MASTER THESIS

Requirements Refinement Using Domain knowledge

Author:

Ayman Younso

5997119

Supervisors:

Dr. Fabiano Dalpiaz

Dr. Sergio España Cubillo

A Master thesis document submitted regarding the requirements

for the degree of Master of Science

in

Business Informatics

Department of Information and Computing Science

November 7, 2018

Utrecht University

Abstract

Requirements Refinement Using Domain knowledge

By Ayman Younso

Getting clear and accurate requirements specifications is still a challenging task for requirements analysts for two main reasons: i) Despite the fact that analysts are experts in software technology, their knowledge in the domain area where the system is developed is often limited, thereby limiting the effectiveness of requirements elicitation; ii) Most of the requirements in agile development methods are captured using natural language, which is vulnerable to different interpretations and understanding leading to inaccurate requirements specifications. Although there are some methods that support the analyst in producing high-quality requirement specifications, a small number of researchers focus on agile requirements engineering with user stories. However, their approaches are either manual or do not use any kind of domain knowledge as a reference to support the analysts in refining the requirements. This thesis has been performed to identify the potential use of domain knowledge in order to help the analyst to deliver high-quality requirement specifications. We developed the ***Story Suggestor Tool***, which uses business process models as source of domain knowledge that can be utilized to suggest new requirements to the analyst. The development of the artifact has been guided by design science methodology. To validate our approach, an experiment has been conducted to check whether or not our developed artifact, the Story Suggestor Tool, helps the analyst in delivering higher-quality requirements. Based on the evaluation of the experiment results, we concluded that the tool helps the analyst in producing relatively more complete and correct requirements.

Keywords: Agile methods, requirements engineering, domain knowledge, BPMN in requirements engineering

Acknowledgment

This journey would not have been possible without the support of my family, professors and friends. I am especially grateful to my family for their continued support and encouragement.

I would like to give special thanks to my thesis advisor Dr. Fabiano Dalpiaz for offering his help on my thesis. His office was always open whenever I had a question about my research or writing. He consistently allowed this paper to be my own work but steered me in the right the direction whenever he thought I needed it. I am also grateful to my second supervisor Dr. Sergio España Cubillo for his valuable feedback.

I would also like to thank the students who were involved in the validation experiment for this research project. Without their passionate participation and input, the experiment could not have been successfully conducted.

Contents

Chapter 1: Introduction

Having clear and accurate requirements is one of the key success factors for software projects. According to Pohl [3], in order to get an optimal collection of requirements, the three dimensions of specification, representation and agreement need to be satisfied as shown in Figure 1.

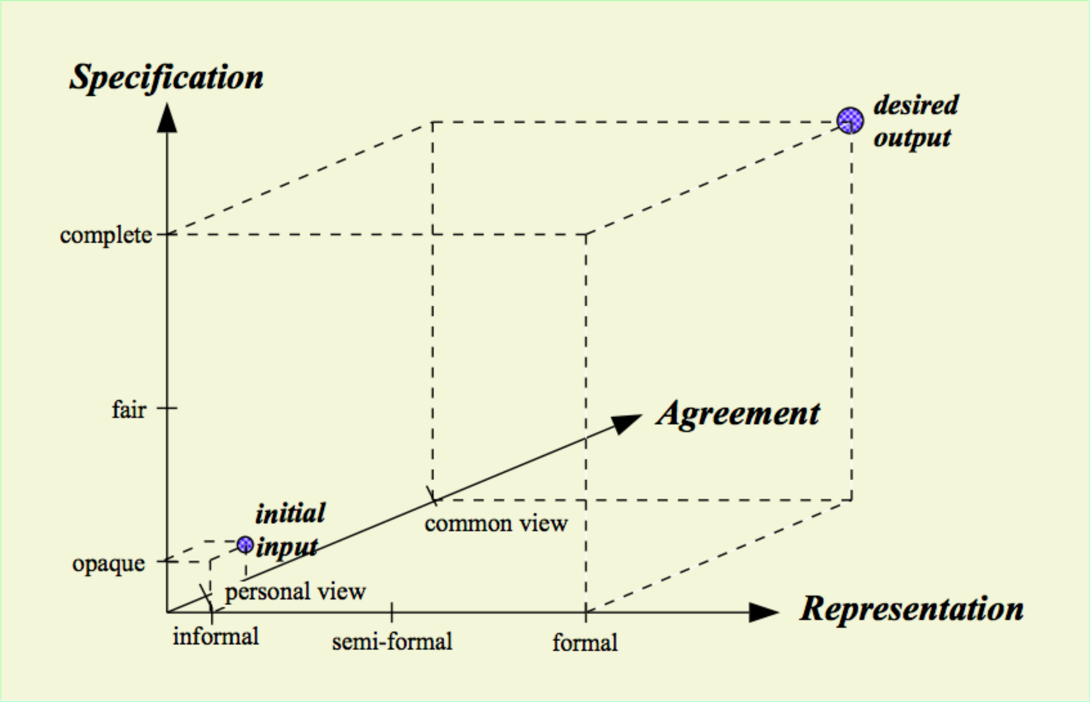

Figure 1. The three dimensions of requirement engineering by Pohl [3]

The goal of the specification dimension is to have full specifications with high quality that cover all the operational needs. Hence, clearly addressing what the system should actually do is the essence of requirements specifications.

The representation dimension focuses on the way stakeholders express their needs. Different stakeholders express their requirements in different ways like using informal language, sounds or pictures. The goal is to get a more formal representation than the initial one.

The agreement dimension reflects the level of accordance among stakeholders on the requirements specifications, as different stakeholders have different interpretations for the same specification. The goal is to move from personal view into common view [3].

In this thesis, we focus on the specification dimension that concerns requirements understanding. As stated by Pohl [3], the specifications dimension represents the baseline for the other two dimensions: representation and agreement.

Obtaining a complete specification is necessary to produce a high-quality requirements specification document that leads to clear expectations about the functionality and the characteristics of the system. Having poor and low-quality requirements specifications will compromise the whole RE process and may lead to serious implications on the system developing process, possibly leading to a system that does not meet its objectives.

Capturing system requirements specifications is a challenging task. In fact, getting vague and ambiguous requirements specifications is common at the beginning of the requirements engineering process for two reasons:

1. Requirements specifications are mainly captured using Natural Language (NL) and expressed in, e.g., English, Dutch or Japanese. Although techniques for natural language processing (NLP) are increasingly advanced [11], processing the requirements documents sufficiently well through computer programs is still a challenging task [1]. Moreover, natural language is more vulnerable to different understandings and interpretations that lead to opaque requirements specifications and may result in undesired system behavior [5].

2. Despite the fact that requirement analysts are experts in software technology, their knowledge of the problem domain where the system will be used is often limited. This lack of domain knowledge leads the analysts to perform poor requirements elicitation and, as a result, to produce low-quality requirements specifications [2].

To overcome the previous challenges, several studies have been conducted using different techniques. Harmain et al. [9] developed a computer-aided software engineering (CASE) tool that supports the analyst in refining the requirements by using an initial UML generated from a semantic network. The UML model represents object classes and the relations between them. Then the UML used as an input for the graphical CASE tool which allows the analyst to refine the requirements manually. By editing the UML model class diagram using the CASE tool the the analyst can add new classes attributes and edit the current relations. Next, for evaluation comparing each class model from the CASE output to reference standard model as the following: correct if it matches an element in the standard model; incorrect if it does not match an element in the standard model; extra if it is valid information from the text but is not in the standard model.

In the context of agile requirements engineering Lucassen et al. [13] extract a conceptual model from a collection of user story requirements using the automated tool VN “Visual Narrator”. The tool and its output can be used to detect dependencies, redundancies, and inconsistencies between requirements. For example, suppose that we have the following user story as an input for the Visual Narrator:

As a student, I want to upload my academic paper to the system, so that I can hand it into my supervisor

After processing the previous user story via VN the conceptual model of the output shown in Table 1.

Subject	**Predicate**	**object**
Student	Upload	Paper

Table 1. Conceptual model as an output of Visual Narrator tool.

Although important results have been achieved by Harmain et al. [9] and Lucassen et al. [13], their approaches are either manual as Harmain et al. [9], or do not use any kind of domain knowledge as a reference to support the analysts in refining the requirements.

The main concern in requirements elicitation is to detect and reveal information from different knowledge sources to clearly identify the system requirements [63]. Nowadays, domain knowledge documentation is considered one of the main sources of valuable information used by companies [64]. Such information is usually captured as written text and as graphical models [65]. Often, graphical representations of the work being conducted, such as business process models, and domain ontologies, are used for communicative purposes between various stakeholders as it helps them in understanding how work is being performed and where improvements can be made.

In this thesis we evaluate the use of domain knowledge in order to refine the requirements engineering user stories, and when we refer to domain knowledge in the context of this thesis, we focus mainly on the use the following domain knowledge:

1. Domain ontology.
2. Business process which represented by BPMN (Business Process Management Notation).

We made this choice because the aforementioned domain knowledge types are the most widely used in information technology field.

Some researchers have investigated the use of domain ontologies as domain knowledge for requirements elicitation, and their studies show promising results. Reubenstein et al. [8] mainly focus on bridging the gap between formal and informal specifications. Particularly, the authors focus on the representation dimension and not the specification one of Pohl's cube [3]. For that they used reusable templates called Cliché to assist a requirements analyst in creating and modifying the requirements. The Cliché provided common forms of requirements specification in a specific domain.

Each cliché comprises of set of roles and constraints between them. Following the example in Figure 2, the first argument of the notation is the name of the cliché. Next, is a list of the parents, the third argument defines the roles of the cliché. The main body of a “def-cliché” defines the constraints on the cliché, it could be for instance: Preconditions, Consequents.

Although the authors addressed some key challenges in requirements acquisition, still their tools do not interact directly with the end-user or the domain expert.

```
(Def-Cliche Tracking-Information-System-Report (Report)
(Def-Roles Target)
:Preconditions (Tracking-Information-System (Home-System ?Self))
:Consequents ( = ?Target (Target (Rome-System ?Self))>
:Overview-Text
(!Text "*The tracking-informationisystem-report ?self provides
information about what the state of the ?target is believed to be."))
```

Figure 2 example cliché frame types, extracted from The Requirements Apprentice: Automated Assistance for Requirements Acquisition, Reubenstein et al. [8].

The work by Breitman and Leite [6] concerns the ontology construction and development process using LEL (Language Extended Lexicon), a kind of electronic version of dictionary that can be used as domain knowledge in requirements elicitation processes. LEL is an approach that derives from the semantic web community [10]. However, their work neither develops a tool nor a method for requirement engineering that directly helps the analyst to get more accurate requirements. Saeki et al. [2] propose a method for requirements elicitation that uses ontologies. They use a semantic function with quality metrics to indicate the relationships between two concepts and to evaluate the requirements specifications quality. Moreover, they use inference rules on domain ontology to allow the analyst to refine low-quality specifications. However, their approach is manual and therefore does not scale up to large specifications.

The limitations of previous works, which either do not use an ontology or use it in a manual fashion, evidence the existence of a gap for semi-automated approaches to get refined requirement specification based on domain ontologies.

As explained by Deemer et al. [56], the current scenario for refining requirements using the Scrum agile development method is that the product owner together with the development team have to work cooperatively to refine and understand the ambiguous requirements in process called Product Backlog Refinement, where the product backlog represents the system requirements in from of User stories, use cases or any other useful requirements approaches. Often the Product Backlog Refinement process is far from being trivial and it require detailed analysis for the requirements, breaking down a large user story into smaller stories and the involvement of other stakeholders that better understand the requirements in the applied domain, the flowchart of the product backlog refinement is shown in Figure 3.

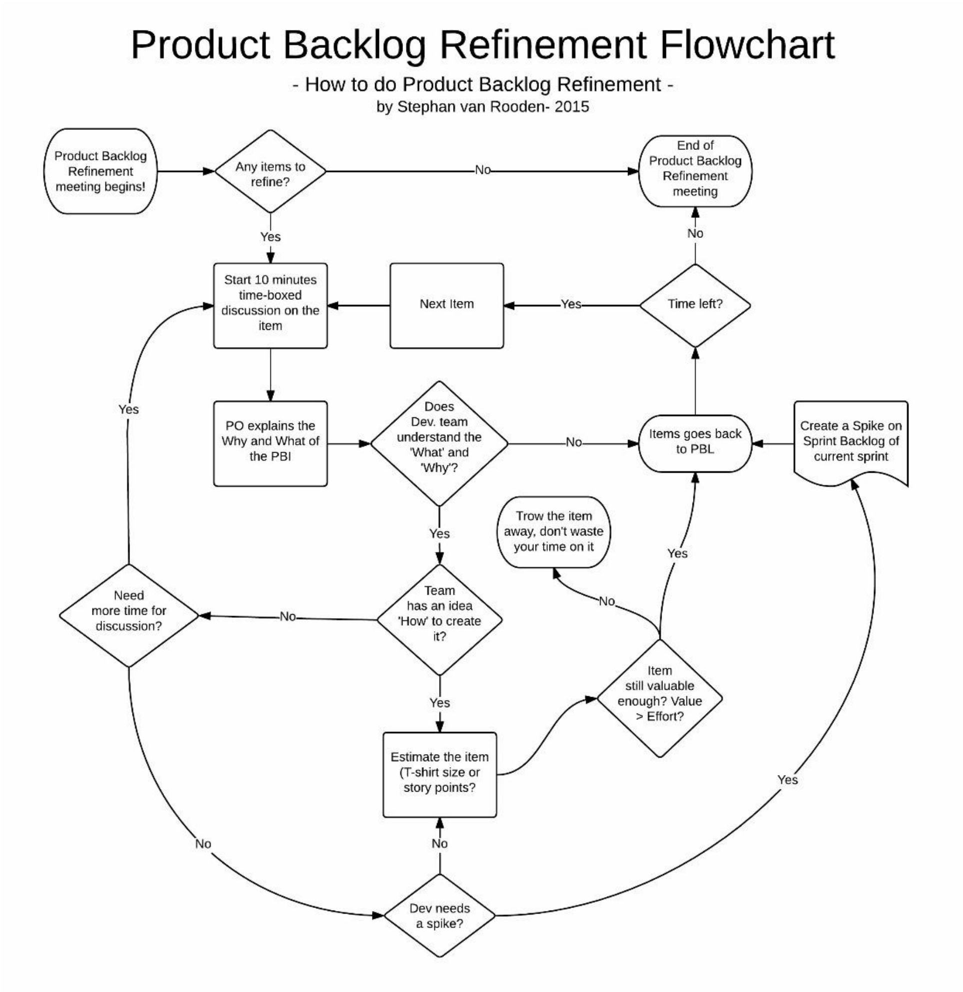

Figure 3. Product Backlog Refinement explained. Retrieved from https://www.scrum.org/resources/blog/product-backlog-refinement-explained-33 S. Rooden [71].

The thesis proposes two approaches for the semi-automated refinement of user stories, we try to avoid the aforementioned traditional method (Product Backlog Refinement). We mainly depend on systematically analyzing the user stories requirements and transforming it from natural language into structured data. The necessity to extract information from natural language documents motivated a lot of research on application of text analysis in requirements engineering [7] and then to store the structured data into domain ontology.

The underlying idea for the first approach, is to disassemble each requirement into a number of typed entities. Our domain ontology system will consist of predefined entities, types and relationships between them for a specific application domain. Also, a new set of entity, type and relation can be added to the ontology incrementally. According to Breitman et al. [6], having a rich and high-quality domain ontology will help the analyst get more insight of the system application domain; indeed, ontologies help both people and machines to communicate concisely, supporting the exchange of semantics and not only syntax [4]. Hence, the domain ontology works as a knowledge base for both domain experts and requirement analyst, and the domain ontology itself is a valuable requirement engineering product, as stated by Breitman et al. [6].

On the other hand, business process models, and in particular models following the Business Process Model and Notation (BPMN) language, are widely used in RE as it has been designed to assist the needs of domain experts and business analysts.

Having a complete Business process is essential to get better insight about the business domain where a new system need to be developed. Business process models are considered one of the most commonly used type of conceptual models, it depicts the business workflow and the business added value that created by different stakeholders [57].

A standard notation for modeling business processes is the Business Process Model and Notation (BPMN) [58]. BPMN is a graphical notation, maintained by the Object Management Group - (OMG), created for the representation of business processes and based on workflows [59], with great advantage as it easy to understand and interpret by all the stakeholders in an organization.

The rest of the paper is structured as follows. Chapter 2 introduces our goal, hypothesis and research questions. Chapter 3 presents our research methodology. Chapter 4 literature review. Chapter 5 presents the potential use of domain ontology to refine user stories. Chapter 6 introduces our main approach in using BPMN to refine user stories. Chapter 7 describes the experimental framework used to validate our developed artifact Story Suggestor Tool. Chapter 8 presents our conclusion and future research.

Chapter 2: Goal, Research Questions and Hypothesis

Our goal is to define a semi-automated requirements refinement process for Agile requirements engineering with user stories, so that we can get high-quality requirements specifications. Through this research, we would like to address the following hypothesis and knowledge questions which are related to the objective of this thesis:

MRQ: Does the use of the semi-automated process for supporting requirements refinement lead to higher quality requirements compared to the traditional product backlog refinement?

Based on our main research question three sub-questions have been formulated as the following:

RQ1.1: What lessons learned can be gained from the existing literature on the use of domain knowledge in RE?

RQ1.2: Do analysts who use domain knowledge deliver more complete requirements than analysts who use the traditional backlog refinement?

RQ1.3: Do analysts who use domain knowledge deliver more correct requirements than analysts who use the traditional product backlog refinement?

The research questions above have been translated into the following hypothesis:

H01: Using domain knowledge as refinement for requirements engineer increases the requirements quality.

According to IEEE 830 standard [12] and Kaiya et al. [1], there are four main characteristics that define the quality of the requirements specifications [12]:

- **Correctness:** A system requirements specification (SRS) is considered correct, if and only if it meet the system objective and for that each system requirement specification should be mapped to another applicable standard to make certain that it is meet each other's. Alternatively, the stakeholders can determine if the SRS correctly reflects the actual needs.)

  An example of an SRS which is not correct, if a one requirement stated that the product owner can rank his own product, whereas the objective of the developed system is to get the end user ranking for the products.

- **Completeness**: SRS is considered complete if it comprises all important requirements in terms of functionality, performance, design constraints, attributes, or external interfaces. Furthermore, adding definition of the responses of the system to all possible input values and also adding definition of all terms and units of measure.

For example: a requirement considered not complete when a system need to be developed for ticketing system and there are no requirements for issuing tickets.

- **Consistency:** SRS consistency focus on internal consistency in which no contradiction between the sub requirements is occurred.

  For example, the following user story violates the internal consistency: One requirement may state that "As a student, I want to have a view access to my grade list" while another may state that "As a student, I want to have a full access to my grade". There is inconsistency between the two user stories.

- **Unambiguity:** Unambiguity for an SRS means that there is only one interpretation for each requirement specifications. In cases where a term used in a particular context for describing requirement specification have multiple meanings, the term should be included in a glossary where its meaning is made more specific.

  An example of ambiguous user story is a user story that uses ambiguous terms such as: "As a student, I want to be able to edit the content that I uploaded to my personal page", without giving a clear definition to what "content" exactly mean, since "content" is broad term and it might refers to a wide range of different things.

As such, a specification R2 is more refined or is of higher quality than specification R1 if and only if:

Correctness R2 > Correctness R1

Or

Completeness R2 > Completeness R1

Or

Consistency R2 > Consistency R1

Or

Unambiguity R2 > Unambiguity R1.

However, in our research we focus solely on improving correctness and completeness as the two quality characteristics, without compromising the other two characteristics consistency and unambiguity.

Chapter 3: Research Method:

We explain the research method that will be used in order to investigate the research questions. We need to iterate over two activities:

- Designing a tool that refines a collection of requirements.
- Empirically investigating the performance of this tool.

