Summary

Agile agent-oriented modelling (AAOM) is analysed in this thesis. AAOM connects requirements engineering (RE) process with goal visualization. Goals are presented in a structured goal model, where higher level goals are connected to lower level sub-goals through links. Model additionally consists of non-functional goals (quality goals) and roles. User stories, part of agile software development, are the lowest level goals in goal model.

The goal of this thesis is to find out how does AAOM create an advantage over other modelling methods. To better analyse this question, it is split into sub-question addressing understandability and communication, visual representation and tooling of AAOM.

Case study method is used to analyse a consumer factoring system development in a finance-industry case. AAOM is used in this case and case subjects consist of customer, analyst and two developers.

Different data is gathered including interviews, history of goal model and lines of code. However, only interviews are used for analysis. The interviews are transcribed and coded. Transcription and coding results, nodes, are grouped in themes and scored by ranking formula to analyse gathered information. Scores are used to better evaluate research participants' statements.

Finally, an answer to the main research question is formulated. AAOM creates an advantage in software development process over alternative methods through better understandability and mutual communication, more effective time usage, generally suitable and clear visual representation of AAOM. Additional value can be added with proper tooling.

Kokkuvõte

Antud lõputöö raames uuriti agiilset agent-orienteeritud modelleerimist (AAOM). AAOM seob endas nõuete kogumise ja nende visualiseerimise. Nõuded esitatakse eesmärgimudelis struktureeritult nii, et iga alameesmärk on ühendatud kõrgema taseme eesmärgiga. Eesmärgimudel sisaldab veel mitte-funktsionaalseid nõudeid ning ka rolle. Madalaima taseme eesmärgiks on agiilsest tarkvaraarendusest tuntud kasutuslood (user stories).

Uurimustöö eesmärk on leida, kuidas ja millise eelise loob AAOM alternatiivsete meetodite ees. Arusaadavamaks uurimiseks on see uurimustöö küsimus jaotatud kolmeks alamküsimuseks: „kuidas saavutada parem arusaadavus ja ühine suhtlus tänu AAOM'le?“, „kuidas mõjutab visuaalne eesmärgimudeli kujutamine AAOM'i?“ ning „kuidas aitab tööriistade kasutamine AAOM'l alternatiivsete meetodite ees eelist luua?“.

Küsimustele vastuste leidmiseks kasutati juhtumiuuringul põhinevat metoodikat (case study) mille käigus uuriti konkreetses finantsasutuses järelmaksu tarkvara loomist kasutades AAOM'i. Projektis osalesid majasisene tellija, analüütik ning kaks arendajat.

Uurimustöö käigus koguti palju erinevaid andmeid, sealhulgas intervjuud, eesmärgimudelite ajalugu ning koodiridade arv. Analüüsiks kasutati kogutud intervjuusid, mis transkribeeriti ning seejärel kodeeriti. Koodidele anti kaalud ning need jaotati teemapõhiselt gruppidesse. Tänu kaaludele oli võimalik hinnata väidete valiidsust.

Juhtumiuuringu abil leiti vastused püstitatud küsimustele. Avastati, et AAOM'i eelis tarkvara arendusprotsessis seisneb parema arusaadavause loomises, meeskonna liikmete tihedamas suhtluses ning kõigile sobivas ja arusaadavas visuaalse eesmärgimudeli kujutamises. Tuvastati ka vajadus spetsiaalse modelleerimise tööriista järele, mis lisaks meetodile olulist lisandväärtust.

A. Appendix – Interview questions

A.1 Client questions

1. Are main concepts of AAOM clear? Can you explain in couple of sentences how you understood?
2. Is visual representation of your intentions in the form of goal models understandable?
3. Did collaborative modelling involve you more into the process?
4. How do you evaluate your participation in goal models creation? Did you want to do more or less? More high or low level?
5. How did elaboration sessions' execution work for you? Time it took, pauses taken, tooling setup, suitable time?
6. How clear was the process from your general idea to User Stories (lowest leaf)?
7. Did you understand User Stories presented to you?
8. Did quality goals and roles attached to functional goals seem reasonable and provide extra value?
9. Did User Stories link to quality goals and roles seem logical or simple?
10. How satisfied you were with time from main idea to implementable chunk of work - User Story?
11. Did tools used for modelling help with understanding or vice versa created confusion?
12. How the modelling method seem to fit in to rest of activities needed to run the project?
13. What do you expect from future iterations?
14. Did the goal model provide value throughout the project?
15. Did all elements in goal models contribute to end-product?
16. Did the system developed meet the goals in goal model?
17. Was the evolution of the goal model sufficient?
18. Would you use AAOM method again in other projects?
19. What other value did the AAOM provide?
20. Any other ideas about the AAOM method?

