Master Business Informatics Thesis

Synthesizing Creative Requirements with Natural Language Processing

Author:
Niels Wever, Bsc
4129075

Supervisors:
Dr. Fabiano Dalpiaz
Dr. Sergio Espana Cubillo

November 20, 2018

Utrecht University

Abstract

This thesis studies creativity in requirements engineering. First, a semi structured literature research explores the subject and studies how creativity in the field of requirements engineering is conceived. This research leads to three dimensions of creativity, usefulness, novelty and surprisingness, and uses these dimensions to assess the creativity of requirements.

Next to this, natural language processing techniques are used to create a program that synthesizes requirements. This program uses semantic role labeling to apply combinational and exploratory creativity in order to synthesize creative requirements. Libraries and resources that are used to do this are PractNLPTools, SemLink, VerbNet, PropBank, and Sense2Vec.

The effectiveness of the program is tested with an experiment. The developed program is used to synthesize creative requirements for two applications. The user stories of these applications are used to do so. One of the applications has additional user stories from similar applications in the domain as input to synthesize requirements. The synthesized requirements of the applications are assessed with a survey on the three identified dimensions of creativity.

Keywords: Requirements engineering, synthesizing requirements, assessing creativity, semantic role labeling

Acknowledgements

This thesis was a difficult project for me which I could not have completed without the help from the people around me. First of all, I want to thank Fabiano for guiding me when I got stuck. Helping me through the process of this exploratory research was most valuable for me. I also want to thank my family and friends for listening to my complains and giving me some needed distractions in between the process. Thanks to you I got my motivation back when I had lost it again. Lastly, I want to thank my roommates. Working full days on our theses and ending it with a nice beer really helped.

Contents

1 Introduction

Requirements engineering (RE) is the discipline that identifies and analyzes the needs of the stakeholders, and turns them into the specification of a system that fulfills such needs (Nuseibeh & Easterbrook, 2000). This process consists of translating goals and needs to functions and constraints for a system. In this process the domain of the system is studied and the requirements are elicited, evaluated, specified, documented and consolidated (van Lamsweerde, 2009). RE is a process with multiple iterations like a life cycle as shown in Figure 1.

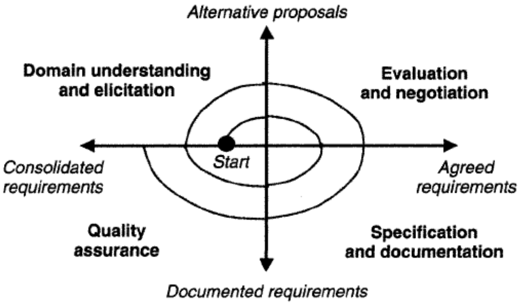

Alternative proposals
Domain understanding and elicitation
Evaluation and negotiation
Consolidated requirements
Start
Agreed requirements
Quality assurance
Specification and documentation
Documented requirements

Figure 1: The RE life cycle described by van Lamsweerde (2009, p. 34)

The RE literature shows that creativity is key in the process. New technologies and innovations change the world fast and companies try to reap the potential benefit of these innovations and technologies (Nguyen & Shanks, 2009). A lot of creativity is involved in the process to achieve this. Next to this, creative requirements are important for organizations so they stay ahead of the competition. As shown in the Kano model in Figure 2, there are *must-haves*, *performance* requirements and *attractive qualities* (Kametani, Nishina, & Suzuki, 2010). The must-haves in requirements engineering are the requirements that users assume are in a product. Next to this every product should have basic requirements to satisfy the user, when these are not present the user will be dissatisfied. These requirements are the performance requirements. However, to distinct your product from others, attractive functionalities should be added to product to increase the satisfaction of the customers. By doing this, an organization can distant itself from the competition. These functionalities are the creative requirements that are researched and this model shows the importance of creativity in the field of RE.

Although the importance of creativity is clear, there is no clear consensus on the meaning of creativity in the field of RE. Different studies have researched the meaning of creativity in the field of RE and the creative aspects of the RE process (Mahaux, Mavin, & Heymans, 2012; Nguyen & Shanks, 2009; Mahaux et al., 2013). In the literature creative requirements are mostly referred as surprising, novel and valuable, but the definition of novelty or surprising is vague. Is something novel when a certain domain has no experience with it? And is something surprising when a certain feature has never been made before in a domain, or is it surprising if something has not even been implemented at all in any domain? To clarify the definition of creativity the first research question of this thesis is as follows:

RQ1. How is creativity conceived in requirements engineering?