Therefore, the design science methodology [14] seems to be the best fit for the research. A design science project is an iterative process between solving design problems and answering the knowledge questions. Designing the problem (design cycle) comprises three main phases: problem investigation, treatment design, and treatment validation. We will repeat the cycle iteratively two times and check whether the treatment satisfy the stakeholders needs as it shown in Table 2. Implementing the treatment and evaluating the result will complete the so called the engineering cycle is out of the scope of this paper. Therefore, this thesis will focus solely on the design cycle [14]. As it shown in Figure 4 and Figure 5.

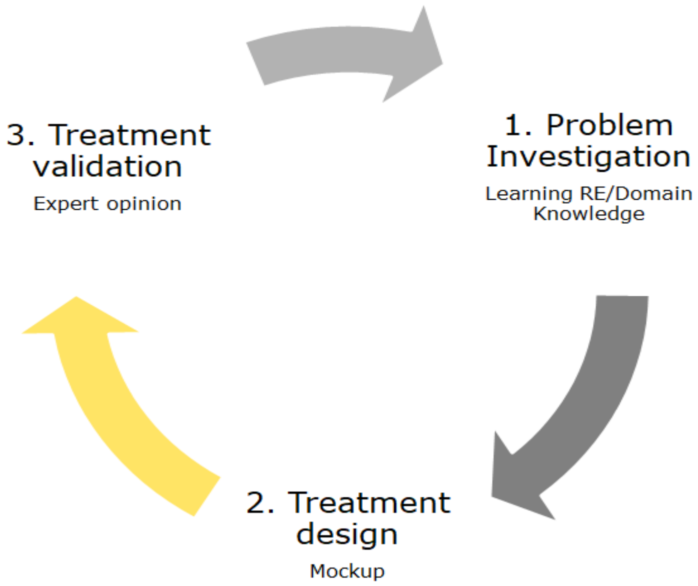

Figure 4 design cycle first iteration

3.1 Problem Investigation

In order to have a get better understanding about the problem that we investigate, for the first iteration, a literature review has been done as presented in Chapter 4 focusing mainly on the use of user stories requirements engineering in Agile methods, different approaches for requirements modeling, in addition to the use of Domain knowledge in RE and the diverse techniques for using domain knowledge sources in RE. For the second iteration we did further investigation about the problem of refining the RE adding to that the expert opinion from the first iteration.

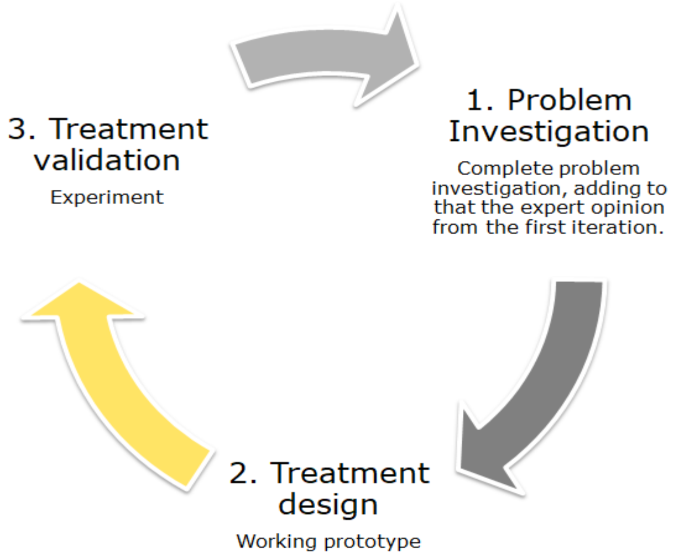

Figure 5 design cycle second iteration

3.2 Treatment Design

We design a semi-automated mockup as a proof of concept as illustrated in Chapter 5 and Chapter 6. We use a set of user stories from the education domain for facilitating the construction of our tool and for testing purposes, for the first iteration we proposed our first approach by using domain ontology as a domain knowledge, for that we developed ontology using the Protege tool, also the we used of Visual narrator tool and ontology matching tool. The aim of the first iteration was to find out whether the proposed concept is helping the requirements analyst to get more precise and high-quality requirements. In the second iteration, we proposed another domain knowledge source, BPMN and for that we will develop a full working prototype.

3.3 Treatment Validation

In order to find whether our artifact solved the aforementioned problem or not. In the first iteration we considered the expert opinion in the RE domain, about the mockup design and the efficiency of the output and we found out in chapter 7 that using DO as domain knowledge is not a feasible solution at least from the companies' point of view, since just a few companies used a DO. The experts were a business analyst or professor who involved in RE projects. Based on the feedback from the expert, more refinement to our solution was made in a sense of using another domain knowledge –business process- to refine the user stories. For the second iteration, we conduct experiment in which it involves a control group and treatment group, then we compare and analyze the results to find whether our proposed solution meet the objective in helping the analyst in refining the requirements.

Phase	First iteration	Second iteration
Problem investigation	Prepare for the design of a refining RE using domain ontologies by learning more about the RE, modeling RE techniques, ontologies building and matching. And the current available RE refining methods and techniques.	Complete problem investigation, adding to that the expert opinion from the first iteration. Further investigation about using BPMN as domain knowledge to refine RE
Treatment design	Mockup/proof of concept/ preliminary design for the tool	Working prototype
Treatment validation	Expert opinion	Experiment

Table 2. The three phases of Wieringa's design cycle during the two iterations.

Chapter 4: Literature Review

To get a better understanding of the domain area of requirements engineering and the domain knowledge, a literature review was performed to address the following question:

RQ1.1: What lessons learned can be gained from the existing literature on the use of domain knowledge in RE?

The literature review helped us to get better insight about Agile requirements engineering with user stories, explore the use of the natural language processing in RE, requirements modeling, building and ontology learning approaches and using BPMN in RE.

4.1 Background on RE

4.1.1 Agile RE User Stories

A user story is a description of a feature that provides business value for both developers and products owner. It is a cooperative working way between product owners and the system developers in order to get a clear insight about the system behavior.

Although many different templates for writing user story exist, in this thesis we use the Connextra template (Lucassen et al. [40]):

Template: "As a <type of user>, I want <goal>, so that <some reason>"

There are many benefits for using user stories [38]. User stories are, to some extent, understandable by both users and developers, thereby empowering users as actual team members and making it possible for them to actively participate in design the system that they are going to use. User stories are an efficient way to transfer the implicit knowledge between the users and the developers, since they foster the informal exchange of ideas between the team members. They also help the developers to manage and prioritize the stories based on their significant. In addition, user stories motivate the opportunistic development approach in which the developers can swap easily between the requirement on different level of detail [38].

User stories support deferring detail, in terms of writing a more general or epic user stories in the first phase and then go into more details later on. Moreover, user stories match well the fast-paced, iterative development method like Scrum and other agile development methods. It is no surprise that user stories are the predominant method to capture requirements in agile software development [39]. Also, many agile methods recommend gathering requirements using user stories [41].

4.1.2 Linguistic Analysis / NLP

Requirements define what the stakeholders need and what the system must include to satisfy the stakeholders' needs, and for that the use of natural language is widely common it considers as the most important medium for requirements documents. According to Mich et al. [42], almost 80% of all requirements documents are written in common natural language.

Many natural language processing (NLP) tools and approaches have been developed to support the processing of NL requirements documents. According to Berry et al. [43], the NL supporting tools fall into four categories with full or semi-automation processing and based on the precision and recall the authors define the tool strength, the four categories are as the following:

1. Requirements quality analysis: tools that helps to detect deviations and defects from best practices NL requirements documents. e.g., ARM and QUARS, by Fabbrini et al. [46] and to highlight ambiguous requirement statements, e.g., SREE and Chantree's nocuous ambiguity finder [72].

2. Requirements model generation: tools to generate models from NL descriptions, it detects classes, variables, and associations form requirement documents and then transform it into abstract model for example the work by Popescu et al. [45].

3. Abstraction identification: tools to identify the key abstractions from NL documents, a tool that help the analyst in get better understanding of an unfamiliar domain. The abstraction comprise the main terms and concepts and it could be represented as domain ontology that helps the analyst to gain knowledge about the domain where he needs to develop the system. The work by Goldin et al. [44] is an example for this category.

4. Links discovery: tools to discover trace links among NL requirements statements or between NL requirements statements and other artifacts of the development process e.g. Poirot tool developed by Lin et al. [47]

4.1.3 Requirement Modeling

Software systems consist of complex processes and the correspondent textual requirements are often difficult to understand. Moreover, following the interaction between different elements based on textual requirements is not an easy task. Therefore, the need to reduce the textual requirements extensivity and complexity become a necessity. Requirement modeling simplifies the process and shows the system behavior in a clear and a more understandable way. Furthermore, the interaction between diverse elements are explicitly illustrated in the modeled requirements. The benefits of modeling requirements are depicted in Figure 6. The left-hand side shows four textual requirements, while the right-hand side shows the model diagram of the corresponding requirements.

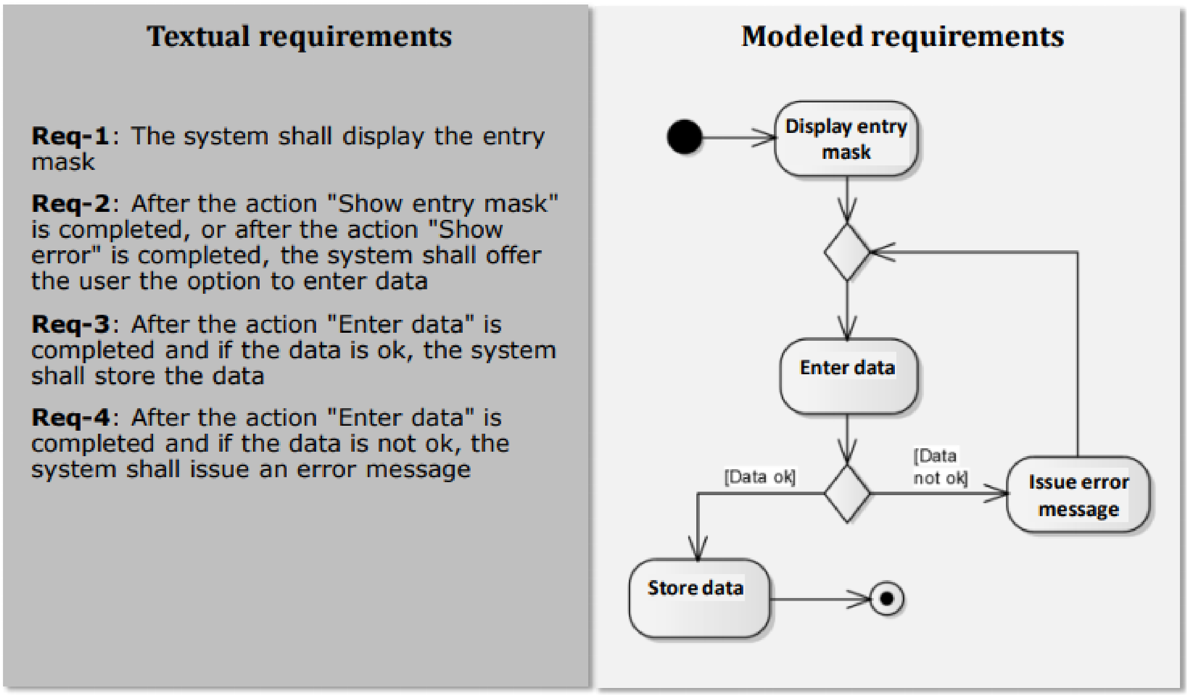

Figure 6. Example of textual requirements vs. modeled requirements extracted from "Handbook of Requirements Modeling IREB Standard" Weyer et al.[48].

From the previous example, we notice that modeling requirements is easier to understand and it provides a clear view of the main activities and the system functionality, while in the textual requirements this information is often presented implicitly [48].

4.1.3.1 Applications for Modeling Requirements

According to Requirements Modeling IREB Standard by Weyer et al. [48], there are three main applications for modeling requirements in requirements engineering:

- *Modeling Requirements as a Means of Specification:*
  In this case, in order to define the system requirements, requirements diagrams are used as a medium for identifying the system requirements. It is also possible to support the requirements diagrams with textual explanations, especially when a text is more evident than diagrams.

- *Modeling Textual Requirements for The Purpose of Testing:*
  A requirements diagram is created to examine the comprehensibility and inconsistencies of textual requirements. Hence, using the requirements diagram helps in fixing all the deficiencies within the textual requirements.

- *Modeling Textual Requirements for Clarity:*
  In order to clarify a complex textual relationship that represent the system behavior, a modeled requirement is used to simplify and to explain the extensive textual requirements.

4.1.3.2 Views in Requirements Modeling

As stated by Pohl et al. [49], three main basic views for functional requirements consider the building block for other different views are as it illustrated in Figure 7: (1) the static-structural view, (2) the behavioral view, and (3) the functional view.

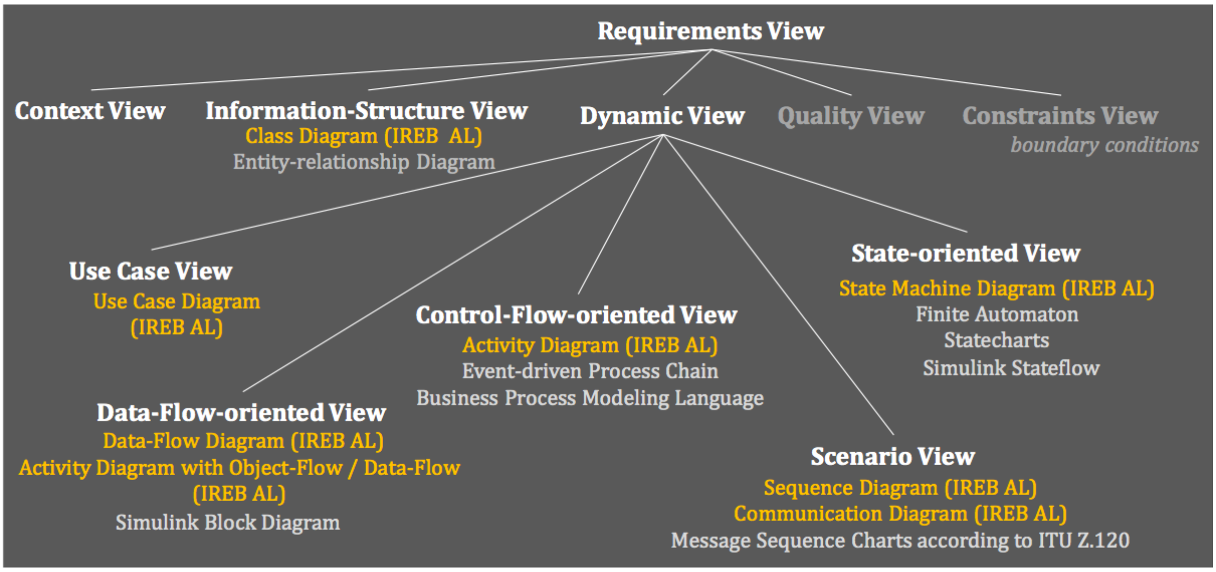

Figure 7: Views in requirements modeling in the IREB advanced level module "Requirements Modeling", extracted from "Handbook of Requirements Modeling IREB Standard" Weyer et al.[48].

- *Context view:* aims at representing how the system interacts with external entities and what their responsibilities are. This viewpoint is also vital to understand who are the main stakeholders and what are their interests concerning the system.

- *Information structure view:* Describes the way that the architecture stores, manipulates, manages, and distributes information and the way that information moves around the system and where the data accessed and modified.

- *Dynamic view:* the main concern of this view is the dynamic aspects of the system functionality, an example of models under the dynamic view are activity diagrams, state machine diagrams and data flow diagrams.

- *Quality view:* focus on the quality aspects of the requirements that affect different system elements for instance performance, reliability and robustness can be modeled by adding a note of explanation to the requirements quality diagram.

- *Constraints view:* mainly focus on a limitation or restriction for requirements for example organizational regulations or technological constraints. Such constraints could be modeled using class diagram.

4.1.3.3 Goal-Oriented Requirements Engineering

Goal-Oriented Requirements Engineering (GORE) has grown into an important area of research in the past decades [51]. (GORE) is defined by Van Lamsweerde [50], is it the desired target that the system should achieve or software-to-be through cooperation of various stakeholders, devices and 3rd party system within the organization.

GORE brings several benefits to RE practice, such as: it supports a wider system engineering perspective compared to the traditional RE methods, focus more on the reasoning behind the requirements and on top of that, GORE gives grounds for each requirement existence. Besides, it provides precise criteria for sufficient completeness of a requirement specification and it could also be used to detect and manage conflict between requirements.

Although many goal modeling languages emerged to support the RE process, KAOS and i* frameworks are the most dominate tools for goal-oriented modeling as stated systematic mapping study that has been done by Horkoff et al. [51].

4.2 Representing Knowledge

4.2.1 Building Ontologies

Ontology is an explicit formal specification of how to represent the entities that exist in a given domain of interest and the relationships between them. The nature of requirements engineering involves capturing knowledge from diverse sources including many stakeholders with their own interests and points of view. Therefore, there is potential in using ontologies in requirements engineering. According to Ballejos et al. [73], there are many benefits of using domain ontology in requirements engineering. The potential uses of ontologies in RE includes revealing of requirements ambiguity and helping in refining insufficient and incomplete requirements. Adding to that the ontology helps in dynamic and changing requirements environment by providing constant requirements revision.

There are several tools that support the development of ontologies. Some of these tools are outdated and not supported anymore, while others keep developing and evolving to support wider variety of tasks that facilitate the ontology building process. In this section we will compare some ontology development tools and briefly discuss some of the state-of-the-art tools.

The main criterion for comparison of these tools are the implementation language, import and export format the availability weather the tool is free open source or a license is required. In addition to versioning capability, the use of ontology libraries and beside the use of reasoner for evaluation and consistency checking. We mainly depend on Rastogi et al. [19], Kaur et al. [20] and Slimani et al. [21] for making our ontology tools comparison table. (See the Appendix A for the full table of tool comparison).

Protégé: a free open source ontology editor created at Stanford University that is very popular in the field of Semantic Web and in computer science research. Protégé, developed in Java and its source code is released under a free license (the Mozilla Public License). Protege is probably the most popular ontology development tool. Protege ontologies can be imported and exported in a variety of different formats, including RDF/RDFS, OWL and XML Schema formats. Protege facilitates rapid prototype and application development and has a very flexible architecture via a plug-and-play environment.

Moreover, variety of plugins have been developed by the researchers (e.g., the PROMPT/Anchor-PROMPT plug-in for ontology merging [15], plug-ins for versioning support [16], and plug-ins for collaborative ontology development [17]). Recently, a lightweight OWL ontology editor for the web (Web-Protege) [18] has been proposed. Protege Web Browser is a Java-based Web application that allows the user to share and set permissions for specific project. Also, it provides a full change tracking and revision history, adding to that it supports multiple formats for upload and download of ontologies (supported formats: RDF/XML, Turtle, OWL/XML, OBO)

Figure 8, shows a simple example of domain ontology for education system created for testing purposes, the main classes are Student, Professor, City and Netherland. In the example “Mike” is an individual or instance for the class student, while the relationships are represented in our example by “Studentin”, IsaCity, hasaCity.