A.2 Analyst questions

1. Are main concepts of agile AAOM clear? Can you explain in couple of sentences how you understood?
2. Was information extraction from “client” easy with goal models? Structured? Logical?
3. Did collaborative modelling seem to involve “clients” more into the requirements elicitation process?
4. Did quality goals provide you as an analyst valuable insight?
5. Did User Story concept as a chunk of implementable value work?
6. Was User Story sufficient and clear goal what to achieve as analyst?
7. Did roles help to compose User Stories?
8. How hard it was to move from general ideas to specific User Stories?
9. How much time did it take to get models cleaned up?
10. How much time did it take to come up with small enough User Stories?
11. Was introducing developers to project using goal models easy? Questions asked?
12. How did tooling help/distract your effort to document models?
13. Did moderating requirements elicitation sessions work as they should? Should there be some instructions how to conduct elicitation session?
14. How the modelling method seem to fit in to rest of activities needed to run the project?
15. What do you expect from future iterations?
16. Did negotiating changes with client go better with goal models? Assuming there were changes?
17. How much did goal models change?
18. How much time did it take to reflect changes back to goal model?
19. How did quality goals affect user stories?
20. How much did existing user stories change?
21. Did developers understand the user stories?
22. How much did developers need extra clarification about user stories?
23. Has the tooling satisfied your needs so far?
24. Did your idea about the project outcome change after implementation?
25. Any other ideas or remarks about the modelling method?
26. Did the AAOM method make extracting requirements easier from client?
27. Did you have to spend much time on goal models?

28. Did creating user stories help your work with communicating requirements from clients to developers?
29. Would you use AAOM method again in other projects?
30. What other value did the AAOM provide?
31. Any other ideas about the AAOM method?

A.3 Developer questions

1. Are main concepts of AAOM clear? Can you explain in couple of sentences how you understood?
2. Did you get the idea why and what are you about to implement?
3. Did you see what value you are about to deliver?
4. Was the work presented to you small enough for implementation? By definition of user story - small enough to be implemented in a day or two?
5. Did you notice quality goals affecting your tasks?
6. How do you see User Stories will be implemented? In timely manner? Without problems?
7. How the modelling method seem to fit in to rest of activities needed to run the project?
8. What do you expect from future iterations?
9. Was the outcome of the implementation satisfactory for you?
10. Would you use AAOM in other projects?
11. What other value did the AAOM provide?
12. Any other ideas about the AAOM method?