Once it is clear what are the factors of creativity in RE, this thesis will explore on a second aspect: does genuine creativity actually exist? If this genuine creativity does

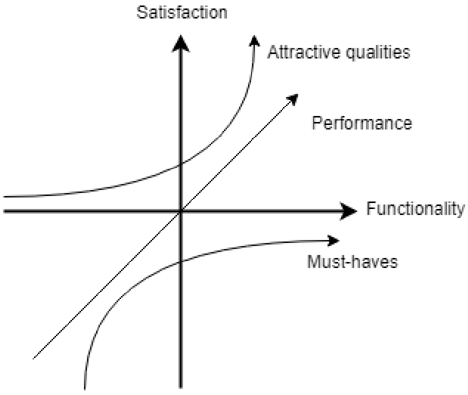


Figure 2: Kano's model

not exist, is it possible to synthesize creativity in requirements? Or is it possible to synthesize creativity when genuine creativity does exist? If this is the case, that would mean that a computer could also create *creative requirements*. In other words, a part of the process of requirements engineering could be (partially) automated, without losing the creativity that is needed in the process. Work of Berry, Gacitua, Sawyer, and Tjong (2012) states that the *thinking* part of requirements engineering should be done by humans. However, the limits of (partially) automating the requirements engineering process are pushed with a tool that incorporates creativity and could be of help to the thinking process of the requirements engineer. This could help to speed up the labor-intensive process, and use the available resources on other parts of the requirements engineering process. To know if it is possible to synthesize requirements the following research question is studied:

RQ2. How to extend a set of requirements with synthesized creativity?

Next to identifying the possibilities of extending a set of requirements, the stakeholder's opinion is needed to know whether this synthesized requirements are creative or not. Therefore the synthesized requirements are needed to be assessed on the creativity to identify wether they could be useful for further used in the RE process or not. Therefore, the following research question is studied:

RQ3. Are synthesized requirements, as made in RQ2, found to be as creative as human made requirements?

This third research question connects the first two research questions. Based on the results of the first research question, a way to asses the creativity could be identified. The results of the second research question will provide a program to come up with synthesized requirements that are needed to be assessed. By answering the three research questions a holistic main research question (MRQ) can be answered to finalize this research.

MRQ. How to automatically synthesize creative requirements?

By answering this main research question, an approach of synthesizing creative requirements is suggested. This is usable for requirement engineers to understand the creative requirements engineering process and make them able (partially) automate it.

2 Research Method

The first research question is answered with a semi-structured literature research on Google Scholar[1] and DBLP[2]. This literature research answers how creativity is conceived in the field of requirements engineering and how creativity could be measured when requirements are synthesized, which is needed for RQ2 and RQ3. The semi structured literature research starts by explaining the field of requirements engineering. Mainly literature indicated as useful by the first supervisor is used to elaborate on requirements engineering. Many sources are available about RE, but the main topic of this thesis is *creativity* in requirements engineering, so main researches in the field of creativity in RE are studied. The main authors of these studies are indicated by the first supervisor of the project, and the relevant researches are picked based on these authors and their references. Furthermore, references and authors are snowballed based on the main papers of the earlier mentioned researches. This literature research is shown in Section 3.

The second research question is concerned with creating new requirements in this experiment. These synthesized requirements should look like requirements that are created by humans so that they are easily understandable and are not an obstacle for the communication of the requirements. In order to achieve this, multiple libraries are identified to have a combination that is able to synthesize requirements. With the help of these libraries a program is created which analyzes existing requirements and synthesizes new requirements based on these existing requirements. The existing requirements are taken from existing applications and documentation. This designed artefact is further elaborated in Section 4.

The last research question is answered with an experiment. Some of the synthesized requirements from the created program will be put together with some of the human made requirements in one list of requirements. This list of requirements is put into a survey. Requirement engineers will be asked to assess the requirements on the different dimensions of creativity. Next to this, an elaboration on this rating is asked. In this way differences could be identified and also the reason of the differences could be explored. This evaluation of requirements is discussed in Section 5.

[1] https://scholar-google-nl.proxy.library.uu.nl/ - The proxy of Utrecht University is used to access papers.