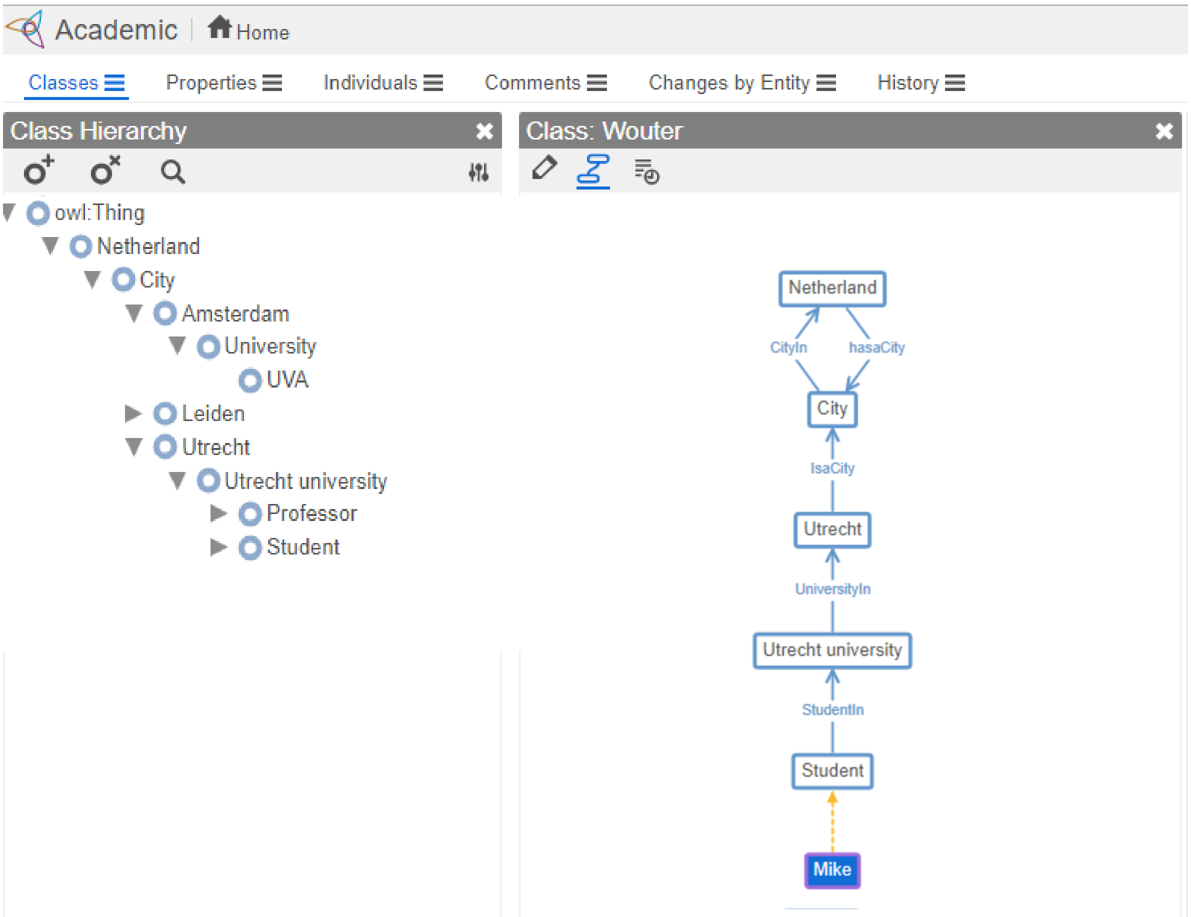

Figure 8. WebProtégé example for simple education system

OILEd: OIL Editor is a simple ontology editor and ontology demonstration tool that supports OIL-based Ontologies construction. The basic design is quite similar to other ontology tools such as Protégé5 and OntoEdit, It integrates a reasoner (FaCT) and extends the expressive power of other frame-based tools. OilEd can import and export ontologies in the RDF, OIL, SHIQ, dotty, DAML + formats.

Ontolingua: a form-based Web interface ontology tool based on the KIF knowledge interchange format for develop ontologies. Ontolingua, supports Ontology creation and browsing in distributed and collaborative environment. Using Ontolingua, it is possible to export or import the following formats: KIF, DAML+OIL, OKBC, Prolog, LOOM.

WebOnto: a tool which provides a web-based visualization, browsing and editing support to develop and maintain ontologies and knowledge models. An ontology can be viewed as a model of the conceptual structure of some domain and WebOnto, provides the capability to represent this graphically. It can export ontology to OCML, GXLRDF and OIL format.

Swoop: a tool for creating, editing, and debugging OWL (Web Ontology Language) Ontologies. SWOOP is a short for Semantic Web Ontology Editor. It provides an environment with a look-and-feel similar to that of a web browser. Reasoning can be performed using an attached reasoner (such as Pellet).

Neon: a toolkit for ontology management which provides run-time and design-time ontology alignment support, it provides storage, reasoning, querying, versioning and security services. The GUI provides user front-end components, including editors with text-based, graph-based and form-based interfaces.

4.2.2 Ontology Learning Approaches

Ontology learning is the process of extracting ontological elements (conceptual knowledge) from input corpus and building ontology from them [22]. Based on the data type from which the ontology learned, the ontology learning systems can be classified to unstructured, semi-structured, and structured data learning.

Unstructured data is documents consists of natural language texts such as Word, PDF documents and books. Semi Structured data is text in HTML, XML files, Wikis and User Tags dictionaries like WordNet [29] or the Wiktionary [30], while structured date are the database schemas, existing ontologies and knowledge bases. [22] [23].

Different techniques and technologies have been used for Ontology learning such as machine learning, knowledge acquisition, natural-language processing, information retrieval, artificial intelligence, reasoning and database management [23] [24]. However, using the natural language processing (NLP) is common among all the techniques, and the following categorization of ontology approaches consider the used technique beside the (NLP) [25].

4.2.2.1 Learning from Structured Data

4.2.2.1.1 Statistical Approach

Sanchez et.al. [26] developed algorithm that analyses a large number of websites to find important concepts for a specific domain by studying the initial keyword's neighborhood. Next, statistical analysis is performed to select the most adequate concepts from a set of candidates. Finally, the selected classes used to build the ontology. To find new terms and to build a hierarchy of concepts, the processes repeated iteratively. The output is taxonomy of terms in which can be used as a base for finding more advanced ontological relations between concepts, or it can be used to guide a search for information or a classification process from a document corpus.

4.2.2.1.2 Natural Language Processing Approach

Ontology learning from text is the process of identifying terms, concepts, relations and dependencies among a set of words and using them to construct and maintain an ontology. Many techniques use natural language processing in the development of ontology learning systems. To find the dependency relation between two words Sabou et al. [27] used a set of syntactic patterns. Mainly, a specialized form of natural language called syntactic regularities which are inherent from the sublanguage nature of web service documentations. The ontology extraction steps are: dependency parsing, syntactic patterns, ontology building and ontology pruning. After the dependency parsing, they set three syntactic patterns categories to identify and extract interesting information from a corpus for ontology building. First pattern is used for identifying domain concepts for that they used the "Noun", for the second pattern they used the "Verb" to identify the functionalities. The last pattern is used for identifying relations using the prepositional phrases. Next, the ontology building step collects the results of the previous syntactic pattern-based extraction. The extracted terms are used for building the domain ontology.

4.2.2.1.3 Integrated Approach

The underlying concept behind the integrated approach is to develop a system with library of algorithms that allow the users to select appropriate learning algorithms for the kind of ontology they want to learn. TextToOnto [17] for instance, is a framework that use different measures to extract terms from the corpus and wide range of algorithms for different ontology learning techniques.

4.2.2.2 Learning from Semi Structured and Structured Data

Techniques like data mining and web content mining are the most commonly used for this learning method. Karoui et al. [31] proposed a method that used the structure of Web pages to extract domain ontology without using a priori knowledge. The approach builds a contextual hierarchy from the web page structure. Next, define the more relevant terms to classify using data preprocessing techniques. Based on the term position in the conceptual hierarchy a weight is added, then the candidate terms are classified automatically, and the concepts are extracted.

Another work, by Davulcu et al. [32] converts the structure of an HTML Web page into a hierarchical semantic structure (as XML) in order to mine it for generating a taxonomy. Web pages are mined to separate important concepts from instances as well as to establish parent-child relationships among the concepts and use that concepts for ontology building [32].

4.2.3 Ontology Matching

Ontology matching or alignment is the process of solving the mismatch problem that is caused by having different ontology representations of the same domain. Such a mismatching may cause unsatisfiable classes, properties and relations for the domain knowledge. Therefore, ontology matching helps in closing the gap between two or more mismatched ontologies by providing one single source of truth for the domain knowledge [53].

The input of the ontology matching process is the targeted ontologies that needs to be aligned and the output will be set possible mapping and correspondences between related entities (classes, properties and relations) of those ontologies [54].

Many ontology matching techniques and approaches have been developed in the last decade. However, choosing from this variety of techniques is far from being a trivial task. According to Otero-Cerdeira et al. [53], the classification of the matching techniques can be based on the interpretation of input information top-down or based on the type of the input bottom-up as it depicted in the Figure 9.

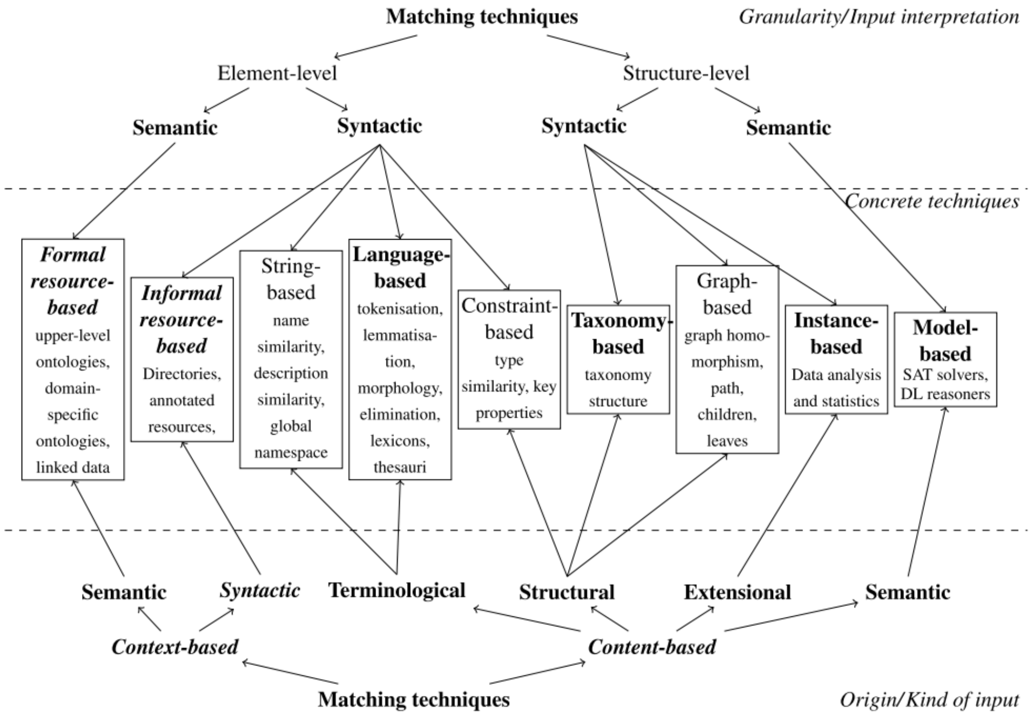

Figure 9. Matching techniques classification. Extracted from the book 'Ontology Matching' Euzenat et al. [55]

Starting with the top-down input interpretation, the matching techniques can be classified in a first level as:

Element-level matchers: This approach tackle the ontologies entities as an independent element, and not as a part of the whole ontology structure.

Structure-level matchers: for the matching, this technique mainly depends on entities structure of the ontology.

At the second level, those two techniques It also comprise the following sub techniques:

Syntactic: focus on fixing the mismatching results when two ontologies are modelled by using different knowledge representation formalisms, for instance, OWL and F-logic.

Semantic: mainly concern resolving the differences between two ontologies that use of different axioms for defining concepts or due to the use of totally different concepts.

Regarding the bottom-up classification, the first level of the matching techniques can be classified into the following categories:

Content-based: these techniques depend on the content of the matched ontologies. This technique comprises four sub techniques categories as the following:

- *Terminological:* focus on matching the variations in names when referring to the same entities in different ontologies. This may be caused by the use of different natural languages.
- *Structural:* match the ontologies entities (entities (classes, properties and relations) based on their structure in the ontology.
- *Extensional:* focus on matching the instances of the two compared ontologies.
- *Semantic:* deductive methods, use some semantic interpretation of the input and usually use a reasoner to deduce the correspondences.
- *Context-based:* these techniques used external sources to find a common ground in term of context between the two compared ontologies.

4.2.4 Business Process Model and Notation (BPMN)

A business process is a network of connected activities and buffers with well-defined boundaries and precedence relationships, which utilize resources to transform inputs into outputs for the purpose of satisfying customer requirements [74].

Through our thesis the business process is represented by a business process model and a notation (BPMN). Weske et al. [75] defined BPMN as a standard for business process modeling that provides flowcharting technique tailored for creating graphical models of business process operations. The BPMN notation consists of four categories of elements: the workflow objects (Events, Tasks, and Decisions); connection objects (streams and Sequence Messages and Associations); swim lanes (pools and lanes) and artifacts (Data objects, Annotations and Groups) [59].

Dumas et al. [57] decompose business processes into elements as depicted in Figure 10.

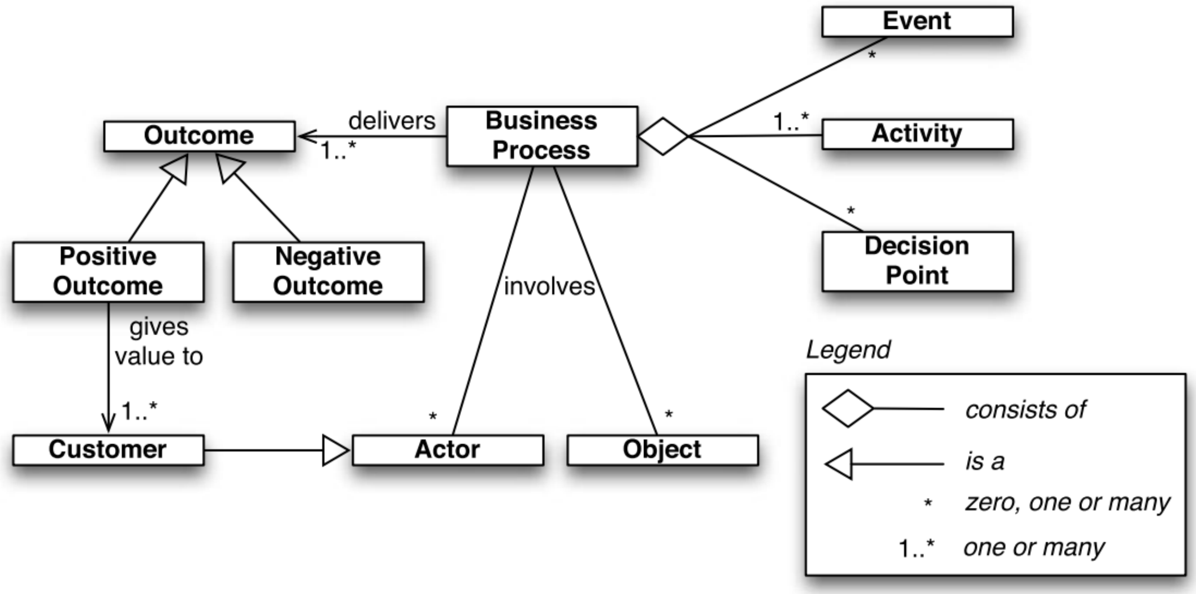

Figure 10. Ingredients of a business process, extracted from Fundamentals of Business Process Management [57]

- *Event*: represents something that happens and triggers a series of activities.
- *Activity*: represent work completed by a group or organization, some activities are atomic (a task) while others are not atomic (process and sub-process) since they can be further decomposed.
- *Decision point:* is a point at which a decision is taken that affects the way the process is executed.
- *Actor*: is someone or something that performs an "activity" or benefits from the output of a process.
- *Object*: can be it physical or immaterial, is a thing consumed or produced by an "activity".
- A process results in "outcome" which can be desirable ("positive outcome") or undesirable ("negative outcome").

BPMN is rich in modeling constructs for representing various types of control flow and events. As a result, BPMN has a high degree of expressiveness, but at the same time is highly complex [62]. Also, BPMN provide a unified notation for both IT and management stakeholders [59]. For this purpose, BPMN includes a basic set of constructs called the "Business Process Diagram (BPD) Core Element Set" (Core Set).

4.3 Use of Domain Knowledge in RE:

4.3.1 Use of Ontologies in RE

A remarkable systematic literature review based on 67 studies has been done by Dermeval et al. [52] as shown in Table 3. Such study provides a good understanding about the using of ontology to support the requirements engineering process and the current application for ontologies in RE filed.

The main finding of the study looks promising, according to Dermeval et al. [52], the main phases of the RE process that have been supported by the use of ontologies is Specification (83.6 %), followed by Analysis and Negotiation (58.2 %), Management (35.8 %), Elicitation (25.4 %) and Validation (6 %). While the most requirements modeling styles that used in combination with ontologies are textual requirements followed by UML, Scenario-based and Goal-oriented.

It is also noteworthy that most of the studies focuses on functional requirements, while little attention has been paid to non-functional requirements. The dominant approaches of ontology-driven RE tackle the ambiguity, inconsistency, incompleteness requirements problem followed by requirements management and evolution problem [52]. According to the authors classification criteria a study could have met more than one phase of the RE process, thus the sum of percentages can be greater than 100 %.

# RE problem	Count	Percentage%
Ambiguity, inconsistency and/or incompleteness	38	56.72
Requirements management/evolution	24	35.82
Domain knowledge representation	18	26.87
Integration between requirements and architecture	3	4.48
Requirements communication	2	2.99
Requirements models interoperability	2	2.99
Distributed requirements elicitation	2	2.99
Goal decomposition	1	1.49
Selection of elicitation technique	1	1.49

Table 3. Existing contributions in ontology-driven RE, extracted from "Applications of ontologies in requirements engineering: a systematic review of the literature" Dermeval et al. [52]

OWL (Web Ontology Language), is the most popular ontology-related language that have been used to support RE activities, while few studies used SPARQL, SWRL, UML and XML.

Only 37.3% (N=25) of the studies provide empirical evaluation that positively support the benefits of using ontologies in requirements engineering [52].

4.3.2 Using BPMN in Requirements Engineering

Requirements engineering (RE) is concerned with eliciting and managing requirements for the life cycle of software systems products. Business processes can be used to elicit and understand software system requirements due to the knowledge they contain [60].

Keeping software systems aligned with business processes is fundamental for companies to remain competitive nowadays. In literature it is noteworthy that, the employment of the modeling of business processes through the notation BPMN, can support the elicitation of requirements. By having a business process model, it is possible to map not only the workflow, but also a series of information related to the activities and identify existing informational systems or even requirements for the construction of new systems [61].

Cardoso et al. [66] proposed a business process-based model for requirements engineering and found that modeling business processes is a common practice in the RE field which helps stakeholders to understand their own business process, facilitates problem understanding, and reveals how the system will meet the needs of the process.

Mathisen et al. [67] use business modelling as an early stage tool that helps capture changes, in this way, develop software systems that fit the customer’s organization and business processes and to ensure that the software is aligned with business strategy.

Also, Mathisen et al. [67] argue that, some of the problems related to missing, incomplete or vague requirements can be referred to an insufficient understanding of the proposed software systems from the business point of view. Furthermore, the development of business process models, prior to, or in early phases of development will reduce the number of high-impact changes the system must go through during the system development life cycle. And for that a sufficient transformation of business requirements into a formal system specification is required and it consider a crucial step in any business-related software development project. In fact, business process models have proven to be an effective means of specification.

4.4 Summary on How the Current Literature Informed Our Research

The aforementioned literature review provides valuable insights about the related topics which are concerned with the requirements engineering and aims to outline the potential use of domain ontologies in requirements refinement.