B. Appendix – Codes and formula

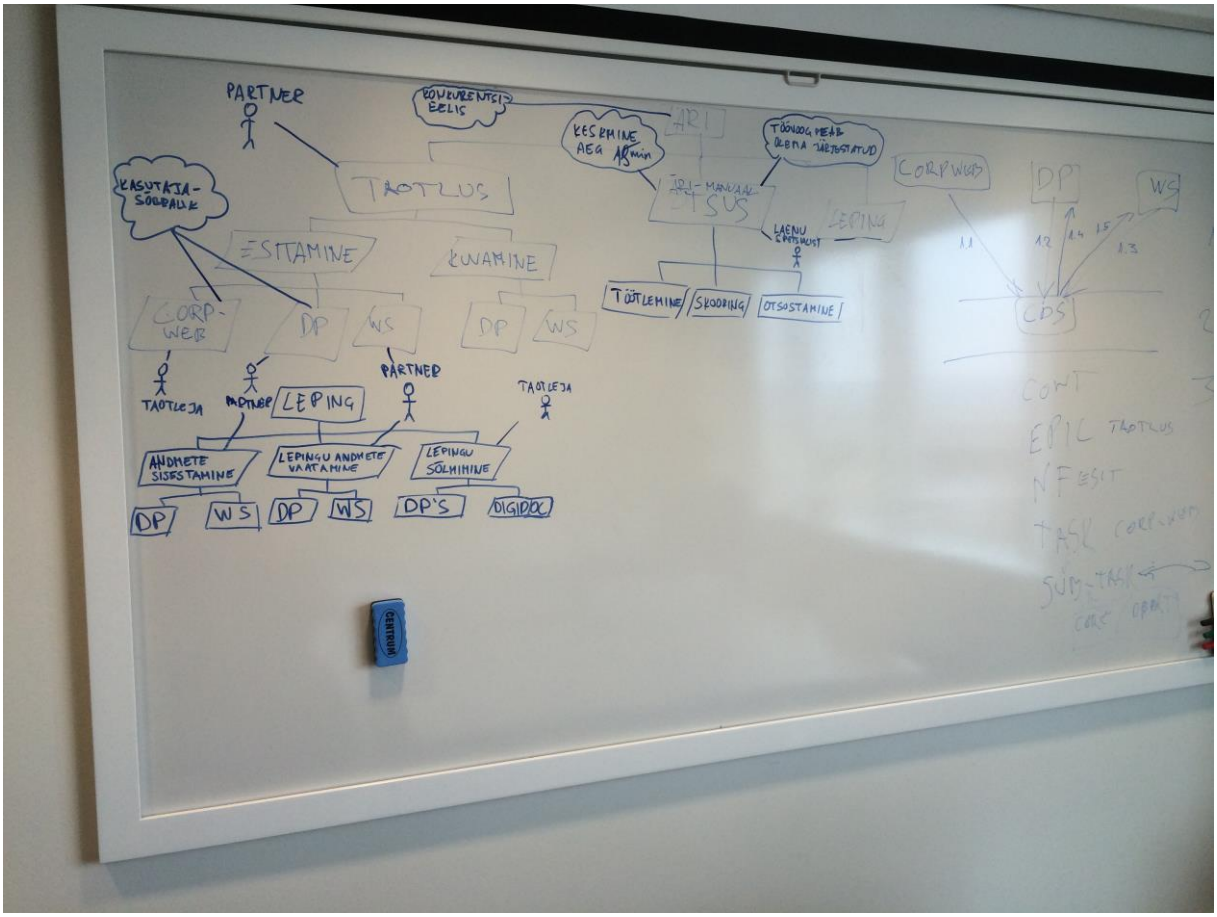
B.1 Codes and formula

	Sources	References	Experience	Analyst	Customer	Dev 1	Dev 2	SUM
Benefits								206
Secure feeling for project direction	8	10	19	3	1	2	2	99
Extracting information from customer better	3	4	9	3	0	0	0	21
Understood the value delivered	2	4	4	0	0	1	1	12
Mutual communication	3	3	9	3	0	0	0	18
System was developed according to goal model	3	3	6	0	1	1	1	15
Estimate work ahead	2	2	4	0	0	1	1	8
User story size good for development task	2	2	5	1	0	0	1	9
Helps to clarify existing documentation and system	2	2	5	1	0	1	0	9
Constructive modelling	1	1	2	0	1	0	0	3
Intuitively understandable	1	1	3	1	0	0	0	4
Roles help better understand the customer	1	1	3	1	0	0	0	4
Goal model shows positive results	1	1	3	1	0	0	0	4
User story too big for development task	1	1	2	0	0	1	0	3
Modelling suitability			0					129
Model didn't change	3	8	8	2	1	0	0	32
Modelling more for analyst and client, not developer	6	6	13	1	1	2	2	49
Would use goal models in other projects	4	4	9	1	1	1	1	25
Clarifies what needs to be done	2	3	6	2	0	0	0	12
Didn't watch model much during development	2	2	4	0	1	0	1	8
Modelling more for analyst, not client	1	1	2	0	1	0	0	3
Analyst has the most responsibility	1	1	2	0	1	0	0	3
Method clarification			0					113
Goal model clear	4	5	9	1	2	0	1	29
User story concept clear	4	5	10	2	1	0	1	30
From goals to user stories clear	3	4	7	1	1	0	1	19
Usage of quality goals clear	2	3	5	1	1	0	0	11
Usage of quality goals unclear	2	2	4	0	0	1	1	8
From goals to user stories unclear	1	1	2	0	0	1	0	3
Usage of roles clear	1	1	2	0	1	0	0	3
User story concept unclear	1	1	2	0	0	1	0	3
Quality goals link to user stories unclear	1	1	2	0	0	1	0	3
Didn't feel the need for quality goals	1	1	3	1	0	0	0	4