[2] https://dblp.uni-trier.de/

3 Literature

In this literature research, the field of requirements engineering will be explained to show the importance of creativity in RE. Next creativity is studied in different fields, starting with requirements engineering. Other fields are studied as well to get a a broad overview of the definition of creativity. The field of computational creativity is especially studied as it contains definitions and processes that could be used to compute creative requirements, which is needed for RQ2. Computational creativity is defined by Jordanous (2012, p. 248) as *"the study and simulation, by computational means, of behavior, natural and artificial, which would, if observed in humans, be deemed creative"*. Additionally the different techniques of natural language processing (NLP) are explained. Natural language generation (NLG) is discussed as well, as this topic also has relevance to this thesis. The insights of this literature study are added as a concluding section.

3.1 The Requirements Engineering Process

As mentioned in the introduction, requirements engineering is the discipline that identifies and analyzes the needs of the stakeholders and turns them into the specification of a system that fulfills such needs. Nuseibeh and Easterbrook (2000) mention that this RE process starts with context and groundwork. Context and groundwork is needed to prepare for a project and includes a definition of which techniques are used in the RE process. The next step is the elicitation of requirements. This is often referred as the first step because the context and groundwork of a project is already clear in a lot of cases. The goal of the elicitation process is to find the boundaries and the goals of a project. This is usually done by considering all the stakeholders. The elicitation techniques that are used in this step depend on the time and resources that are available. Requirements are modeled during and after the elicitation process to analyze different parts of the project, communicate with the stakeholders, and validate the requirements that are identified.

Pohl's three dimensions of requirements engineering

Pohl (1994) defined three dimensions of requirements engineering in a framework to identify the goals of requirements engineering:

- the specification dimension;
- the representation dimension; and
- the agreement dimension.

These dimensions are based on the main goals of RE identified by Pohl. These goals are completing the specification of a system, turning this specification into formal language and get a common agreeing on this formal specification. In these dimensions the initial input represents informal specification of opaque personal views on the system, and the desired output is a formal common specification of the system. Both initial input and desired output are shown in Figure 3.

The first dimension, the specification dimension, is concerned with building a complete specification of of the system. The next dimension, the representation dimension, is concerned with different representations of a system: the informal representations and the formal representations. Informal and semi-informal representations are used to get a good and quickly understandable overview of the system, while formal representations are used to make sure the requirements are understood in one way only. All the representations need to stay consistent during to process. The agreement dimension is concerned with all the views of the different stakeholders and results in a common agreement on the requirements of a system. Therefore, the different views of each stakeholder should be taken into account and conflicts between these views should be solved.

The process from initial input to desired output is not a straight way through the different dimensions. Improvements in one dimensions could lead to setbacks in other dimensions. This is also shown in Figure 3.

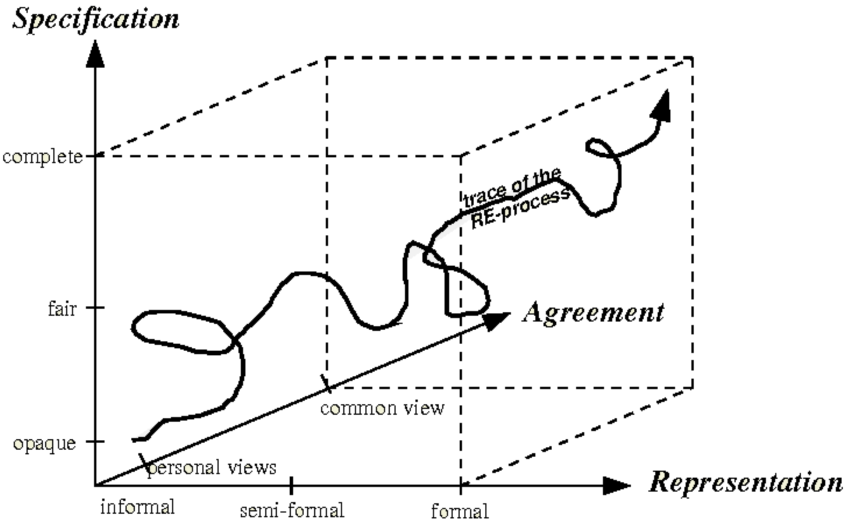


Figure 3: The three dimensions of Pohl (1999, p. 25)

From an as-is to a to-be system

van Lamsweerde (2009) describes requirements engineering as a process where the to-be system needs to be identified and where an as-is system always exists. The to-be system is the system that should work in the place of the as-is system. Based on this, a preliminary definition of requirements engineering is given as follows: "*In this setting, we may apprehend requirements engineering more precisely as a coordinated set of activities for exploring, evaluating, documenting, consolidating, revising and adapting the objectives, capabilities, qualities, constraints and assumptions that the system-to-be should meet based on problems raised by the system-as-is and opportunities provided by new technologies.*" (van Lamsweerde, 2009, p. 6).