According to Agile RE, user stories capture the system features from an end-user perspective and for that it describes the type of users, what they want and why they want it. Since the user stories are represented in natural language, literature review has been performed to learn more about the use of (NLP) in requirements engineering.

We gained a better understanding about the significance of Requirement modeling activity in terms of maintaining consistency and completeness of the requirements. The use of ontologies in RE is the core concept of our approach. Because of that, we evaluated the current state of the literature reviewed and we identified the gaps in existing knowledge. The gaps stemmed from focusing on dimensions other than specification dimension like representation dimension, from tools that have lack of interaction with the stakeholders or from manual approaches. Our first approach however aims to develop interactive semi-automated tool for requirements refinement.

Literature about building and learning ontologies approaches helped us in finding the best tool to build an ontology and the learning techniques that fit our approach. Applications of ontology matching approaches demonstrated outstanding results for refining requirements, as our first approach depends on the matching between two inputs: The predefined domain ontology 1 and the output of the Visual narrator tool which is considered as ontology 2.

The main goal of the business process modeling is to provide common language for communities of software and business engineers. As stated by Giaglis et al. [69] software that supports the business must be aligned with the business processes.

Nowadays, business is closely tied to the application of appropriate software systems and its usage, business process management brings the business perspective and the IT infrastructure together. However, the concrete specification of usage or transformation of business process models into the software models is still relatively limited [68]. To address these limitations, we investigated the potential use of BPMN in refining requirements engineering user stories.

Chapter 5

5. Using Domain Ontology to Refine User Stories.

We will discuss different scenarios that illustrate a co-evolution process of a domain ontology and a collection of requirements expressed as user stories. Such co-evolution process outlines the key idea of this process: how ontologies can be used to refine user stories. This process shown in Figure 11.

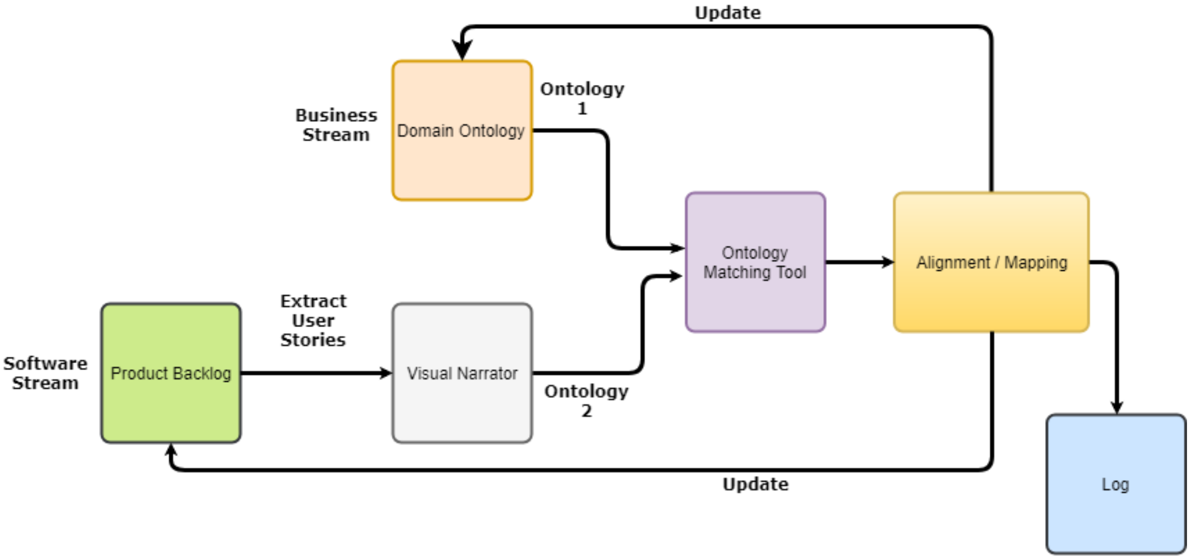

Figure 11. User stories refinement process

Consider an existing product backlog, which lists the features that the product owner desires for the final product. Product backlogs contain user stories that are sorted from higher to lower priority and may also include acceptance criteria that articulate precisely when the user stories are done [33].

Second, suppose the product owner disposes of a domain ontology that describes individuals (instances), classes (concepts), attributes, and relationship between the instances and the concepts.

The refinement process starts by taking the highest priority set of the user story from the product backlog. The first set of the user story will be processed by the visual narrator (VN) that automatically extract a conceptual model from a set of user story requirements [35].

The output from the VN can be considered as ontology, and then it will be checked against the domain ontology using an ontology matching tool. For example, we may use AgreementMaker system [34] for matching schemas and ontologies. In order to handle many different matching scenarios, AgreementMaker uses a wide range of iterative matching methods. The tool helps in making alignment between the two schemas or ontologies by depicting new mapping between two concepts or adding a new relations, concepts and instances.

The matching process can be done automatically or manually by expert intervention. However, in order to do the comparison, we have to assign the source and the target ontology that we want to compare. First, we will set the domain ontology as the source and the output from VN as the target for refinement. Next, we will do it the other way around, so the VN output will be the source and the domain ontology will be the target.

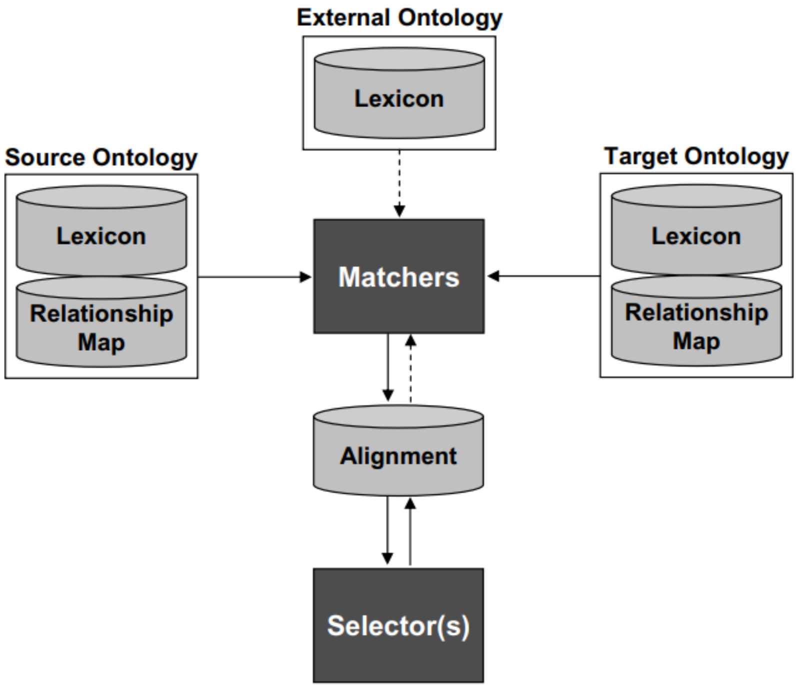

Figure 12. Schema of the AgreementMakerLight Ontology Matching Module, extracted from The agreementmakerlight ontology matching system [76].

Figure 12. Shows the schema of the AgreementMaker ontology matching model. Where Lexicon contains the local names of all listed classes, their labels, and all their synonyms. Relationship Map contains the ”is a” and part of relationships between all listed classes. Matchers are algorithms that compares two ontologies and return an Alignment between them. Alignment is a data structure used by the ontology matching module to store mappings between the input ontologies. Selectors are algorithms used to trim an Alignment by excluding mappings below a given similarity threshold [76].

In our example, the underlying idea about the matching process is to develop both the user stories and the domain ontology simultaneously. Adding, updating and deleting a new classes, properties and relations using the matching tool will reduce the mismatching between the user story and the domain ontology. Hence, co-evolution of refining the user stories and building the domain ontology would be possible as a result of the matching process.

5.1 Scenario for Using Domain Ontology to Refine User Stories:

Assume that we have the following scenario: we are developing a system that supports the university education system, where the students can submit their thesis online, professors can check the students thesis and make notes, comment and eventually grade the uploaded document. The professor assistant also plays a role in terms of following up the students work, managing and keep tracking of the students deliverables.

Some suggested user stories could be as the following:

S1: As a student, I want to upload my academic paper to the system, so that I can hand it in to my supervisor.

S2: As a student, I want to edit my paper, so that I can make the required modification

S3: As a professor, I want to read the uploaded paper, so that I can grade it

S4: As a supervisor, I want to edit the paper, so that I can make a notes

S5: As an Assistant, I want to read the paper, so that I can follow up the changes

S6: As an Assistant, I want to edit the paper, so that I can give my feedback

S7: As a student, I want to update my grade, so that I can improve my GPA

The output of visual narrator has many formats as it mentioned previously one as shown in Table 4:

ID Number	**Subject**	**Predicate**	**Object**	**Occurs in**
S1	Student	Upload	Paper	1
S2	Student	Hand To	Supervisor	1
S3	Student	Edit	Paper	2
S4	Student	Make	Modification	2
S5	Professor	Read	Paper	3
S6	Professor	Grade	-pron-	3
S7	Supervisor	Edit	Paper	4
S8	Supervisor	Make	Note	4
S9	Assistant	Read	Paper	5
S10	Assistant	Follow	Change	5
S11	Assistant	Edit	Paper	6
S12	Assistant	Give	Feedback	6
S13	Student	Update	Grade	7
S14	Student	Improve	GPA	7

Table 4. Report of user story parsing, and conceptual model creation

After processing the output from the visual narrator and the domain ontology, we can find the following cases:

- Two terms refer to the same concept. For instance, the user stories S6 and S7 the concept supervisor and professor handled separately while in the domain ontology it is explicitly mentioned that the two concepts are actually the same as it shown in the ontology concepts synonym. So, the analyst can merge the two concepts in the user stories. Likewise, for the two individual (master thesis -graduation project) and (Lecturer- Professor). As it shown in the Figure 13.

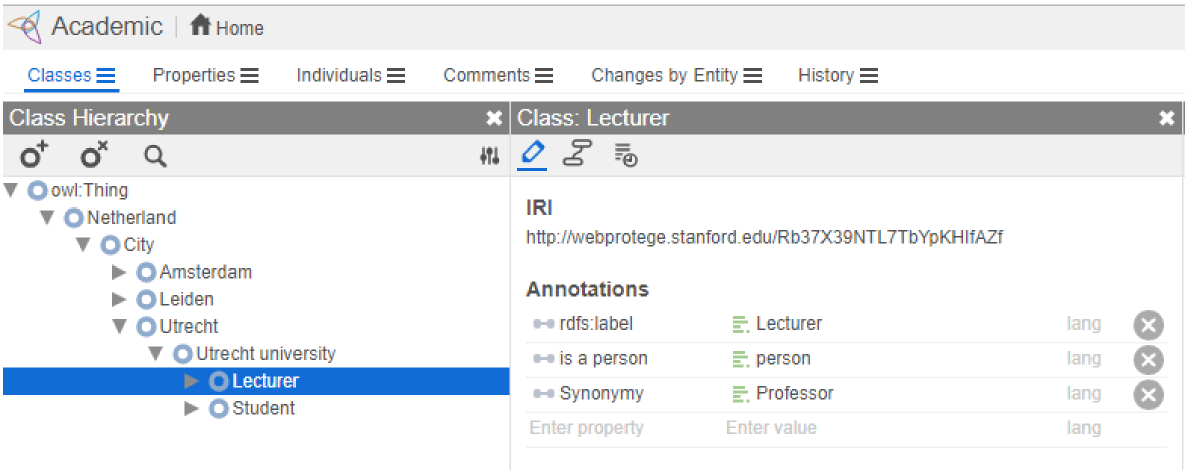

Figure 13. Class annotation for the lecturer concept

- From the report of user story parsing Table 4, S6 shows an undefined object and that lead to ambiguity in the user story S3 , while at the domain ontology it is clear that the concept professor has an object property "canGrade" the academic paper. So, the analyst can modify the user story number 6 to become the Refine S3: As a professor, I want to read the uploaded academic paper so that, I can grade the academic paper as it shown in the Figure 14.

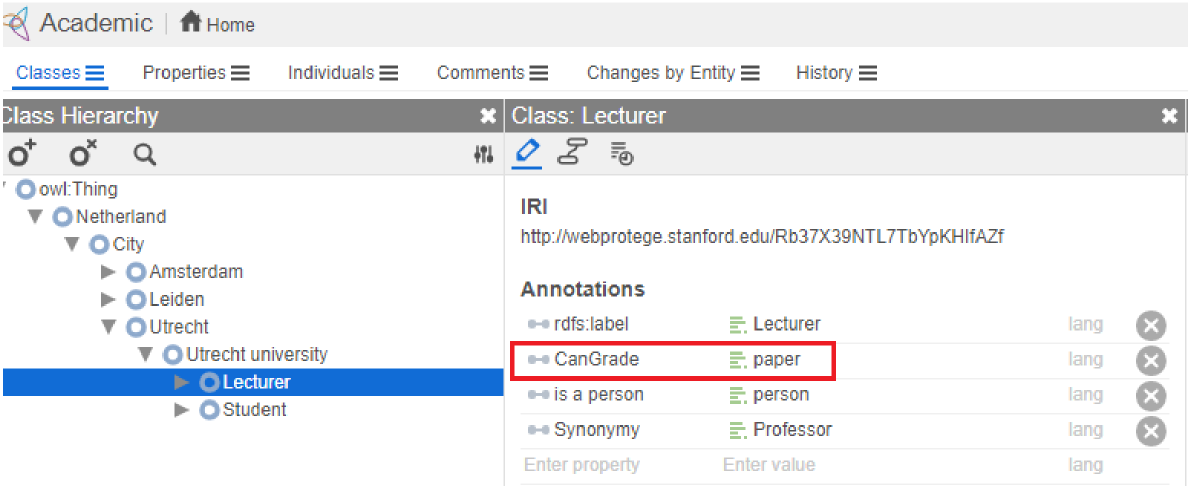

Figure 14. Property assertion for the concept lecturer

- The user story number S7 suggest a new object property that not exist in the domain ontology "a student can update his own grade" for this case a distinction between open word assumption and close word assumption is needed. The Closed World Assumption (CWA) is the assumption that what is not known to be true must be false. On the contrary, the Open World Assumption (OWA) it is the assumption that what is not known to be true is simply unknown. If the analyst considers the concept under the (CWA) then the new concept should be deleted from the user stories. Otherwise, a new concept should be added to the user stories [36].

- An ontology considered as a vague if it has at least a vague definition of a concept [37]. From our example the domain ontology shows that the assistant has a property to "help" the professor such a vague property might be confusing for the analyst. While the user stories S5 and S6 shows that the assistant can "read "and "edit" the uploaded academic paper. The analyst then has to add the two new properties to the domain ontology as an object property for the assistant. So that the individual Max which has a class of assistant, can read and edit the students academic paper.

- Ontology Log: At the same time, in order to keep track of all the changes and the reasoning behind it and to avoid any confusion in the future, all the previous changes for both the ontology and the user stories have been registered and documented in log by the matching tool.

After all the seven user stories have been refined and stored in the product backlog, a new set of user stories processed again in the same manner. Repeat the same process recursively until all the user stories in the backlog have been refined.

Although the previous approach looks theoretically promising, according to the expert opinion it might be not a practical solution for the following reasons:

- Nowadays, just few companies use domain ontology to represent their knowledge.
- Due to the structure of the domain ontology, in practice, it is difficult to extract user stories from it, since most of the relations are (is-a, has-a) relations which are not necessarily helpful in creating new user stories.
- Domain ontology mainly focuses on information structure instead of process structure. Thus, it does not contain activities, tasks or workflows that help in describing processes. Processes are useful because they support the analysts in understanding the domain area.

For the previously mentioned reasons, there was a need for another domain knowledge. For that, we investigated the use of business processes as domain knowledge since business processes are widely used by companies and they support modeling of different types of activities, tasks and workflows. Weske et al. [75] defines business process as a set of activities that are performed in coordination in an organizational and technical environment. These activities jointly realize a business goal.

Chapter 6

6.1 Using BPMN to Refine User Stories

We present the potential use of a BPMN-based business process for refining user stories in a specific business domain.

The process shown in Figure 15 has two input streams: the first one is the software stream represented by the product backlog which contains a set of user stories, and the second is the business stream represented by the BPMN diagram.

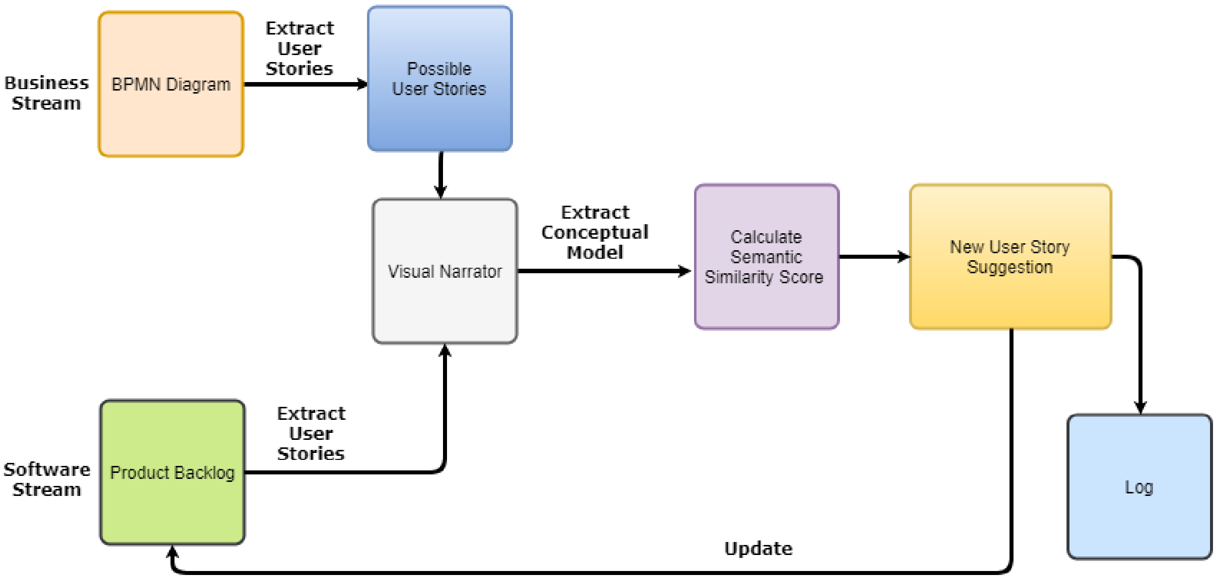

Figure 15. RE user story refinement using BPMN diagram

The first input, the user stories from the Product Backlog, is directly processed via the VN tool (visual narrator tool). VN tool extracts a conceptual model from the user stories and as a result the triple (Subject1, Predicate1, Object1) is generated.

The second input - BPMN diagram - is processed in two phases:

In the first phase, all possible user stories from the BPMN diagram are extracted, however the focus was solely on the pool/lane and on the task. The extraction is carried out as the following:

The Connextra template is used - Lucassen et.al. [41]:

Template: "As a <type of user>, I want <goal>, so that <some reason>"

The <type of user> is replaced by <pool / lane> and the <goal> is replaced by the <task>

Thus, the following new user story template based on BPMN diagram is generated:

"As a <pool / lane>, I want <Task>".

At the end of the first phase a set of possible user stories are extracted from the BPMN diagram. Then, the user stories are processed via VN to extract the conceptual model and the triple output of the second phase becomes: (Subject 2, Predicate 2, Object 2).