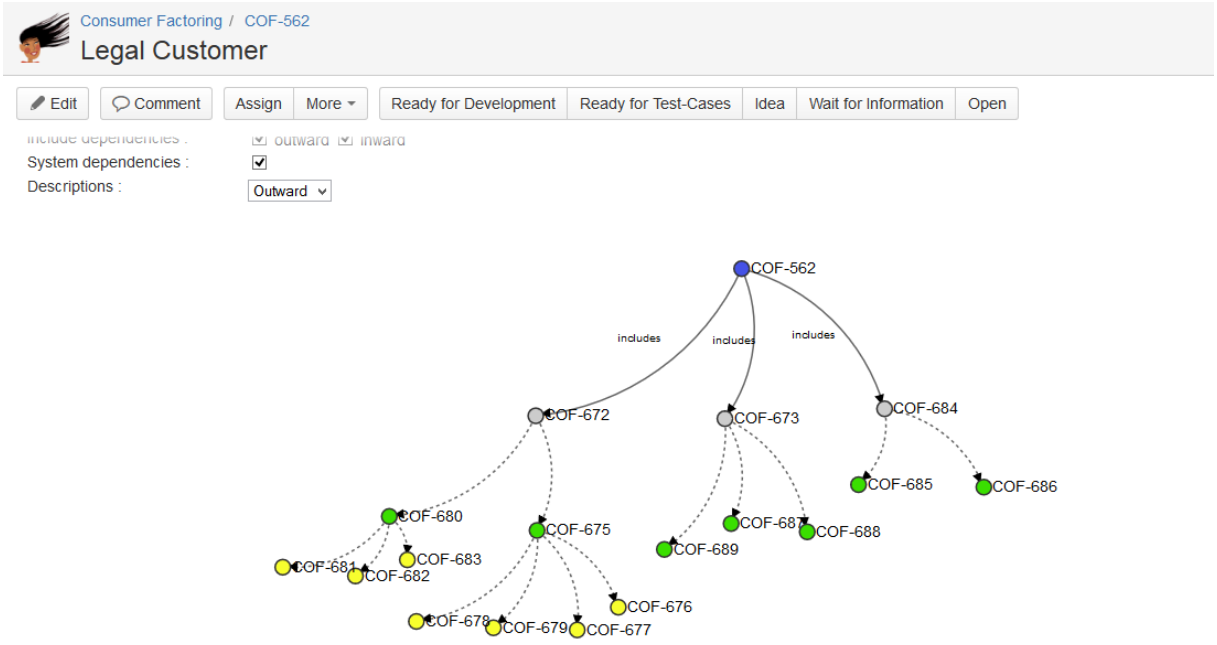
Collaborative modelling			0					50
Improving understandability	2	4	5	1	1	0	0	13
Involving participants	3	4	8	2	1	0	0	20
Having everyone on the same page	2	2	6	2	0	0	0	10
Few feelings about collaboration	1	1	2	0	1	0	0	3
Pinpointing problems	1	1	3	1	0	0	0	4
Expectations			0					45
Common expectations between stakeholders	3	3	7	1	1	0	1	16
Less unexpected changes	2	2	4	0	0	1	1	8
Customer ability to adapt with already delivered product	2	2	5	1	1	0	0	9
Can use existing model for future developments	1	1	2	0	1	0	0	3
Time estimation correct	1	1	2	0	0	0	1	3
Use agile methodologies	1	1	2	0	0	1	0	3
Get customer feedback	1	1	2	0	0	1	0	3
Visual representation			0					63
Visual representation - suitable in general	5	10	13	3	2	0	0	63
Tools usage			0					53
Dedicated tool support (is needed)	5	6	12	2	1	1	1	42
Need more integration	1	1	3	1	0	0	0	4
Commercial better	1	1	3	1	0	0	0	4
Good enough for starters	1	1	2	0	1	0	0	3
Time taken for modelling activities			0					30
Time used effectively	2	2	5	1	1	0	0	9
From idea to user story fast	2	2	5	1	1	0	0	9
Before using method (AOM) took lot of time	1	1	3	1	0	0	0	4
Refining goal models fast	1	1	3	1	0	0	0	4
Refining goal models slow	1	1	3	1	0	0	0	4
Initial user stories take time	1	1	3	1	0	0	0	4
New ideas			0					14
Models can be used for system documentation	1	3	2	0	1	0	0	5
Use AOM for business development, not only IT	2	2	5	1	0	1	0	9
Method comparison			0					12
Better than making notes	2	2	5	1	1	0	0	9
Similar to what was done earlier	1	1	2	0	1	0	0	3