Based on this definition van Lamsweerde explains three dimensions of requirements engineering in his book: the *why*, *what*, and *who* dimensions. To identify the to-be system the different dimensions are identified. The *why*-dimension is concerned with the objectives of the system to-be. In this dimension, the current limitations of the as-is system are identified, the opportunities are studied and domain knowledge is acquired. In the *what*-dimension, the functional requirements and the constraints of the to-be system are identified to satisfy the objectives of the *why*-dimension. Next to this, assumptions of the system are made. These assumption help to decide how the system should work. Based on the functional requirements, constraints and assumptions the *who*-dimension is addressed. In this dimension the responsibilities are assigned. This can either be to the to-be system or to the environment. The dimensions are visually shown in Figure 4.

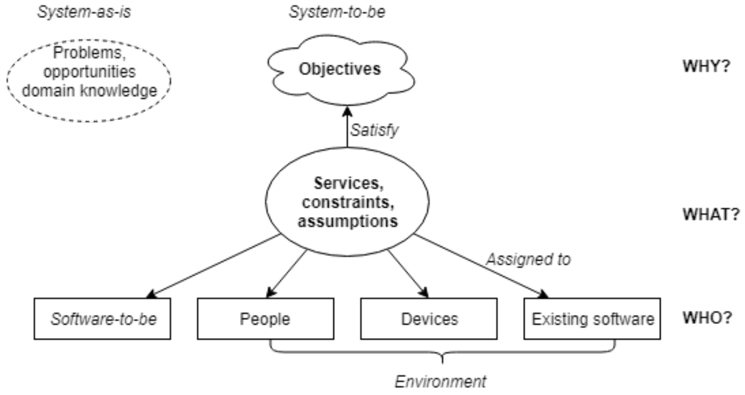


Figure 4: The three dimensions of van Lamsweerde (2009)

Next to the different dimensions in RE, van Lamsweerde (2009) describes the types of statements that are involved in the RE process. Descriptive statements, are statements that are always true, independent of the behavior of the system. Prescriptive statements on the contrary, are statements that should be true through system behavior. Therefore, prescriptive statements can be discussed but descriptive statements cannot. Prescriptive statements that should be enforced by the system to-be are system requirements, statements that only should be enforced by the software to-be are defined as software requirements. Because the software is part of a system, software requirements are per definition system requirements. Domain properties are descriptive statements that are often correspond to physical laws. These domain properties describe the problem world. When these are implemented in the system, assumptions are made. These assumptions are prescriptive as they constrain the behavior of the system. The last type of statement that is described by van Lamsweerde (2009) are definitions. They give an precise meaning to domain concepts and terms.

Categories of requirements

Requirements can be categorized according to multiple frameworks. RE knows functional requirements, non-functional requirements, and quality requirements (van Lamsweerde, 2009). Functional requirements describe the effect that the software to-be has on its environment. Sometimes these requirements are called features. Non-functional requirements, better named as constraints, describe the constraints that the software to-be should satisfy when the system is being developed. Quality requirements describes *how well* the system should perform certain tasks. Quality requirements can be further specified in safety, confidentiality, privacy, integrity, availability, reliability, accuracy, performance and interface requirements.

Requirements life cycle

van Lamsweerde (2009) describes that the requirements engineering process can be seen as a life cycle. This life cycle, as shown in Figure 1, has the following phases:

1. *Domain understanding*: The domain of the project should be studied. The as-is situation and stakeholders are identified and with these stakeholders the to-be situation is discovered.
2. *Requirements elicitation*: Candidate requirements are discovered in this phase by looking into opportunities, points to improve, organizational and technical constraints, typical scenario's and the domain properties.