Next, the two tuples (Subject, Predicate, Object) from the Product backlog and BPMN are compared against each other. The source is the triple set from the BPMN diagram and the destination is the triple set from the product backlog as shown in Figure 16

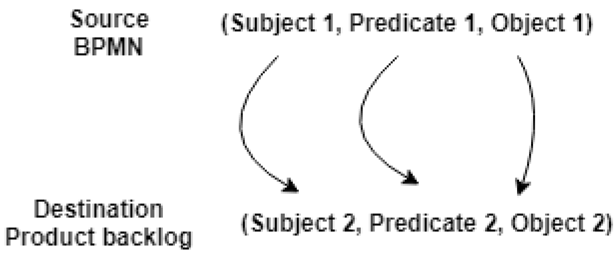

Figure 16. Comparison between elements from source and the corresponding destination tuples

Next, three thresholds T, T2 and T3 are specified for each element of the tuple: subject, predicate and object, respectively. The process starts by computing the semantic similarity of the subjects from the source and destination. If the similarity score is above T1, we compute the semantic similarity of the predicates, otherwise suggest a whole new user story based on the source. Likewise, we compute the semantic similarity of the two predicates. If the similarity score is above T2, we compute the semantic similarity of the objects, otherwise suggest a new user story from the source and continue to the objects and calculate their semantic similarity. If the similarity score is above T3, finish the current process and pick the next user story from the product backlog, otherwise suggest a new user story from the source. This is depicted in the flowchart Figure 17:

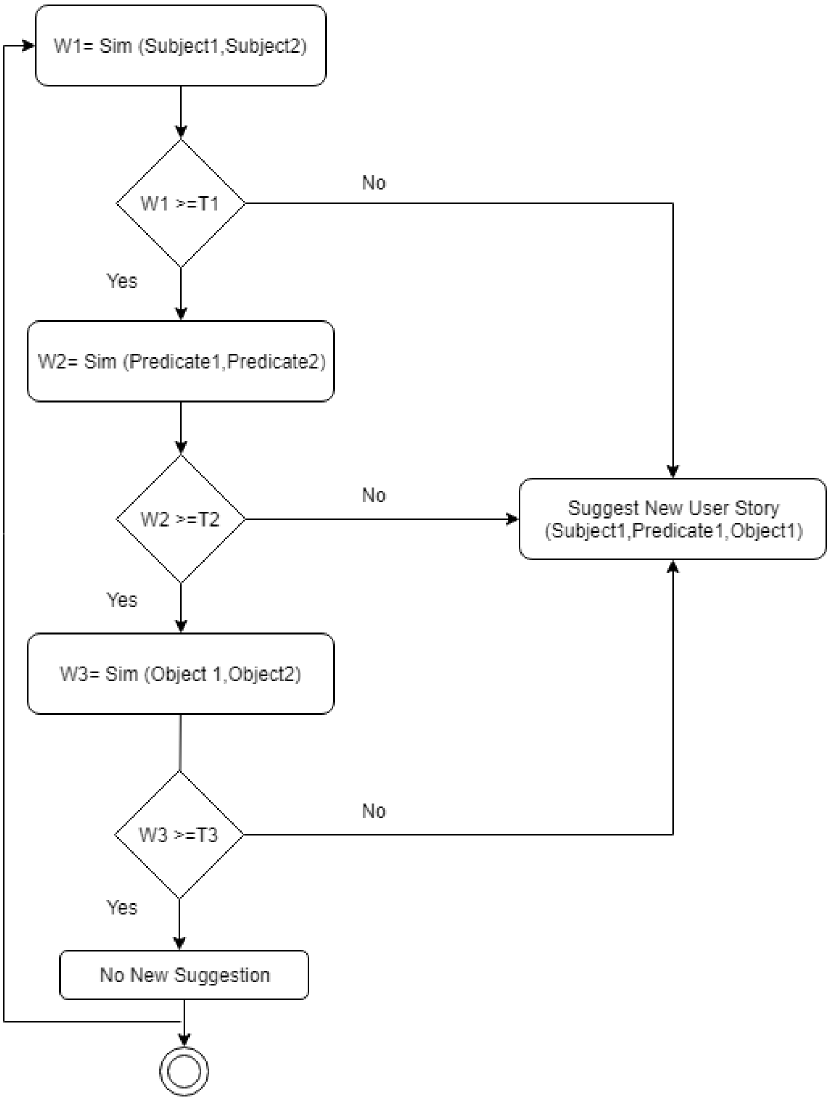

Figure 17. Story Suggestor Flowchart

6.2 Scenario:

Suppose that we are developing a system that supports the university education system, where the students can submit their thesis online, professors can check the students thesis and make notes, comment and eventually grade the uploaded paper. The professor assistant also plays a role in terms of following up the students work and managing and keeping track of the student's deliverables. The BPMN diagram for the university education system is shown in Figure 18.

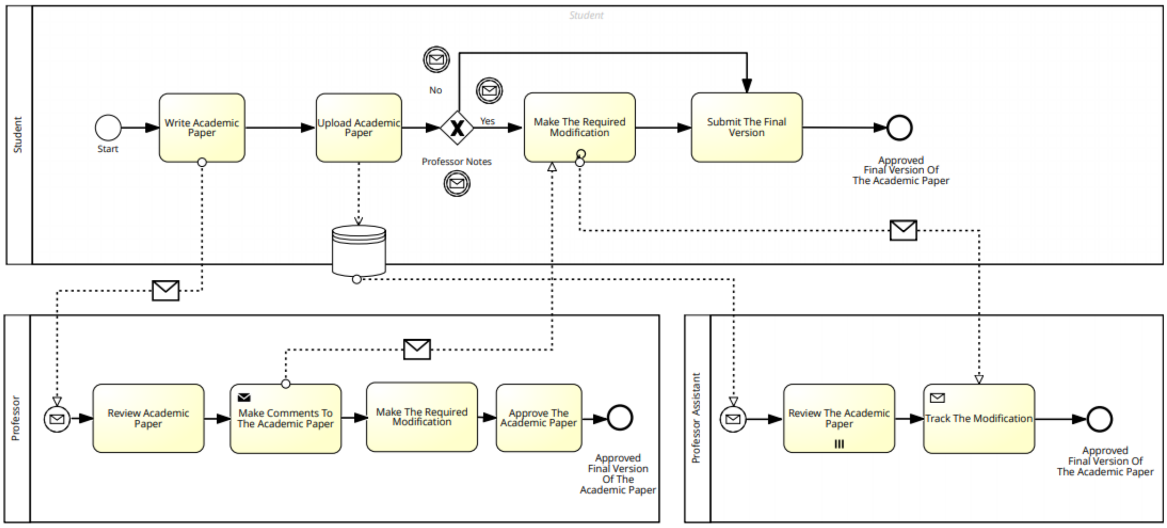

Figure 18. BPMN diagram for the university education system

As aforementioned, first all user stories from the BPMN diagram need to be extracted. The extraction is done using the following template:

“As a <pool / lane>, I want <Task>”.

The Pool / lane and the corresponded task is shown in Table 5:

Pool/ Lane	**Task**
Student	Write academic paper
	Upload academic paper
	modify the academic paper
	Submit the final version
Professor	Review the academic paper
	make comments to the academic paper
	Grade the academic paper
	Approve the academic paper
Professor assistant	Review the academic paper
	follow up the students modifications

Table 5. Pool / lane and the corresponded task, as it extracted from the BPMN diagram

The extracted user stories are as follows:

S1: As a student, I want to Write academic paper

S2: As a student, I want to Upload academic paper

S3: As a student, I want to modify the academic paper

S4: As a student, I want to Submit the final version of the academic paper

S5: As a professor, I want to Review the academic paper

S6: As a professor, I want to make comments to the academic paper

S7: As a professor, I want to Grade the academic paper

S8: As a professor, I want to Approve the academic paper

S9: As a professor assistant, I want to Review the academic paper

S10: As a professor assistant, I want to follow up the student's modifications

Suppose the product backlog has the following set of user stories:

S11: As a Student, I want to write my academic paper

S12: As a student, I want to submit my thesis to the system so, that I can hand it in to my supervisor.

S13: As a student, I want to edit my paper so that I can make the required modification

S14: As a professor, I want to read the uploaded paper so that I can grade it

S15: As a professor, I want to consent to the uploaded paper so that I can grade it

S16: As a supervisor, I want to edit the paper so that I can make a notes

S17: As an Assistant, I want to read the paper so that so that I can follow up the changes

S18: As an Assistant, I want to edit the paper so that so that I can give my feedback

S19: As an Assistant, I want to Check the paper so that so that I can give my feedback.

To extract the conceptual model (Subject, Predicate, Object), first both user stories from BPMN and Product backlog are processed via the Visual Narrator tool.

The outputs are shown in Table 6 and Table 7.

Subject	**Predicate**	**Object**
Student	Write	Paper
Student	Upload	Paper
Student	Modify	Paper
Student	Submit	Version
Professor	Review	Paper
Professor	Make	Comment
Professor	Grade	Paper
Professor	Approve	Paper
Professor Assistant	Review	Paper
Professor Assistant	Follow up	Paper

Table 6. output of VN after processing the user stories from the BPMN diagram

The following table shows the output of VN after processing user stories from the product backlog

Subject	**Predicate**	**Object**
Student	Write	Paper
Student	Submit	thesis
Student	Hand To	Supervisor
Student	Edit	Paper
Professor	Read	Paper
Professor	Grade	Paper
Professor	Consent To	Paper
Supervisor	Edit	Paper
Assistant	Read	Paper
Assistant	Check	Paper

Table 7. output of VN after processing user stories from the product backlog

After that, the semantic similarity for the two tuples is calculated. For example:

(student, submit, thesis) vs (student, upload, paper)

Table 8. shows the semantic similarity score for the previous example

		BPMN		
		Student	Upload	Paper
Product Backlog	Student	1		
	Submit		0.5	
	Thesis			0.4

Table 8. semantic similarity score matrix for both tuples from BPMN and product backlog

Next, three thresholds T, T2 and T3 are specified for each element of the tuple: subject, predicate and object, respectively. The process starts by computing the semantic similarity for the subjects from the source and destination and comparing it to the threshold T1=1.

From the given example:

Sim (student, student) = 1 >= T1 is True.

Next, the semantic similarity for the predicates is computed and compared with the threshold T2= 0.7

From the given example:

Sim (submit, upload) = 0.5 >= 0.7 is False. Therefore, a new user story based on our source triple- BPMN triple- is suggested as the following:

As a student, I want to upload a paper.

The requirements analyst has the option to accept or reject the suggested user story. Then, start processing the next user story from the product backlog and repeat the same process iteratively until all the user stories from the product backlog are refined.

Chapter 7

7. Validation and Experiment Design

In the previous chapters, we proposed techniques for using the BPMN as domain knowledge to refine requirements engineering in agile methods. In this chapter, a validation for our proposal is presented by conducting an experiment using our Story Suggestor Tool.

7.1 Goal

The goal of the experiment is to compare the quality of the requirements specification when writing a collection of user stories that represent the requirements for a given system.

We compare two treatments: i). the participants left to freely define their own method and, ii). the participants using the Story Suggestor tool.

Our purpose is to evaluate the benefits and drawbacks of using the Story Suggestor tool from a junior analyst point of view in the context of Utrecht University master students.

7.2 Experimental Subjects

The subjects participating in the experiment are MBI master students from Utrecht University. The experiment uses a balanced design, which means that there is the same number of subjects in each group, and for that we asked 10 students to participate in the experiment: 5 students form the control group, and 5 students form the treatment group. All subjects had previous knowledge in agile software development and user stories and all subjects voluntarily participated in the experiment.

7.3 Response Variables and Metrics

The following variables are considered in our research:

A. Independent variables (IV):

Represented by the treatment: whether the participants used their own self-defined method, or they used the Story Suggestor tool when writing a set of user stories. A nominal scale is used to measure this variable.

B. Dependent variables (DV):

Represented by the quality of a requirements specification

- Completeness
- Correctness

Ordinal scale is used to measure these variables using the range: (most relevant, less relevant, irrelevant)

7.4 Experimental Questions

Through this experiment, we wanted to address the following questions which are related to the objective of this thesis:

MRQ: Does the use of the semi-automated process for supporting requirements refinement lead to higher quality requirements compared to the traditional backlog refinment?

Based on our main research question, two sub-questions have been formulated as the following:

RQ1.2: Do analysts who use domain knowledge deliver more complete requirements than analysts who use the traditional backlog refinement?

RQ1.3: Do analysts who use domain knowledge deliver more correct requirements than analysts who use the traditional product backlog refinement?

The research questions above have been translated into the following hypothesis:

H01: Using domain knowledge as refinement for requirements engineer increases the requirements quality.

We mainly focused on correctness and completeness as two criteria to measure the quality of the participant user stories.

In Figure 19, our GQM (Goal/Question/Metric) model (which provides a graphical visualization of how the metrics and the questions refer to the goal) is provided. The interpretation of the model is as follow:

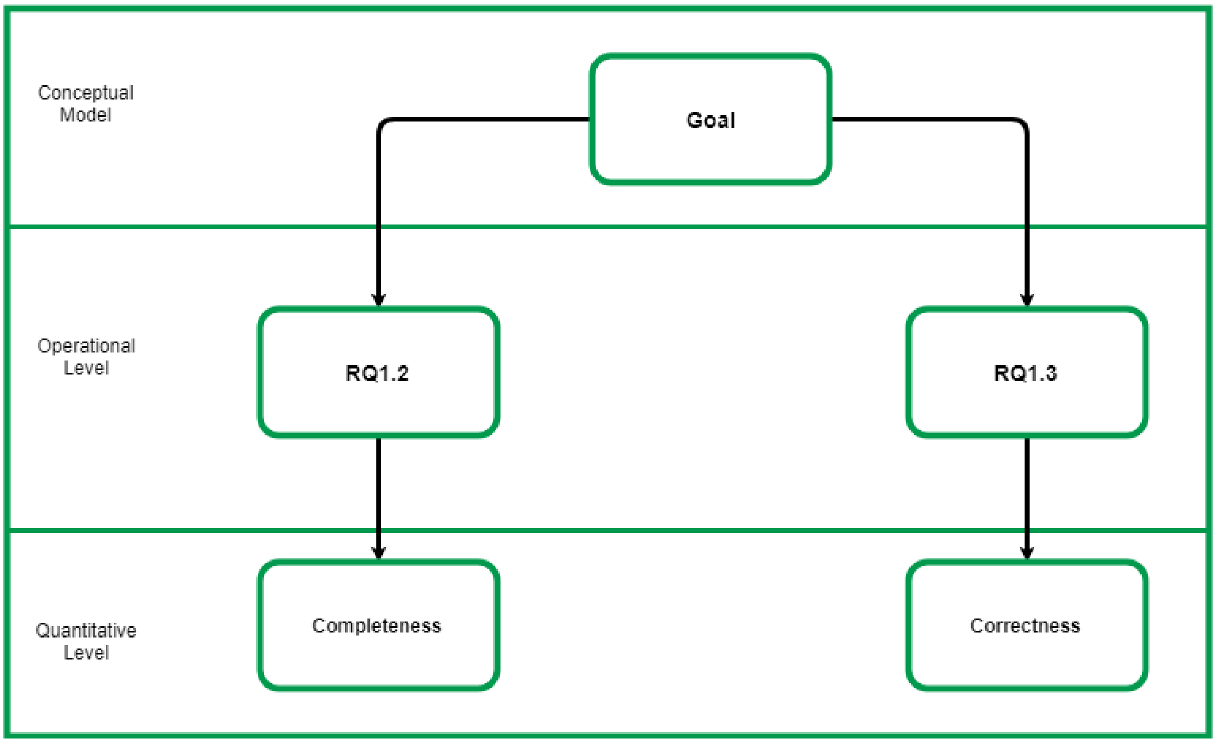

Figure 19. GQM Model

- The goal, placed in the conceptual level, refers to the research goal itself.
- Research Questions RQ1.2 and RQ1.3, in the operational level, are the research questions. The answers to those questions will determine the achievement of the goal.
- Last, located in the quantitative level, the 'Metric' entity is represented in our research by the quality of requirements specification.

The metrics will be measured based on the information retrieval context as the following:

$$Completeness = \frac{Number\ of\ relevant\ requirements\ identified\ by\ subject\ S}{Total\ number\ of\ relevant\ requirements\ identified\ by\ all\ subjects}$$

$$Correctness = \frac{Number\ of\ relevant\ requirements\ identified\ by\ subject\ S}{Number\ of\ identified\ requirements\ identified\ by\ subject\ S}$$

Table 9 shows the research questions related to the hypothesis, metrics and the scale that were used to conduct the experiment.

RQ	Hypothesis	Response variable	Scale
RQ 1.2	H01	Completeness	Ordinal scale
RQ 1.3	H01	Correctness	Ordinal scale

Table 9. Research questions related to hypothesis, variables and metrics

7.5 Experimental Design

The design is divided into two groups using different subjects. In the control group (Group A) the subjects were asked to use their own criteria in writing the user stories. While, in the treatment group (Group B) the subjects were asked to write user stories assisted by the Story Suggestor tool. Next, in order to evaluate the quality of the user stories, a comparison of the results from the two groups in terms of (Correctness and Completeness) is performed. Finally, evaluation and analysis of the results and hypothesis testing are conducted.

7.6 Context

The experiment is planned to be conducted at the premises of Utrecht university. The university provides a wide range of available rooms and study places. However, due to availability of the subjects, the experiment was conducted individually with each subject performing the experiment online from home. The idea was to let the participants choose their ideal working area that helps them focus on the task without any distraction.

7.7 Instrumentation

The subjects were provided with several instruments to execute the experiment. Both groups were asked to use their own laptops to write the user stories.

Also, one of the instruments was the Connextra user story guidelines (see Appendix B for the Connextra guidelines). The guidelines helped the participants in the writing user stories according to the agile Connextra template. The guidelines were also supported by an example which assisted the subjects in obtaining complete user stories.

Another instrument was the description document (experiment handout) of the system that needed to be developed (see Appendix B for the experiment handout). The document provided a complete description of the system's main functionality and system's stakeholders and their roles. The document was used to aid the subjects in getting better understanding about the features and the behavior of the system and it assisted the subjects in defining the intended functionality that satisfied the needs of different stakeholders. Also, the document helped to define at a high-level the main business processes that the system supported.

Furthermore, for the treatment group a tutorial video (see Appendix B for video tutorial link) was made that contained a step-by-step explanation about the use of the Story Suggestor tool

7.8 Experimental Procedure

The experiment took one week, the execution procedure of the experiment can be seen in Figure 20. For the control group A, the subjects were asked to write 20 user stories each. The subjects were provided with the previously mentioned instruments apart from the additional user stories from the Story Suggestor tool and the tutorial video. For the control group A, the subjects were prompted to use their own criteria in writing the user stories

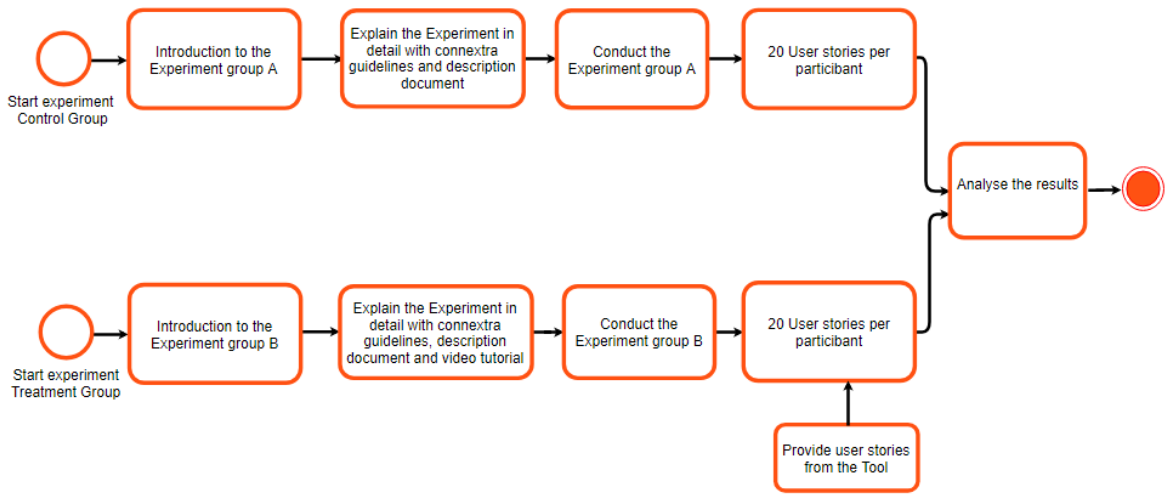

Figure 20. Experimental procedure

The procedure for the treatment group B is very similar to the first one, the only difference was that the subjects were provided with extra instruments, namely the additional user stories that were created by using the Story Suggestor tool and the tutorial video.