C. Appendix - Model

C.1 Model drawn on white-board



C.2 Model drawn based on JIRA tasks



7 References

[1] K. Taveter and L. S. Sterling, The Art of Agent-Oriented Modelling, 2009.

[2] R. Grau and K. Lauenroth, "Requirements engineering and agile development - collaborative, just enough, just in time, sustainable -," p. 18.

[3] T. Tenso and K. Taveter, *Requirements Engineering With Agent-Oriented Models.*

[4] E. Hull, K. Jackson and J. Dick, Requirements Engineering, 2011.

[5] Wikipedia, “Data visualization,” [Online]. Available: http://en.wikipedia.org/wiki/Data_visualization.

[6] "Manifesto for Agile Software Development," 2001. [Online]. Available: http://www.agilemanifesto.org/.

[7] G. Fairbanks, Just Enough Software Architecture: A Risk-Driven Approach, Marshall & Brainerd, 2010.

[8] "Software Usability and Learning Curves," [Online]. Available: http://tynerblain.com/blog/2007/03/12/software-usability-learning-curves/.

[9] Trello, Inc, "Trello," [Online]. Available: https://trello.com.

[10] JGraph Ltd., "Draw.io," [Online]. Available: https://www.draw.io/.

[11] Wikipedia, "Factoring (finance)," [Online]. Available: http://en.wikipedia.org/wiki/Factoring_%28finance%29.

[12] Wikipedia, "Kanban (development)," [Online]. Available: http://en.wikipedia.org/wiki/Kanban_(development).

[13] Wikipedia, "Lean software development," [Online]. Available: http://en.wikipedia.org/wiki/Lean_software_development.

[14] Wikipedia, "Test driven development," [Online]. Available: http://en.wikipedia.org/wiki/Test-driven_development.

[15] Wikipedia, “Just-in-Time Manufacturing,” [Online]. Available: http://en.wikipedia.org/wiki/Just-in-Time_Manufacturing.

[16] P. Runeson, M. Höst, A. Rainer and B. Regnell, "Case study research in software engineering," p. 241, 2012.

[17] A. v. Lamsweerde, "Goal-Oriented Requirements Engineering: A Guided Tour," 2010.

[18] S. S. Johnson, *Requirements Engineering in an Agile Environment*, 2010.

[19] S. W. Ambler, "User Stories: An Agile Introduction," [Online]. Available: http://www.agilemodeling.com/artifacts/userStory.htm.

[20] M. Cohn, User Stories Applied: For Agile Software Development, 2004.

[21] S. Kuo, "Task Sizing in Agile Software Development".

[22] M. McLaughlin, "Agile Iteration Planning".

[23] Wikipedia, "Non-functional requirement," [Online]. Available: http://en.wikipedia.org/wiki/Non-functional_requirement.

[24] B. Flyvbjerg, “Five Misunderstandings About Case-Study Research”.

[25] "Nvivo10," QSR International, [Online]. Available: http://www.qsrinternational.com/products_nvivo.aspx.

[26] A. Strauss and J. Corbin, Basics of Qualitative Research. Grounded Theory Procedures and Techniques, CA: Sage Publications, 1990.

[27] J. Saldana, The Coding Manual for Qualitative Researchers.

[28] I. Kelder, *Mobile development using agile agent-oriented modelling*, 2015.

[29] M. Cohn, "Non-functional Requirements as User Stories," [Online]. Available: http://www.mountaingoatsoftware.com/blog/non-functional-requirements-as-user-stories.

[30] M. Cohn, "User Stories and User Story Examples," [Online]. Available: http://www.mountaingoatsoftware.com/agile/user-stories.

[31] Atlassian, "Confluence," [Online]. Available: https://www.atlassian.com/software/confluence.