3. *Evaluation and agreement*: In this phase, the candidate requirements are discussed. Conflicting concerns and risks are identified and resolved, and alternative options are discussed. A prioritization is made next.
4. *Specification and documentation*: The requirements are specified and documented when they are known in the requirements document.
5. *Requirements consolidation*: For quality assurance the requirements are validated with the stakeholders to make sure the actual needs are satisfied and verified to make sure no inconsistencies or conflicts arise.

The phases are done several times in multiple iterations. Typically iterations exist within the RE process itself, but also during the software development and after the software deployment. The requirements should all be complete, consistent, adequate, unambiguous, measurable, relevant, feasible and comprehensible to make sure the requirements document is comprehensible and complete. Next to this, the requirements document should be good structured and modifiable, and the context should be traceable.

Companies and customers

Companies are trying to make the elicitation process faster, so products can be released to the market in less time (Mustasfa, 2014). Next to this, business are looking for requirements that satisfy the customer and keep the product better than products of the competitors. Therefore, businesses count on high quality requirements elicitation work to prevent errors caused by missing or wrong requirements. Next to this, businesses have to keep customers satisfied. Kano proposed a model for the product quality measurement and shows the customers satisfaction when certain requirements are met (Kametani et al., 2010; Mustasfa, 2014). In this model, three types of requirements are identified as shown in Figure 2. The must-have requirements are taken for granted and should be in the product. The performance requirements are the requirements that customers expect in the product and satisfies customers when they are implemented, but dissatisfies customers when they are not. The attractive requirements are the requirements that are not expected by the customer and improves the customer satisfaction quickly.

User stories

User stories are used to express requirements and communicate them in a structured way, while capturing the goals of different kinds of users and their reason behind it (Lucassen, Dalpiaz, Werf, & Brinkkemper, 2016). The structure of a user story is as follows:

As a [user], I want to [goal], so that [benefit].

They are mostly used in combination with agile software development and less with traditional methods such as the waterfall method. Especially in the agile software development, requirements engineering plays an important role. The standardized way is useful to represent the requirements that need to be developed in an understandable way. It also helps to define the right requirements and explain them. Practitioners agree that user stories help with assuring the quality and productivity of the software development process (Lucassen et al., 2016). However, they do not think it's use increases the speed of the development process.

3.2 Creativity in Requirements Engineering

This thesis is started by looking into the two frameworks created in 2009 and 2012 by two research groups who were studying creativity in the field of requirements engineering (Nguyen & Shanks, 2009; Mahaux et al., 2012), as main authors of these studies were indicated by the first supervisor as leading researchers in the field. Based on the papers of these frameworks, more work of the authors of these framework is further studied and relevant papers from the references are researched with the use of the snowballing technique. By researching both frameworks, multiple perspectives are researched. Next to this, snowballing into different domains is used to get a general definition of creativity. These other domains will be discussed in the next subsections.

Creative elements, levels and loci

Nguyen & Shanks (2009) created the first theoretical framework in the field of RE on the topic of creative requirements. They define RE as a creative process that consists of three dimensions: creative elements, levels and loci. These dimensions are used to understand creativity in RE and are further defined in subdimensions:

- The creative elements are separated into five elements: product, process, people, domain and socio-organizational context.
- The creative levels are consist of three different levels: the psychological level, the historical level and the situated level.
- The creativity loci are defined in two different loci: production and recognition, and adoption and diffusion.

Based on the five creativity elements Nguyen & Shanks (2009) identified implications for RE and listed related research on the topic. They state that the elements should be empirically examined to study how these apply to RE and to include them in the RE process. Especially novelty and surprisingness, which they mention as factors of a creative product (an element of the creative elements dimension), are mentioned to incorporate into existing quality requirement frameworks to encourage creativity in the RE process.

Contextual factors and dimensions

Mahaux et al. (2012) did a similar research, but came up with a different framework than Nguyen and Shanks (2009). The paper mentions that this framework is not necessarily better or more complete, but it does provide another perspective on creativity in RE. In this paper contextual factors are identified as the elements that influence creativity and dimensions are identified as elements that characterize creativity. To create to this framework, they did a literature research to cover the field of RE, and used the book "Explaining Creativity" of Sawyer (2006) to cover studies in social sciences. With the created framework the creativity in a project can be analyzed.