After both groups finished writing the user stories, the output of each subject was collected and evaluated.

7.9 Sampling

The experiment was conducted with 10 subjects. Some of the subjects were MBI students at Utrecht University, others were computers engineers who have experience in requirements engineering projects. All subjects voluntarily participated in the experiment. The subjects were assigned to the experiment groups randomly. As aforementioned, the experiment lasted for one week and the subjects participated according to their availability.

7.10 Preparation

Before starting the experiment, a short phone introduction was made for each subject in both groups to explain the experiment task and the system that needed to be developed to ensure that the participants had good understanding on how to perform the experiment. On top of that a brief introduction about the Story Suggestor tool was made to the subjects of the treatment group B.

7.11 Experiment Execution and Data Collection

7.11.1 Execution

The experiment was mainly executed digitally. For the control group A, the participants did the experiment using Google Document to write the user stories and to fill in the feedback form. While for the treatment (Group B), the subjects used the Google Document for the feedback form and Google Colaboratory in which the Story Suggestor tool was developed. The Google Colaboratory is a free cloud service environment that requires no setup and runs entirely in the cloud and it supports Python programming language.

Conducting the experiment in general went as planned. However, not all subjects were able to participate at the same time, so several time slots over one-week time were proposed by email. The communication of these arrangements was made mainly through email communication and mobile phones apps in order to receive fast replies.

For the control group (Group A), no substantive questions were asked. This can be explained by the fact that the subjects were meant to use their own criteria in writing the user stories. Whereas, for the treatment group (Group B) the subjects asked significantly more questions, mainly about the use of the Story Suggestor tool.

Some of the subjects had a hard time in extracting user stories out of the proposed developed system. When questions about the system description were asked, more explanation was provided. Few subjects have used different user story templates, in that case the subjects were referred to the appropriate Connextra template guideline.

In general, there were some deviations from the original experimental plan. The original plan was to conduct the experiment at the same day for all the subjects as previously explained. However, due to the availability of the subjects, the experiment was conducted over a one-week period. Apart from that, the experiment ran as planned.

7.11.2 Data Collection

For both groups of the experiment, data collection was conducted during the entire week. This choice was made because the subjects were not be able to participate at the same time. Once the subjects read and studied the material that was provided, they noted the time it took them to complete the experiment. In general, all participants reported spending approximately 30 minutes to complete the task.

Since all participants performed the experiment digitally using their laptops via Google Document and Google Colaboratory data collection was reasonably easy. For the control group (Group A), subjects wrote the user stories directly on the shared Google Document. While for the treatment group (Group B), the subjects wrote the user stories using the Story Suggestor tool which was built on the Google Colaboratory platform.

Moreover, all group B participants were asked to fill in a Feedback Form after they finished using the Story Suggestor tool.

All documents from the experiment were collected and sorted in accordance to the experimental group case. Next, the user stories from both groups were transferred to excel sheet.

After having all the data collected from the subjects, the analysis phase started. First, the data was organized and ordered per subject and per experimental group (control/treatment). Then a unique id number was given for each story. To identify the relevancy of each story, all stories were tagged according to the following criteria:

- *Most Relevant:* in case the user story is relevant to the process of the developed system and does meet the system objective. Moreover, it should be directly related to one of the system functionalities or linked to a statement in the system description as described in the experiment handout.

- *Less Relevant:* in case the user story is relevant to the context of the process and is useful to have in the developed system, but it is not related to the developed system's main functionality as described in the experiment handout.

- *Irrelevant:* in case the user story is not related to any of the system functionalities and does not meet the system objective or out of the scope of the developed system as described in the experiment handout.

For the aforementioned tagging we asked a domain expert for help. The expert was provided with a set of user stories sorted randomly, without any reference to the subject name or experimental group. Furthermore, the experiment handouts were provided, in order to help the expert to gain a comprehensive understanding of the experiment context, developed system and the tagging criteria.

The complete user stories, experiment video tutorial link and experiment handouts for both groups can be found in Appendix C.

Chapter 8

8.1 Results and Discussion

In order to answer the main research question and the two sub-research questions stated below, we conducted literature review, developed a Story Suggestor Tool using the design science methodology and conducted an experiment.

MRQ: "Does the use of the semi-automated process for supporting requirements refinement lead to higher quality requirements compared to the traditional backlog refinement method?"

RQ1.2: Do analysts who use domain knowledge deliver more complete requirements than analysts who use the traditional backlog refinement?

RQ1.3: Do analysts who use domain knowledge deliver more correct requirements than analysts who use the traditional product backlog refinement?

The research questions above have been translated into the following hypothesis:

H01: Using domain knowledge as refinement for requirements engineer increases the requirements quality.

Therefore, the answers of the sub-research questions lead to answer the main research question. Eventually, we can decide whether to accept or reject our hypothesis.

We calculated the precision, recall and F1 Score for both the control and the treatment groups by using their formulation for the information retrieval context.

In information retrieval, precision reflects the accuracy of the system and is represented by the fraction of retrieved documents that are relevant to the query. In that sense the correctness corresponds to the precision. The following formula is used to calculate precision[70]:

$$Correctness = \frac{Number\ of\ relevant\ requirements\ identified\ by\ subject\ S}{Number\ of\ identified\ requirements\ identified\ by\ subject\ S}$$

Instead, recall reflects the completeness of the results produced by the system and is represented by the fraction of the relevant documents that are successfully retrieved [70]. In that sense the completeness corresponds to the recall. The following formula is used to calculate recall:

$$Completeness = \frac{Number\ of\ relevant\ requirements\ identified\ by\ subject\ S}{\text{Total number of relevant requirements identified by all subjects}}$$

Both previous formulas applied to one subject S. To get more insight on how exactly our developed tool helped the subjects in term of (most relevant, less relevant, irrelevant). We calculated the precision and recall two times, first for both "most relevant and less relevant" and second, for just the "most relevant"

In case we combine both "most relevant" and "less relevant" as "Relevant" and anything else as "Irrelevant", the results are shown in Table 10:

Measurement	**Control Group**	**Treatment Group**
Precision	0.760	0.820
Recall	0.241	0.260
F1 Score	0.366	0.395

Table 10. Precision, Recall and F1 Score for Control Group and Treatment Group

The precision of the control group on this data set is 0.760, while the recall is 0.241. However, for the treatment group we got a slightly better precision 0.820 and relatively higher recall 0.260.

In case we focus solely on "most relevant", without considering "less relevant", the results are shown in Table 11:

Measurement	**Control Group**	**Treatment Group**
Precision	0.590	0.780
Recall	0.240	0.318
F1 Score	0.342	0.452

Table 11. Precision, Recall and F1 Score for Control Group and Treatment Group

In general, from the previous two tables we can find that the treatment group has higher precision, recall and F1 score. And that could be due to the fact that the treatment group get useful user stories suggestions using the Story Suggestor Tool.

- Answer to sub research question RQ1.2:

> ***RQ1.2:*** *Do analysts who use domain knowledge deliver more complete requirements than analysts who use the traditional backlog refinement?*

From Table 11, the recall of the control group on this data set is only 0.240, while the recall of the treatment group is 0.318. This answers our sub-research question RQ1.2: Analysts who use the domain knowledge seems to exhibit slightly more complete requirements than the analysts who use the traditional backlog refinement.

- Answer to sub research question RQ1.3:

> ***RQ1.3****: Do the analysts who use domain knowledge deliver more correct requirements than the analysts who use the traditional product backlog refinement?*

From Table 11, the precision of the treatment group is 0.780. Whereas, the precision of the control group is only 0.590. Therefore, the results answer our sub research question RQ1.3: Analysts who use domain knowledge seems to deliver relatively more correct requirements compared to analysts who use the traditional backlog refinement.

From the previous Table 11, we notice that the Story Suggestor Tool performs good in supporting the subjects in writing user stories that were directly related to the developed system functionality. In our case these user stories were tagged as “most relevant”.

Whereas, the tool provided little help in writing user stories that were related to the context of the developed system but not directly linked to one of the system main functionality. In our case these stories were tagged as “less relevant”.

This can be explained by the fact that the tool extracts the information from the BPMN diagram which covers only the main system functionality. On the other hand, our hypothesis stated that

> ***H01:*** *Using domain knowledge as refinement for requirements engineer increases the requirements quality.*

To test our hypothesis, an unpaired two tailed t-test has been conducted to compare the means of the two groups. The results of the t-test were as the following:

There was no significant difference in the scores of the control group (M=11.80, SD= 4.44) and of the treatment group (M=15.60, SD=1.34) conditions; $t = 1.8325$, $p = 0.1042$.

In general, the results are inconclusive. The results seem to confirm the hypothesis that the treatment helps (with a higher improvement for recall than for precision), but without statistical significance we cannot make a definitive statement. And even if we had statistical significance, the results are limited because of the small sample size as we collected data from 10 subjects only.

8.2 Validity Threats and Limitations

In this section, we discuss the external and internal validity of the research and its limitations.

8.2.1 External Validity:

The external validity refers to the extent the results of the study could be generalized at a large scale or to the extent they could be applied to other groups.

The artifact Story Suggestor Tool is developed to support the analyst to get higher quality requirement for all projects in agile development methods. The subjects of the experiment are analysts involved in requirements engineering projects and MBI students. However, only 10 subjects were able to participate in the experiment. Therefore, it is difficult to identify whether our experiment findings can be generalized due to the small sample size. It might be an option for future research to choose more diverse groups and bigger sample size for the experiment. Also, the Story Suggestor Tool was tested one time only using one experiment example: “Graduation process for MBI student at Utrecht university”. More examples in different domain areas are needed to fully test the capability of the tool.

8.2.2 Internal Validity

The internal validity is concerned with whether the treatment caused an outcome to occur, or whether there were other confounding factors that might have caused the outcome. All participants of the experiment were people with good understanding of requirements engineering in agile methods, either through courses or through being involved in requirements engineering projects. Although the subjects were assigned randomly to the experiment groups, many of them are MBI students who are already familiar with the experiment example: MBI graduation process. This might have affected the results of the experiment as the subjects might have been able to deliver high-quality requirements regardless whether or not they used the Story Suggestor Tool.

Adding to that, in our experiment example the: “The graduation process of MBI student at Utrecht university” at least 34 user stories are needed to cover the entire process and to support the proposed system main functionality, whereas subjects of both groups were asked to write only 20 user stories each. Hence the subjects could not cover the entire process using the experiment guidelines. The reason the subjects were asked to write only 20 user stories was to make it more appealing for them to participate in our experiment. The more user stories were asked, the smaller number of subjects were willing to participate. Thus, a tradeoff between the number of stories written by a subject and the number of subjects was made.

8.2.3 Limitations

The main limitation of our research is the small sample size of the experiment subjects. Due to the time limit, only 10 participants were involved in the experiment and this makes it hard to define the complete benefits and drawbacks of the Story Suggestor Tool.

Another limitation of the Story Suggestor Tool is that the new extracted user stories based on BPMN were not always syntactically correct. That was due to the fact that the user stories extraction process replaced the “type of user” with “pool / lane” and the “goal” with “task” as it is explained in the chapter 6.

Consequently, the subjects accepted the suggested user stories as a context but then they made some rephrasing for the sentence to be more syntactically correct.

8.2.4 Future Research

Several opportunities for future research were identified. In this section, we present some of our ideas for further research that will help the research community of requirements engineering to achieve its goal of capturing high quality requirements specification.

Firstly, as aforementioned in chapter 2, there are four main characteristics that define the system requirements specifications. However, in our research we solely focused on two characteristics only: completeness and correctness without compromising the other two: ambiguity and consistency. We do recommend for further research that it includes the other two characteristics.

Secondly, our approach used a semi-automated tool to refine the requirements in agile development methods. Although the Story Suggestor Tool automatically processes the user stories that have been written by the analyst, calculates the semantic similarity and automatically suggests user stories, the part of extracting the user stories from the BPMN diagram is done manually. Therefore, more development and investigation are needed in order to develop a fully automated tool that is able to extract the potential user stories from a BPMN diagram.

Thirdly, we developed our Story Suggestor Tool using Google Colaboratory which is a free cloud service environment that requires no setup and runs entirely in the cloud. Despite the fact that the tool is easy to use as most of the subjects stated in their feedback, the need for a more intuitive, interactive and user-friendly interface is needed. Nonetheless, we consider that the Story Suggestor Tool is already a relevant contribution towards producing more correct and complete requirements specifications.

Bibliography

1. Kaiya, H., & Saeki, M. (2005, September). Ontology based requirements analysis: lightweight semantic processing approach. In Quality Software, 2005.(QSIC 2005). Fifth international conference on (pp. 223-230). IEEE. Chicago

2. Kaiya, H., & Saeki, M. (2006, September). Using domain ontology as domain knowledge for requirements elicitation. In Requirements Engineering, 14th IEEE International Conference (pp. 189-198). IEEE.

3. Pohl, K. (1993, June). The three dimensions of requirements engineering. In International Conference on Advanced Information Systems Engineering (pp. 275-292). Springer, Berlin, Heidelberg.

4. Maedche, A., & Staab, S. (2001). Ontology learning for the semantic web. IEEE Intelligent systems, 16(2), 72-79.

5. Fantechi, A., Gnesi, S., Lami, G., & Maccari, A. (2003). Applications of linguistic techniques for use case analysis. Requirements Engineering, 8(3), 161-170.

6. K. K. Breitman and J. C. Sampaio do Prado Leite. Ontology as a requirements engineering product. In Proceedings of the 11th IEEE International Requirements Engineering Conference, pages 309–319. IEEE Computer Society Press, 2003.

7. Kof, L. (2004). Natural language processing for requirements engineering: Applicability to large requirements documents.2nd International Workshop on Intelligent Technologies for Software Engineering (WITSE04), co-located with ASE2004, 21st September 2004, Linz, Austria, 2004

8. H.B. Reubenstein and R.C. Waters. The Requirements Apprentice: Automated Assistance for Requirements Acquisition. IEEE Trans. on Software Eng., 17(3):226–240, Mar. 1991.

9. Harmain, H. M., & Gaizauskas, R. (2000, September). CM-Builder: an automated NL-based CASE tool. In Automated Software Engineering, 2000. Proceedings ASE 2000. The Fifteenth IEEE International Conference on (pp. 45-53). IEEE.

10. Leite, J. D. P., & Franco, A. P. M. (1993, January). A strategy for conceptual model acquisition. In Requirements Engineering, 1993., Proceedings of IEEE International Symposium on (pp. 243-246). IEEE.

11. Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language processing research. IEEE Computational intelligence magazine, 9(2), 48-57.

12. IEEE Recommended Practice for Software Requirements Specifications, 1998. IEEE Std. 830-1998.

13. Lucassen, G., Robeer, M., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2017). Extracting conceptual models from user stories with Visual Narrator. Requirements Engineering, 22(3), 339-358.

14. Wieringa, R. J. (2014). What is design science? In Design Science Methodology for Information Systems and Software Engineering (pp. 3-11). Springer, Berlin, Heidelberg.

15. Noy, N. F., & Musen, M. A. (2001, August). Anchor-PROMPT: Using non-local context for semantic matching. In Proceedings of the workshop on ontologies and information sharing at the international joint conference on artificial intelligence (IJCAI)(pp. 63-70).

16. Noy, N. F., & Musen, M. A. (2004). Ontology versioning in an ontology management framework. IEEE Intelligent Systems, 19(4), 6-13.

17. Tudorache, T., Vendetti, J., & Noy, N. F. (2008, October). Web-Protege: A Lightweight OWL Ontology Editor for the Web. In OWLED (Vol. 432).

18. Tudorache, T., Vendetti, J., & Noy, N. F. (2008). Web-protégé–protégé going web. *Stanford Center for Biomedical Informatics Research, Stanford University, CA, US*..

19. Rastogi , N., Verma, P., Dr, & Kumar, P., Dr. (2017). analyzing ontology editing tools for effective semantic information retrieval. international journal of engineering sciences & research technology, 6(5), 40-47. doi:10.5281/zenodo.571592.

20. Kaur, N., & Aggarwal, H. (2017). Evaluation of information retrieval based ontology development editors for semantic web. International Journal of Modern Education and Computer Science, 9(7), 63.

21. Slimani, T. (2015). Ontology development: A comparing study on tools, languages and formalisms. Indian Journal of Science and Technology, 8(24), 1-12.

22. Shamsfard, M., & Barforoush, A. A. (2003). The state of the art in ontology learning: a framework for comparison. The Knowledge Engineering Review, 18(4), 293-316.

23. Gómez-Pérez, A., & Manzano-Macho, D. (2003). A survey of ontology learning methods and techniques. OntoWeb Deliverable D, 1(5).

24. Sabou, M., Wroe, C., Goble, C., & Mishne, G. (2005, May). Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In Proceedings of the 14th international conference on World Wide Web (pp. 190-198). ACM.

25. Hazman, M., El-Beltagy, S. R., & Rafea, A. (2011). A survey of ontology learning approaches. database, 7(6).

26. Sanchez, D., & Moreno, A. (2004). Creating ontologies from Web documents. Recent advances in artificial intelligence research and development. IOS Press, 113, 11-18.

27. Sabou, M., Wroe, C., Goble, C., & Mishne, G. (2005, May). Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In

Proceedings of the 14th international conference on World Wide Web (pp. 190-198). ACM.

28. Cimiano, P., & Völker, J. (2005, June). text2onto. In International conference on application of natural language to information systems (pp. 227-238). Springer, Berlin, Heidelberg.

29. Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11), 39-41.

30. Zesch, T., Müller, C., & Gurevych, I. (2008, May). Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary. In LREC (Vol. 8, No. 2008, pp. 1646-1652).

31. Karoui, L., Aufaure, M. A., & Bennacer, N. (2004, September). Ontology discovery from web pages: Application to tourism. In In the Workshop of Knowledge Discovery and Ontologies.

32. Davulcu, H., Vadrevu, S., & Nagarajan, S. (2004, May). OntoMiner: bootstrapping ontologies from overlapping domain specific web sites. In Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (pp. 500-501). ACM.

33. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Kern, J. (2001). The agile manifesto.

34. Cruz, I. F., Antonelli, F. P., & Stroe, C. (2009). AgreementMaker: efficient matching for large real-world schemas and ontologies. Proceedings of the VLDB Endowment, 2(2), 1586-1589.

35. Lucassen, G., Robeer, M., Dalpiaz, F., van der Werf, J. M. E., & Brinkkemper, S. (2017). Extracting conceptual models from user stories with Visual Narrator. Requirements Engineering, 22(3), 339-358.

36. Hustadt, U. (1994, September). Do we need the closed world assumption in knowledge representation?. In KRDB.

37. Bourahla, M. (2015, June). Reasoning over vague concepts. In International Conference on Artificial Intelligence and Soft Computing (pp. 591-602). Springer, Cham.

38. Cohn, M. (2004). User stories applied: For agile software development. Addison-Wesley Professional.

39. Wang, X., Zhao, L., Wang, Y., & Sun, J. (2014). The role of requirements engineering practices in agile development: an empirical study. In Requirements Engineering (pp. 195-209). Springer, Berlin, Heidelberg.

40. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, and S. Brinkkemper. The Use and Effectiveness of User Stories in Practice. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), pages 205–222, 2016a. (Cited on pages 2, 14, 19, 44, 82, 85, 86, 94, 102, 103, 105, 144, 146, and 164.)

41. Lucassen, G. G. (2017). Understanding User Stories: Computational Linguistics in Agile Requirements Engineering(Doctoral dissertation, Utrecht University).

42. Luisa, M., Mariangela, F., & Pierluigi, N. I. (2004). Market research for requirements analysis using linguistic tools. Requirements Engineering, 9(1), 40-56.

43. Berry, D., Gacitua, R., Sawyer, P., & Tjong, S. F. (2012, March). The case for dumb requirements engineering tools. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 211-217). Springer, Berlin, Heidelberg.

44. Goldin, L., & Berry, D. M. (1997). AbstFinder, a prototype natural language text abstraction finder for use in requirements elicitation. Automated Software Engineering, 4(4), 375-412.

45. Popescu, D., Rugaber, S., Medvidovic, N., & Berry, D. M. (2007, September). Reducing ambiguities in requirements specifications via automatically created

object-oriented models. In Monterey Workshop (pp. 103-124). Springer, Berlin, Heidelberg.

46. Fabbrini, F., Fusani, M., Gnesi, S., & Lami, G. (2001). The linguistic approach to the natural language requirements quality: benefit of the use of an automatic tool. In Software Engineering Workshop, 2001. Proceedings. 26th Annual NASA Goddard (pp. 97-105). IEEE.

47. Lin, J., Lin, C. C., Cleland-Huang, J., Settimi, R., Amaya, J., Bedford, G., ... & Zou, X. (2006, September). Poirot: A distributed tool supporting enterprise-wide automated traceability. In Requirements Engineering, 14th IEEE International Conference (pp. 363-364). IEEE.

48. Van Akkeren, E., Baumann, L., Cannegieter, J. J., Hood, C., Hruschka, P., Lampe, M., ... & Zandhuis, J. (2016). Handbook of Requirements Modeling According to the IREB Standard.

49. Pohl, K., & Rupp, C. (2011). Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-Foundation Level-IREB compliant.

50. Van Lamsweerde, A. (2000, June). Requirements engineering in the year 00: a research perspective. In Proceedings of the 22nd international conference on Software engineering (pp. 5-19). ACM.

51. Horkoff, J., Aydemir, F. B., Cardoso, E., Li, T., Maté, A., Paja, E., ... & Giorgini, P. (2017). Goal-oriented requirements engineering: an extended systematic mapping study. Requirements Engineering, 1-28.

52. Dermeval, D., Vilela, J., Bittencourt, I. I., Castro, J., Isotani, S., Brito, P., & Silva, A. (2016). Applications of ontologies in requirements engineering: a systematic review of the literature. Requirements Engineering, 21(4), 405-437.

53. Otero-Cerdeira, L., Rodríguez-Martínez, F. J., & Gómez-Rodríguez, A. (2015). Ontology matching: A literature review. Expert Systems with Applications, 42(2), 949-971.

54. Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Cheatham, M., & Hassanzadeh, O. (2017). OM-2017: Proceedings of the Twelfth International Workshop on Ontology Matching. In OM 2017-12th International Workshop on Ontology Matching (pp. 1-233). No commercial editor.

55. Euzenat, J., & Shvaiko, P. (2013). Ontology matching. Springer Science & Business Media.

56. Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2012). A lightweight guide to the theory and practice of scrum. Ver, 2, 2012.

57. Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). Fundamentals of business process management (Vol. 1, p. 2). Heidelberg: Springer.

58. Ottensooser, A., Fekete, A., Reijers, H. A., Mendling, J., & Menictas, C. (2012). Making sense of business process descriptions: An experimental comparison of graphical and textual notations. Journal of Systems and Software, 85(3), 596-606.

59. Model, B. P. (2011). Notation (BPMN) version 2.0. OMG Specification, Object Management Group, 22-31.

60. Santana, F., Nagata, D., Cursino, M., Barberato, C., & Leal, S. (2016, January). Using BPMN-based Business Processes in Requirements Engineering: the Case Study of Sustainable Design. In Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government (EEE) (p. 79). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

61. Monsalve, C., April, A., & Abran, A. (2011). Requirements elicitation using BPM notations: focusing on the strategic level representation. ACACOS, 11, 235-241.

62. Recker, J., Rosemann, M., Indulska, M., & Green, P. (2009). Business process modeling-a comparative analysis. Journal of the Association for Information Systems, 10(4), 1.

63. Zowghi, D., & Coulin, C. (2005). Requirements elicitation: A survey of techniques, approaches, and tools. In Engineering and managing software requirements (pp. 19-46). Springer, Berlin, Heidelberg.

64. Coulin, C. R. (2007). A situational approach and intelligent tool for collaborative requirements elicitation (Doctoral dissertation, Université Toulouse III-Paul Sabatier).

65. Adam, S., Riegel, N., Gross, A., Uenalan, O., & Darting, S. (2012). A conceptual foundation of requirements engineering for business information systems. In Enterprise, Business-Process and Information Systems Modeling (pp. 91-106). Springer, Berlin, Heidelberg.

66. Cardoso, E. C. S., Almeida, J. P. A., & Guizzardi, G. (2009, September). Requirements engineering based on business process models: A case study. In Enterprise Distributed Object Computing Conference Workshops, 2009. EDOCW 2009. 13th (pp. 320-327). IEEE.

67. Mathisen, E., Ellingsen, K., & Fallmyr, T. (2009, January). Using business process modelling to reduce the effects of requirements changes in software projects. In Adaptive Science & Technology, 2009. ICAST 2009. 2nd International Conference on (pp. 14-19). IEEE.

68. On the other hand, Štolfa, S., & Vondrák, I. (2008). Mapping from business processes to requirements specification. Retrieved on 7th Aug.

69. Giaglis, G. M. (2001). A taxonomy of business process modeling and information systems modeling techniques. International Journal of Flexible Manufacturing Systems, 13(2), 209-228.

70. Kamps, J., Lalmas, M., & Pehcevski, J. (2007, July). Evaluating Relevant in Context: Document retrieval with a twist. In Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval (pp. 749-750). ACM.

71. Stephan van Rooden (2016, March 20). Product Backlog Refinement explained. Retrieved from https://www.scrum.org/resources/blog/product-backlog-refinement-explained-33

72. Tjong, S. F., & Berry, D. M. (2013, April). The design of SREE—a prototype potential ambiguity finder for requirements specifications and lessons learned. In International Working Conference on Requirements Engineering: Foundation for Software Quality (pp. 80-95). Springer, Berlin, Heidelberg.

73. Castañeda, V., Ballejos, L., Caliusco, M. L., & Galli, M. R. (2010). The use of ontologies in requirements engineering. Global Journal of Research In Engineering, 10(6).

74. Laguna, M., & Marklund, J. (2013). Business process modeling, simulation and design. CRC Press.

75. Weske, M. (2012). Business process management architectures. In Business Process Management (pp. 333-371). Springer, Berlin, Heidelberg.

76. Faria, D., Pesquita, C., Santos, E., Palmonari, M., Cruz, I. F., & Couto, F. M. (2013, September). The agreementmakerlight ontology matching system. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"(pp. 527-541). Springer, Berlin, Heidelberg.

Appendix A:

The following table presents the full ontology tool comparison

Tools Features	Import format	Export format	Consistency	Extensibility	Availability	Built in	Ontology	Multi User	Web Support	Merging
Protégé 5.2.0	XML, RDF, XML Schema, HTML,UML	XML, CLIPS, FLogic, RDF,OWL	Yes, via PAL and FaCT	Yes	Free	Yes, with PAL	Files & DBMS	Yes	Yes via protégé OWL plugin	Yes via PROMPT, OWL Diff (Noy
Neon	OWL, RDF	RDF, OWL	Yes	Yes	Free & open	Yes	Files	No	Yes via Neon plug	Yes
SWOOP	OWL, XML, RDF and text	RDF(S), OIL, DAML,OWL,	No	Yes	Free & open	Yes	Files, HTML model	Yes	Limited namespaoe	Yes
Onto Analyzer	C++, Java	Java, C++	No	Yes	Free	No	Files	No	Yes	Yes
Onto Clean	C++, RDF	C++, RDF	Yes	No	Free	No	NI (Not identified)	No	Yes	Yes
RaDON	KIF, XML	KIF, XML	Yes	Yes	Commercial	Yes	Files	Yes	Yes with RSS	Yes
Onto lingua	IDL, KIF	KIF, IDL,OKBC	No	No	Free	No	Files	Yes via write only locking	Yes	Yes via Chimaera
Web Onto	OCML	OCML,	Yes	No	Free	Yes	Files	Yes via global	Web based	None
OilEd	RDF, OIL,DAML	RDF,OIL, SHIQ,dotty,	Yes, via FaCT	No	Free	Yes, with FaCT	Files	No	Limited namespaoe	None
OntoSaurus	LOOM,KIF, C++ IDL,	LOOM, IDL,KIF, C++	Yes	No	Free	Yes	Files	No	No	None
AeroDAML	RDF, DAML	RDF, DAML	Yes	Yes	Free	Yes	Files	Yes	Yes	Yes
COHSE	UML RDF	UML' RDF	NI	No	Free	Yes	File	No	Yes via URL	Yes
DUET	DAML + OIL	DAML + OIL	Yes via FaCT	Yes	Free	Yes	Files	Yes	Yes web based	No
Sesame	RDBMS	RDBMS, OWL	No, tuple	Yes	Free	No	DBMS	Yes	Yes tuple checking	No
Inkling	RDBMS, RDF	RDBMS, RDF	Yes, via updation in	Yes	Free	No	DBMS	Yes	Yes	No
rdiDB	RDF, Java	RDF, Java/Jess	Yes	Yes	Free	No	DBMS(JDBC)	Yes	No	No
Redland	OWL, OCIL	OWL, OCIL	Yes	Yes	Commercial	No	Files	Yes	Yes	NI
Jena	RDF, OWL	RDF, OWL	Yes	Yes	Free	Yes	Files	Yes	Yes via Jena API	Yes
Cerebra	RDF + OVL	RDF + OWL	Yes	Yes	Free	Yes	Files	Yes	Yes	Yes
FCA merge	UML' RDF	UML, RDF, OWL	NI	No	Free	No	NI	No	No	NI

Ontology	Large ontology	Taxonomy	Graphical View	Collaborative	Syntactic	Semantic	Pragmatic	Implementation	Reasoner	Querying	Versioning	**Tools Features**
Yes	Yes	Standalone, Web based, Client/server	Yes	Yes	Yes	Yes	Yes	Java	Pellet	Yes	Yes via change tracker	Protégé 5.2.0
Yes	Yes	Standalone	Yes	Yes	Yes	Yes	Yes	Java Eclipse	Pelle12,HermiT	Yes	Yes	Neon
NI	Yes	Standalone	Yes	Yes	Yes	Yes	No	Java	Pellet	No	Yes	SWOOP
NI	No	2-Tier	Yes	No	No	No	Yes	Java	Pellet	No	No	Onto Analyzer
NI	No	2-Tier	Yes	No	Yes	Yes	Yes	Java	Pellet	No	No	Onto Clean
Yes	Yes	Standalone	Yes	Yes	No	No	No	Lisp	Raoer	Yes	NI	RaDON
Yes	Yes	Client/Server	No	Yes	No	Yes	No	Lisp	JTP, Prolog CML	No	No	Onto lingua
Yes	NI	Client/Server	Yes	Yes	Yes	Yes	No	Lisp	YIN	YIN	YIN	Web Onto
Yes	No	Standalone	No	No	No	Yes	Yes	Java	FaCT	No	No	OilEd
No	NI	Client/server	No	Yes	Yes	Yes	Yes	Lisp	PowerLoom, Stella	No	No	OntoSaurus
Yes	Yes	Standalone	Yes	Yes	Yes	Yes	Yes	Java	FaCT	NI	No	AeroDAML
Yes	Yes	Standalone	No	Yes	Yes	Yes	Yes	Java	Pellet	No	No	COHSE
No	No	Client-Server	No	No	No	No	Yes	Java	Pellet	Yes	NI	DUET
Yes	NI	Client-Server	Yes	No	Yes	No	No	Java	Racer	Yes	NI	Sesame
No	NI	Client-Server	No	Yes	Yes	No	Yes	JDBC	OntoBroker	Yes	NI	Inkling
Yes	Yes	Standalone	No	Yes	No	Yes	No	JDBC	NI	Yes	No	rdiDB
Yes	No	Tier	Yes	Yes	No	Yes	No	JDBC	NI	No	No	Redland
Yes	Yes	Standalone	Yes	Yes	No	Yes	Yes	Java	Pellet, HenniT	Yes	Yes	Jena
Yes	Yes	Standalone	Yes	Yes	Yes	Yes	Yes	Java	Pellet	Yes	Yes	Cerebra
No	NI	Client Server	Yes	No	Yes	No	Yes	Java	HenniT	No	NI	FCA merge

Appendix B:

Experiment Handout Group A

Introduction:

Welcome and thank you for participating in the experiment. My research is about using domain knowledge to refine Agile requirements engineering user stories.

Experiment Description:

In this experiment, we will ask every participant to write 20 user stories that cover the main functionality of a given system, so that the developers can use the stories to develop the system.

About the developed system:

The proposed system is an online MBI Graduation Project Management System. This workflow system will reduce paper work and time consumption, maintain information of students and teachers, computerize the entire activities and operations that concern the graduation phase of the MBI students. The stakeholders of the system will be the Students, Thesis Supervisor and Graduation Project Coordinator

The system supports the MBI graduation project, which is split into two phases: project proposal phase and thesis phase.

A graduation project consists of project idea, a graduation supervisor, and a graduation project facilitator. The latter is the organization where the project is physically conducted, and can either be a company or Utrecht University. In case of a project supported by a company, approval from the graduation project coordinator is required.

The process starts with the student defining a unique project idea in coordination with the student graduation supervisor. Meanwhile, the student is obligated to enroll in MBI biweekly colloquium, in which students present and discuss their own thesis research.

Together with the supervisor, the student assembles the short proposal document that formalizes the topic of the thesis project. Next, the student has to submit a graduation request form. The graduation request form needs to be approved by the Graduation project coordinator.

It is important for the students to subscribe to the MBI graduation group mailing list, so that s/he is informed of important communications from the MBI team.

After 3 months since the beginning of the project, the student needs to schedule and do the first presentation in the MBI colloquium; the result of this phase will be approved by the student graduation supervisor.

Five months after finishing the first phase the student need to schedule and do the second presentation. Likewise, the first phase, the second phase also need to be approved by the student graduation supervisor.

Next, the graduation defense should arrange by the student, the thesis supervisor, and a second examiner. Having approved on the graduation date, the student should schedule and book room with beamer for the final defense also the student must fill out and send assessment form to the supervisor at least two days before the defense. The grade of the defense will be decided and announced afterwards by the student graduation supervisor.

Finally, the student has to publish the thesis to Utrecht University library website, and attends the graduation ceremony to get his diploma.

Task:

You will write 20 user stories concerning the system described above, focussing on the functional requirements of that system.

- Use the Connextra user stories template***:***

“As a <type of user>, I want <goal>, so that <some reason>”

Example: *As a Student, I want to write my academic paper*

Note: you can drop the reason from the previous template

TASK: Please write down 20 user stories based on the above description:

User Story 1:	
User Story 2:	
User Story 3:	
User Story 4:	
User Story 5:	
User Story 6:	
User Story 7:	
User Story 8:	
User Story 9:	
User Story 10:	
User Story 11:	
User Story 12:	
User Story 13:	
User Story 14:	
User Story 15:	
User Story 16:	
User Story 17:	
User Story 18:	
User Story 19:	
User Story 20:	

- Student Name: ……………….
- Student signature: ………………..
- Date: ………………..

Good Luck!

Experiment Handout Group B

Introduction:

Welcome and thank you for participating in this experiment. My research is about using domain knowledge to refine user stories, a language for specifying requirements in agile development. In this context, I have developed a tool called ***Story Suggestor***, which aims to help the analyst in getting higher quality requirements.

In particular, the tool extracts information from a domain knowledge and provides suggestions to the analyst by recommending a number of user stories.

Experiment Description:

In this experiment, we will ask every participant to write 20 user stories that cover the main functionality of a given system, so that the developers can use the stories to develop the system.

About the developed system:

The proposed system is an online MBI Graduation Project Management System. This workflow system will reduce paper work and time consumption, maintain information of students and teachers, computerize the entire activities and operations that concern the graduation phase of the MBI students. The stakeholders of the system will be the Students, Thesis Supervisor and Graduation Project Coordinator

The system supports the MBI graduation project, which is split into two phases: project proposal phase and thesis phase.

A graduation project consists of project idea, a graduation supervisor, and a graduation project facilitator. The latter is the organization where the project is physically conducted, and can either be a company or Utrecht University. In case of a project supported by a company, approval from the graduation project coordinator is required.

The process starts with the student defining a unique project idea in coordination with the student graduation supervisor. Meanwhile, the student is obligated to enroll in MBI biweekly colloquium, in which students present and discuss their own thesis research.

Together with the supervisor, the student assembles the short proposal document that formalizes the topic of the thesis project. Next, the student has to submit a graduation request form. The graduation request form needs to be approved by the Graduation project coordinator.

It is important for the students to subscribe to the MBI graduation group mailing list, so that s/he is informed of important communications from the MBI team.

After 3 months since the beginning of the project, the student needs to schedule and do the first presentation in the MBI colloquium; the result of this phase will be approved by the student graduation supervisor.

Five months after finishing the first phase the student need to schedule and do the second presentation. Likewise, the first phase, the second phase also need to be approved by the student graduation supervisor.

Next, the graduation defense should arrange by the student, the thesis supervisor, and a second examiner. Having approved on the graduation date, the student should schedule and book room with beamer for the final defense also the student must fill out and send assessment form to the supervisor at least two days before the defense. The grade of the defense will be decided and announced afterwards by the student graduation supervisor.

Finally, the student has to publish the thesis to Utrecht University library website, and attends the graduation ceremony to get his diploma.

Task:

You will write 20 user stories concerning the system described above, focusing on the functional requirements of that system.

Please note:

- -Use the Connextra user stories template:

“As a <type of user>, I want <goal>, so that <some reason>”

Example: As a Student, I want to write my academic paper

Note: you can drop the reason from the previous template.

- You have to use the Story Suggestor Tool, the tool mainly takes your user stories as an input and can suggest new user stories as an output. These suggested user stories are derived with the help of known domain knowledge.

- You have free choice to use or not to use the suggested user stories from the Story Suggestor Tool.

How Story Suggestor Tool Works:

The tool is built on Google Colaboratory. Please watch the attached video "Story Suggestor Tutorial" for full description.

Story Suggestor Tutorial Link: https://drive.google.com/open?id=1G_UYevA6k7batSiTyoxI4RFyPuuBR3QT

Story Suggestor tool link:

https://drive.google.com/open?id=1HDyFXrZB5XWGzlw3ZN9E8FQmV58akMCe

- Student Name: ……………….
- Student signature: ………………..
- Date: ………………..

Good Luck!