The framework consists of three contextual factors and five dimensions and these are shown in Figure 5. The first contextual factor is culture: *"Culture is the set of shared values, goals, attitudes, and practices that characterises a group of people"* (Mahaux et al., 2012). Each project has its own local culture and this changes over time. Painting nature as good as possible was considered creative in the renaissance, while currently being innovative and creating new things is often considered as creative. The domain factor considers the application domain of the RE process. The creativity in a domain can differ from other domains. For example, a cook has other creative applications than a designer in the gaming industry. These differences are domain-specific factors. Next to the differences, there are also domain-independent factors such as intelligence, motivation and openness. At last, resources for a project have influence. More resources do not always mean that there is more creativity. Based on these factors Mahaux et al. propose research areas to study the correlation between creativity in the RE process and each contextual factor. Five dimensions are identified next to the three contextual factors. These dimensions and their explanations are shown in Table 1.

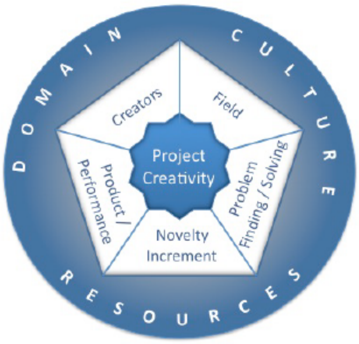


Figure 5: The three contextual factors and five dimensions of the framework of Mahaux, Mavin, and Heymans (2012)

Table 1: The five dimensions and their explanations of the creativity framework of Mahaux, Mavin, and Heymans (2012)

Dimension	**Explanation**
The creative group	The activities of the RE process of a person working alone differs from the activities of a group. Even between groups creative activities and results differ. A group working together is dependent on the collaboration, openness and culture within the group. Therefore, the group has influence on the creative process and outcome.
The field	The size of the field could differ from an individual with some insights to a history of research were many people are involved. A large field could require more resources to be creative than a small field.
The size of novelty increment	The evolution of a product could create new value to a product with some risk. When a product is radically changed, the risk of acceptance is much higher, but the payoff could be higher as well.
Performance and product orientation	Performance oriented creativity focuses on the process itself, such as a concert that is given. A product oriented focuses much more on the result, the product, which could be a book for example. This difference is especially found in social sciences which were studied with the book of Sawyer (2006). No RE authors identified the difference between these orientations.
Problem-finding and problem-solving orientation	The last dimensions is the difference between problem-finding and problem-solving creativity. Problem-finding is a very divergent approach, while problem-solving is a convergent solution finding approach.

The complete framework is not completely relevant, as the definition of creativity for a requirement is researched in this research. However, Mahaux et al. (2012) show the different dimensions and factors that do impact the result in the RE process.

Individual and Social Dimension

Dallman, Nguyen, Lamp, and Cybulski (2005) did an empirical study on contextual factors that influence the creativity of individuals and groups. The research mentions the importance of creativity of information systems in the future, as it will influence the market and economic trends. Therefore they researched the contextual factors by studying students that worked in group projects for different clients. The researchers studied the groups' progress and their collaboration. Based on this they created a list of individual and social context factors that influenced the creativity of the project. These factors are seperated in two dimensions: factors that had influence on the individual dimension and factors that had influence on the social dimension. The factors that influenced the individual dimension are motivation, personal agenda, self perception of being creative, perception and knowledge of creativity, creativity education, experience and design bias, and conformance versus risk taking. The factors that influence the social dimension are national culture, project group behavior, management, organization and stakeholder. In this dimension project group behavior is split into leadership and team/group dynamics, organization is split into constraints and consequences, and stakeholder is split in expectations and conflict. The effect of the identified factors is not studied and thus not elaborated in the study of Dallman et al. (2005).

A collaborative creative process

RE is described as a collaborative creative process in the follow-up paper on which several authors of the previous mentioned papers worked (Mahaux et al., 2013). The introduction of this paper immediately starts with the following definition of creativity: *"the ability to produce work that is both novel (i.e. original, unexpected) and appropriate (i.e. useful, adaptive to task constraints)"*. Based on this definition they state that creativity is based on the environment as raises questions compared to what something is creative and for whom something is appropriate. The research itself focuses on the process of RE and studies whether it can be considered as a creative collaborative process or not. They tried to do so by doing a literature study and using a non-anonymous Delphi technique. The literature study concludes that RE is dealing with a complex and multidimensional nature of a problem and solution space, which can be addressed best by working together. This collaboration also helps to be creative. The Delphi technique is used to identify the factors of the creative collaborative process. At this point these factors are identified, but how to use the factors to influence creativity in a positive way has yet to be studied. The identified factors are shown in Figure 6 and explained in Table 2 and 3.