Appendix C:

The table below shows the output of the experiment for both treatment and control group:

ID	Subject	Group	Story	Relevancy
B1	Student6	Treatment	As a Student, I want to apply for a supervisor	Most Relevant
B2	Student6	Treatment	As a Student, I want to submit a project Idea	irrelevant
B3	Student6	Treatment	As a Supervisor, I want to discuss the research topic	Irrelevant
B4	Student6	Treatment	As a Student, I want to submit my short proposal document	Most Relevant
B5	Student6	Treatment	As a Supervisor, I want to examine the short proposal document	Most Relevant
B6	Student6	Treatment	As a Supervisor, I want to provide feedback for the short proposal document	Most Relevant
B7	Student6	Treatment	As a Graduation Project Coordinator, I want to keep an eye on the MBI biweekly colloquium attendance list	Most Relevant
B8	Student6	Treatment	As a Graduation Project Coordinator, I want to approve an external graduation project	Less Relevant
B9	Student6	Treatment	As a Student, I want to enroll in the MBI biweekly colloquium	Most Relevant
B10	Student6	Treatment	As a Student, I want to submit a graduation request form	Most Relevant
B11	Student6	Treatment	As a Student, I want to subscribe to mailing list	Most Relevant
B12	Student6	Treatment	As a Student, I want to schedule the first presentation in the MBI colloquium	Most Relevant
B13	Student6	Treatment	As a Supervisor, I want to approve the first project phase	Most Relevant
B14	Student6	Treatment	As a Student, I want to schedule the second presentation in the MBI colloquium	Most Relevant
B15	Student6	Treatment	As a Supervisor, I want to approve the second project phase	Most Relevant
B16	Student6	Treatment	As a Student, I want to set the graduation date	Most Relevant
B17	Student6	Treatment	As a Supervisor, I want to approve the graduation date	Most Relevant
B18	Student6	Treatment	As a Graduation Project Coordinator, I want to approve graduation date	Most Relevant
B19	Student6	Treatment	As a Student, I want to book a room with beamer	Most Relevant
B20	Student6	Treatment	As a Student, I want to fill out and send the assessment form	Most Relevant
B21	Student7	Treatment	As a student, I want to submit project Idea	Irrelevant

B22	Student7	Treatment	As a student, I want to submit my project idea	Irrelevant
B23	Student7	Treatment	As a student, I want to register for the graduation project	Most Relevant
B24	Student7	Treatment	As a student, I want to enroll in MBI colloquium	Most Relevant
B25	Student7	Treatment	As a student, I want to subscribe to mailing list	Most Relevant
B26	Student7	Treatment	As a student, I want to set my graduation date	Most Relevant
B27	Student7	Treatment	As a student, I want to request a beamer	Most Relevant
B28	Student7	Treatment	As a student, I want to schedule my thesis defines	Most Relevant
B29	Student7	Treatment	As a supervisor, I want to approve project idea	Most Relevant
B30	Student7	Treatment	As a supervisor, I want to approve student's short proposal	Most Relevant
B31	Student7	Treatment	As a supervisor, I want to grade the first presentation	Most Relevant
B32	Student7	Treatment	As a supervisor, I want to grade the second presentation	Most Relevant
B33	Student7	Treatment	As a supervisor, I want to grade the final defense	Most Relevant
B34	Student7	Treatment	As a graduation coordinator, I want to agree on the short proposal	Most Relevant
B35	Student7	Treatment	As a graduation coordinator, I want to approve the mailing list subscription	Most Relevant
B36	Student7	Treatment	As a graduation coordinator, I want to approve student enrollment	Irrelevant
B37	Student7	Treatment	As a graduation coordinator, I want to approve the project facilitator	Most Relevant
B38	Student7	Treatment	As a graduation coordinator, I want to approve the workplacement agreement	Most Relevant
B39	Student7	Treatment	As a graduation coordinator, I want to approve the graduation request.	Most Relevant
B40	Student7	Treatment	As a student, I want to conduct thesis defense	Most Relevant
B41	Student8	Treatment	As a Student, I want to write my thesis	Irrelevant
B42	Student8	Treatment	As a Student, I want to define a unique project proposal	Most Relevant
B43	Student8	Treatment	As a thesis supervisor, I want to supervise student	Irrelevant
B44	Student8	Treatment	As a Student, I want to formalize the project proposal	Most Relevant
B45	Student8	Treatment	As a thesis supervisor, I want to approve short proposal	Most Relevant
B46	Student8	Treatment	As a Student, I want to enroll the MBI colloquium	Most Relevant

B47	Student8	Treatment	As a student, I want to submit graduation request form	Most Relevant
B48	Student8	Treatment	As a Graduation Project Coordinator, I want to approve graduation request form	Most Relevant
B49	Student8	Treatment	As a Student, I want to subscribe to the MBI mailing list	Most Relevant
B50	Student8	Treatment	As a Student, I want to schedule the first presentation	Most Relevant
B51	Student8	Treatment	As a thesis supervisor, I want to grade the first presentation	Most Relevant
B52	Student8	Treatment	As a graduation project coordinator, I want to approve the first presentation result	Irrelevant
B53	Student8	Treatment	As a Graduation Project Coordinator, I want to approve the second phase	Irrelevant
B54	Student8	Treatment	As a Student, I want to schedule the second presentation	Most Relevant
B55	Student8	Treatment	As a student, I want to set graduation date	Most Relevant
B56	Student8	Treatment	As a Student, I want to book a class room with beamer	Most Relevant
B57	Student8	Treatment	As a Student, I want to inform both supervisors about the specific date and time	Less Relevant
B58	Student8	Treatment	As a Student, I want to send the assessment form to the first supervisor	Most Relevant
B59	Student8	Treatment	As a Student, I want to publish my thesis in the Utrecht University library	Most Relevant
B60	Student8	Treatment	As a Student, I want to attend the graduation ceremony	Less Relevant
B61	Student9	Treatment	As a Student, I want to write down my academic paper	Irrelevant
B62	Student9	Treatment	As a student, I want the graduation project to be managed digitally, so I don't have to do a lot of paper work.	Most Relevant
B63	Student9	Treatment	As a student, I want to enroll to the graduation procedure.	Most Relevant
B64	Student9	Treatment	As a student, I want to hand in my short proposal.	Most Relevant
B65	Student9	Treatment	As a student, I want to hand in my long proposal.	Most Relevant
B66	Student9	Treatment	As a student, I want to plan my first colloquium presentation.	Most Relevant
B67	Student9	Treatment	As a student, I want to arrange my graduation defense.	Most Relevant
B68	Student9	Treatment	As a teacher, I want to approve certain parts of the procedure.	irrelevant
B69	Student9	Treatment	As graduation project coordinator, I want to approve the graduation.	Irrelevant
B70	Student9	Treatment	As a student, I want to plan my second colloquium presentation.	Most Relevant

B71	Student9	Treatment	As a student, I want to arrange a room with a beamer for my thesis defense.	Most Relevant
B72	Student9	Treatment	As a student, I want to find a teacher to help me with my project idea.	Most Relevant
B73	Student9	Treatment	As a student, I want to publish my thesis to the university library website.	Most Relevant
B74	Student9	Treatment	As a student, I want to receive my grade.	Most Relevant
B75	Student9	Treatment	As a teacher, I want to communicate the grading of the project.	irrelevant
B76	Student9	Treatment	As a student, I want to see which colloquium sessions I visited.	Most Relevant
B77	Student9	Treatment	As a student, I want to collect the credits for the colloquium session.	Most Relevant
B78	Student9	Treatment	As a student, I want to subscribe to a mailing list, so I am informed of important information.	Most Relevant
B79	Student9	Treatment	As a supervisor, I want to track the progress of my students.	Most Relevant
B80	Student9	Treatment	As a supervisor, I want to be informed of important upcoming dates for my students.	Less Relevant
B81	Student10	Treatment	As a student, I want to find a graduation supervisor.	Most Relevant
B82	Student10	Treatment	As a student, I want to define a unique project idea	irrelevant
B83	Student10	Treatment	As a graduation project coordinator, I want to know if the student and graduation supervisor agreed on a project idea.	Most Relevant
B84	Student10	Treatment	As a graduation project coordinator, I want to know the formalized topic.	Irrelevant
B85	Student10	Treatment	As a student, I want to enroll in MBI colloquium	Most Relevant
B86	Student10	Treatment	As a student, I want to schedule my colloqium presentations.	Most Relevant
B87	Student10	Treatment	As a student, I want to submit a graduation request form.	Most Relevant
B88	Student10	Treatment	As a graduation project coordinator, I want to asses graduation request forms.	Most Relevant
B89	Student10	Treatment	As a thesis supervisor, I want to approve colloqium presentations of the student.	Most Relevant
B90	Student10	Treatment	As a thesis supervisor, I want to grade the thesis.	Most Relevant
B91	Student10	Treatment	As a student, I want to book a room for the thesis defense.	Most Relevant
B92	Student10	Treatment	As a student, I want to schedule a date with the thesis supervisor and the second examiner.	Most Relevant
B93	Student10	Treatment	As a student, I want my graduation project facilitator approved by my thesis supervisor.	Irrelevant

B94	Student10	Treatment	As a student, I want to attend the colloquim presentations	Irrelevant
B95	Student10	Treatment	As a student, I want to know how many colloquim presentations I have attended.	Most Relevant
B96	Student10	Treatment	As a student, I want to pass my long proposal.	Irrelevant
B97	Student10	Treatment	As a thesis supervisor, I want to approve the colloquim presentations of students.	Most Relevant
B98	Student10	Treatment	As a student, I want to publish my thesis to the Utrecht University website.	Most Relevant
B99	Student10	Treatment	As a student, I want to get notified graduation ceremony	Most Relevant
B100	Student10	Treatment	As a student, I want to send the final assessment form two days the defense.	Most Relevant
A1	Student1	Control	As a graduation project coordinator, I want to approve a project supported by a company	Most Relevant
A2	Student1	Control	As a student, I want to define my project idea in a company coordination with student gradation supervisor.	Most Relevant
A3	Student1	Control	As a student, I want to submit a graduation request form.	Most Relevant
A4	Student1	Control	As a graduation project coordinator, I want to approve the graduation request.	Most Relevant
A5	Student1	Control	As a student, I want to schedule my first presentation	Most Relevant
A6	Student1	Control	As a student graduation supervisor, I want to approve the first phase after the presentation	Most Relevant
A7	Student1	Control	As a student, I want to schedule my second presentation.	Most Relevant
A8	Student1	Control	As a student graduation supervisor, I want to approve the second phase after the presentation	Most Relevant
A9	Student1	Control	As a student, I want to schedule my graduation defense	Most Relevant
A10	Student1	Control	As a student, I want to book a room with beamer for the final defense	Most Relevant
A11	Student1	Control	As a student, I want to fill in a form before my defense	Irrelevant
A12	Student1	Control	As a student, I want to send the assessment form to the supervisor	Most Relevant
A13	Student1	Control	As a student, I want to publish the thesis to Utrecht University Library website.	Most Relevant
A14	Student1	Control	As a student, I want to define my project idea in Utrecht University coordination with student gradation supervisor	Most Relevant
A15	Student1	Control	As a student, I want to enroll in MBI biweekly colloquium	Most Relevant

A16	Student1	Control	As a student, I want to create the short proposal document with the supervisor	Less Relevant
A17	Student1	Control	As a supervisor, I want to formulate the short proposal with the student	Less Relevant
A18	Student1	Control	As a student, I want to subscrive to the MBI graduation mail list	Irrelevant
A19	Student1	Control	As a student, I want to perform a presentation for the first phase	Irrelevant
A20	Student1	Control	As a student, I want to perform a presentation for the second phase	Irrelevant
A21	Student2	Control	As a student, I want to be automatically enrolled for the colloquium when my short proposal is officially approved, so that I do not miss any colloquia.	Less Relevant
A22	Student2	Control	As a student, I want to log in with my osiris id, so that I can acces personalized functionalities.	Less Relevant
A23	Student2	Control	As a student, I want to see the colloquia schedule, so that I can plan my presentation	Most Relevant
A24	Student2	Control	As a student, I want to see an overview of the current researched topics of my fellow students, so that I can get inspired when I search for my own topic for a thesis.	irrelevant
A25	Student2	Control	As a student, I want to see how many students are graduating under certain teachers, so that it helps with my decision of finding a supervisor.	Less Relevant
A26	Student2	Control	As a student, I want to have access to the UU thesis repository, so that I can read and learn from previous projects.	irrelevant
A27	Student2	Control	As a student, I want to see how many months I have left for my short proposal so that I can plan better	Less Relevant
A28	Student2	Control	As a student, I want to see how many months I have left for my long proposal so that I can plan better	Less Relevant
A29	Student2	Control	As a student, I want to see an overview of the teachers research line so that I can find the right supervisor	Less Relevant
A30	Student2	Control	As a teacher, I want to see an overview of my graduate students so that I can manage better	irrelevant
A31	Student2	Control	As a teacher, I want to see which topics my certain students are interested in, so that I can contact them to discuss ideas	irrelevant
A32	Student2	Control	As a student, I want to add my topic interests on my profile, so that teachers can find me as well	Less Relevant

A33	Student2	Control	As a student, I want to search which companies have projects available, so that I can do an internship	Irrelevant
A34	Student2	Control	As a student, I want to be able to upload my graduation request online so that it is can be approved	Most Relevant
A35	Student2	Control	As a teacher, I want to be able to approve graduation requests that are uploaded by my students so that the process is faster	irrelevant
A36	Student2	Control	As a student, I want to be able to book a room for my defense, so that it is easier to plan	Most Relevant
A37	Student2	Control	As a student, I want to be able to to see the availability of my supervisors, so that it is easier to book my defense	Less Relevant
A38	Student2	Control	As a teacher, I want to be able to specify my availability so that my students can plan accordingly	irrelevant
A39	Student2	Control	As a second supervisor, I want to have access to the draft versions of the theses, so that I can reread the thesis.	Most Relevant
A40	Student2	Control	As a first supervisor, I want to have access to my students previously submitted drafts, so that I can refresh my mind.	Most Relevant
A41	Student3	Control	As a student, I want to define my project idea.	Irrelevant
A42	Student3	Control	As a student, I want my supervisor to access my project proposal, so that s/he can give feedback.	Most Relevant
A43	Student3	Control	As a student, I want to enroll in the MBI colloquium.	Most Relevant
A44	Student3	Control	As a student, I want to access the document where the attendance for MBI colloquium is kept track of.	Less Relevant
A45	Student3	Control	As a student, I want to see how many spots for presenting in the MBI colloquium are available in advance. (for example: in the beginning of the project proposal, I would like to check my planned date for MBI colloquium and see whether it is already full so that I can plan and navigate all the project dates around that)	Most Relevant
A46	Student3	Control	As a student, I want my supervisor to have access to my first presentation, so that they can grade it.	Most Relevant
A47	Student3	Control	As a student, I want my supervisor to have access to my second presentation, so that they can grade it.	Most Relevant

A48	Student3	Control	As a student, I want to submit an online graduation request form.	Most Relevant
A49	Student3	Control	As a student, I want more information/help about the fields that need to be filled in the graduation form, so that I don’t waste time emailing back and forth my supervisor.	Less Relevant
A50	Student3	Control	As a student, I want to be automatically subscribed to the MBI graduation group mailing list once I send the graduation request.	Most Relevant
A51	Student3	Control	As a student, I want to schedule the thesis defense on an online system, so that I can choose a room myself.	Most Relevant
A52	Student3	Control	As a student, I want to check which rooms are equipped with a beamer, so that I know which room to reserve.	Most Relevant
A53	Student3	Control	As a student, I want to have joint calendar with the thesis supervisor and second examiner, so that I can schedule the thesis defense.	Less Relevant
A54	Student3	Control	As a student, I want to have my defense date approved on an online system.	Most Relevant
A55	Student3	Control	As a student, I want to check what activities are missing (like publishing the thesis to UU library website) on an online system.	Less Relevant
A56	Student3	Control	As a student, I want an online system that presents my milestones.	Less Relevant
A57	Student3	Control	As a student, I want to check my progress/phases on an online system. (for example: after the project proposal is approved, it is marked with green and I see that the next milestone is the colloquium presentation)	Less Relevant
A58	Student3	Control	As a student, I want to see how many spots are available in the graduation ceremony in advance, so that I can plan the date.	Irrelevant
A59	Student3	Control	As a student, I want to fill out the assessment form in an online system.	Most Relevant
A60	Student3	Control	As a student, I want to read about the graduation ceremony schedule on an online system.	Most Relevant
A61	Student4	Control	As a student, I want to define a new project idea.	Most Relevant
A62	Student4	Control	As a student, I want to update an existing project idea.	Most Relevant
A63	Student4	Control	As a student, I want to enroll in MBI biweekly colloquium.	Most Relevant

A64	Student4	Control	As a student, I want to create the short proposal document.	Most Relevant
A65	Student4	Control	As a student graduation supervisor, i want to approve the proposal.	Most Relevant
A66	Student4	Control	As a student graduation supervisor, i want to reject the proposal.	Irrelevant
A67	Student4	Control	As a student, I want to update an existing (rejected) short proposal document.	Most Relevant
A68	Student4	Control	As a student, I want to submit a new graduation request form.	Most Relevant
A69	Student4	Control	As a Graduation project coordinator, I want to approve the graduation request form.	Most Relevant
A70	Student4	Control	As a Graduation project coordinator, I want to reject the graduation request form.	Irrelevant
A71	Student4	Control	As a student, I want to update an existing (rejected) graduation request form.	Most Relevant
A72	Student4	Control	As a student, I want to subscribe to the MBI graduation group mailing list.	Most Relevant
A73	Student4	Control	As a student, I want to schedule my first presentation in the MBI colloquium.	Most Relevant
A74	Student4	Control	As a student, I want to reschedule my first presentation in the MBI colloquium.	Most Relevant
A75	Student4	Control	As a student, I want to schedule my second presentation in the MBI colloquium.	Most Relevant
A76	Student4	Control	As a student, I want to reschedule my second presentation in the MBI colloquium.	Most Relevant
A77	Student4	Control	As a student graduation supervisor, i want to approve the second phase.	Most Relevant
A78	Student4	Control	As a student graduation supervisor, i want to reject the second phase.	Irrelevant
A79	Student4	Control	As a student, I want to schedule graduation defense.	Most Relevant
A80	Student4	Control	As a thesis supervisor, i want to approve the the graduation defense schedule	Most Relevant
A81	Student5	Control	As a supervisor, I want to grade the defense	Most Relevant
A82	Student5	Control	As a supervisor, I want to announce the grade of my student	Most Relevant
A83	Student5	Control	As a student, I want to attend the graduation cetemony to obtain my diploma	Irrelevant
A84	Student5	Control	As a thesis supervisor, i want to reject the the graduation defense schedule	Irrelevant
A85	Student5	Control	As a second, i want to approve the the graduation defense schedule	irrelevant
A86	Student5	Control	As a second supervisor, i want to reject the the graduation defense schedule	Irrelevant
A87	Student5	Control	As a student, I want to reschedule graduation defense.	Most Relevant

A88	Student5	Control	As a student, I want to schedule and book room with beamer for the final defense.	Most Relevant
A89	Student5	Control	As a student, I want to reschedule and book room with beamer for the final defense.	Most Relevant
A90	Student5	Control	As a student, I want to fill out and send assessment form to the supervisor at least two days before the defense.	Most Relevant
A91	Student5	Control	As a student graduation supervisor, I want to submit the grade of the graduation defense to be announced.	Most Relevant
A92	Student5	Control	As a student, I want to publish the thesis to Utrecht University library website.	Most Relevant
A93	Student5	Control	As a student, I want to attend the graduation ceremony to get my diploma.	Irrelevant
A94	Student5	Control	As a Student, I want to be able to upload my draft versions, so that my teachers can read it and create feedback for the next meeting	Most Relevant
A95	Student5	Control	As a student, I want to be able to access the tool from my phone, so that I can check my details at any time	irrelevant
A96	Student5	Control	As a student, I want to be able to access the tool from my laptop, so that I can check my details any time	Most Relevant
A97	Student5	Control	As a student, I want to be able to be able to download the drafts that I have previously uploaded	Less Relevant
A98	Student5	Control	As a student, I want to be able to read the feedback of my supervisors, so that I can improve my thesis	Most Relevant
A99	Student5	Control	As a teacher, I want to be able to upload feedback on my students draft, so that the feedback process is fast and easier	irrelevant
A100	Student5	Control	As a teacher, I want to be able to inform my students about the topics at the colloquia	irrelevant