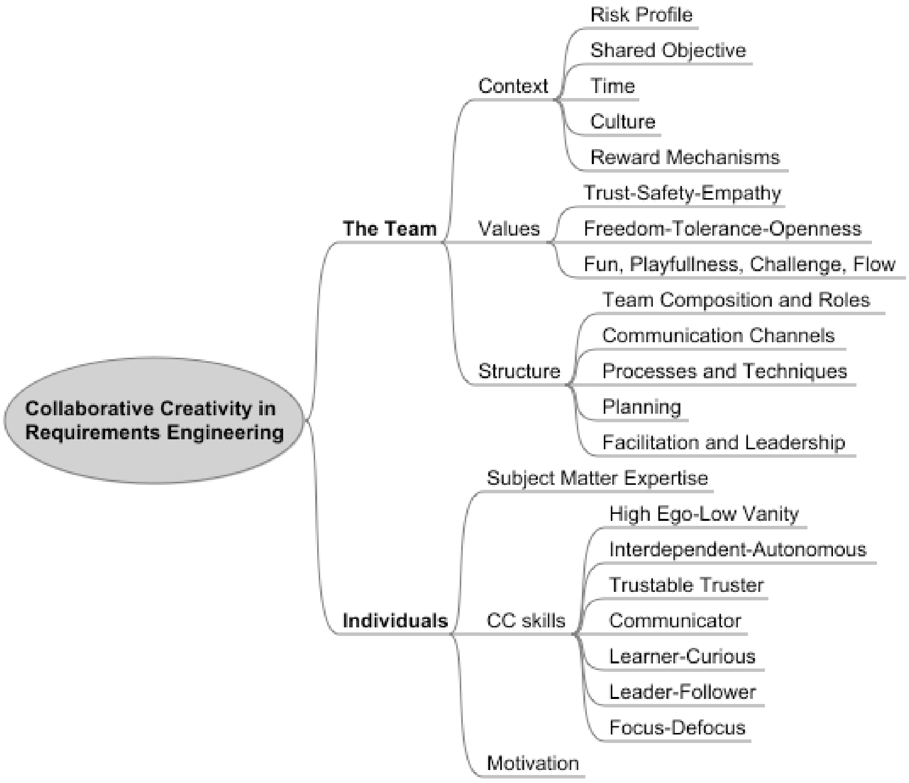


Figure 6: The identified factors that influence the creative collaboration process (from Mahaux et al., 2013)

Table 2: Explanation of the identified individual factors that influence the creative collaboration process

Subject Matter Expertise	The right skills should be available in the team, but often do not have influence in the creative collaborative process.
CC Skills	The experts focused mostly on collaborative team factors, and identified the following collaborative creativity skills for individuals that should be taken into account: *High Ego-Low Vanity*, *Interdependent-Autonomous*, *Trustable Truster*, *Communicator*, *Learner-Curious*, *Leader-Follower*, and *Focus-Defocus*
Motivation	The commitment of every individual has influence on the complete group. Also the agenda of every individual should be aligned with the agenda of the whole group.

Table 3: Explanation of the identified team factors that influence the creative collaboration process

Context	*Risk Profile*	Companies have norms which should be clear. By breaking these norms risk is involved. Conservative companies should have incremental innovation to expand the norms slowly without having to much risk, as the risk is equal to the distance between the new idea and the norm.
	Shared Objective	For a good collaboration a common vision or objective is needed. This objective should not be to restrictive to allow creativity, but also should not be too vague to prevent chaos.
	Time	Limiting the time prevents a hindered focus, but too little time prevents proper exploration and discussing. A balance is needed and flexibility is advised.
	Culture	Awareness of the differences between cultures is needed to support a creative collaboration.
	Reward Mechanisms	In a creative collaborative process it is difficult to reward an individual for their participation in a certain solution, because you cannot split up the result into individual contributions. Next to this, a team reward could decrease the willingness to participate.
Values	*Trust, Safety, Empathy*	A relaxed environment, psychological and physical, is needed to have an open collaboration.
	Freedom-Tolerance-Openness	Discussion between different team members could lead to new creative ideas that satisfy all participants. Therefore, an open atmosphere is needed to support and allow this.
	Fun, Playfulness, Challenge, Flow	A relaxed tone that supports elements of play is advised by the experts of this research.
Structure	*Team composition and Roles*	Teams should have persons with different expertises to have a range of knowledge within the group. However, the members of the teams should have enough overlap to understand each other.
	Communication Channels	There are a lot of possible communication channels and ways of communication possible which affect the creativity during the process.
	Processes and Techniques	Many methods are developed to encourage a creative process. While using these methods the whole environment should support these to support the collaborative creativity.
	Planning	Planning should give a time-frame to each activity but should be flexible too.
	Facilitation and Leadership	In a collaborative creative process there should not be someone that acts as a leader, but rather someone that acts as a facilitator. All members should take the responsibility of the final output.

Problem solving

A paper of Cybulski and Nguyen (2003) tried to explore creativity in RE as well. A focus group of practitioners, consisting of one academic and five company employed members, is used in this research to study this. They identify RE as a problem-solving activity with three elements: context, process, and outcome. One of the conclusions of these elements was that a creative outcome of RE has an high priority on the information system professional agenda. The solutions however, are usually created in a short time-frame and have to provide value to a business. An effective solution can therefore also be to make systems appear creative without creative development process to support it. This conclusion shows the relevance of this research and makes clear that business may be interested in synthesized creativity RE.

Maiden et al. (2010) framed requirements engineering as a creative problem solving process and motivated a research agenda. They stated that requirements engineering could produce more useful and novel outcomes if the process would be treated as a creative problem solving process. In this research the definition of Sternberg is used for creativity: "*the ability to produce work that is both novel (i.e. original, unexpected) and appropriate (i.e. useful, adaptive tot task constraints)*". To further specify the definition of novel, the work of Boden is used. This work is discussed in Section 3.3. Four drivers are explained to motivate the research of Maiden et al.:

A. The strategic importance of creativity and innovation: creativity and innovation is an acknowledged competitive advantage that is even more important in difficult economic times.
B. A wider range of solution technologies.
C. The growth in agile development methods: short development cycles could discourage incubation of new creative ideas. They argue that new development methods should create new opportunities to discover new requirements.
D. Trends in requirements research and practice: the gap between the need of RE practitioners and the current RE research needs to be reduced.

Next to these drivers different creativity models that are applicable to requirements engineering are discussed. One of these models are the three different types of creativity of Boden: exploratory, transformational and combinational creativity. With these types the different ways of interpreting RE are discussed. The work of Ritchie (2001) defined formal rules to asses novelty. Based on these rules, a novel requirement is defined by Maiden et al. as a requirement that is dissimilar to reference requirements. The difference of incubation and illumination in creativity is also of importance in RE. Often new ideas surface in an illumination phase, but to get these ideas accepted the incubation period is important. The different types of creative process are important as well. The structuralist processes are focused on rational techniques and mostly evolve current ideas, while inspirationalist and situationalist processes are mostly focused on seeking new ideas with different insights or by looking to the surrounding. According to Maiden et al. a combination of all techniques is best to get the most effective creative process.

To support creativity techniques, new requirements engineering techniques and tools should be developed according to Maiden et al. (2010). These techniques should focus more on finding a lot of incomplete new ideas than precisely work everything out. To do this, tools should be available to support the new RE techniques. These tools should support exploration and experimentation without any restrictions.

A combinational approach

Bhowmik, Niu, Savolainen, and Mahmoud propose a novel framework to use topic modeling and part-of-speech tagging to generate new requirements with the help of combinational creativity. Combinational creativity is creativity where new ideas, artifacts or processes are created by combining different ideas, artifacts or processes (Boden, 2009). This concept will be further discussed in Section 3.3. Capturing requirements that are new to stakeholders is considered as creative in RE (Maiden et al., 2010). In the framework of Bhowmik et al. dissimilarity is used to find unfamiliar combinations to create requirements (output) based on different requirements documents (input). The framework consists of five phases. In the first phase a requirements documents and communication about requirements is taken as input. The second phase consists out of grouping different stakeholders together in groups that have the same mindset and thus the same kind of ideas. The main topics that are discusses within these groups are then identified by using Latent Dirichlet Allocation (LDA) in phase three. In the fourth phase all the familiar combinations between these topics are eliminated, such that only unfamiliar topic combinations are available. Based on these unfamiliar combinations new requirements are generated in the last phase. The five phases of this framework are visually shown in Figure 7.

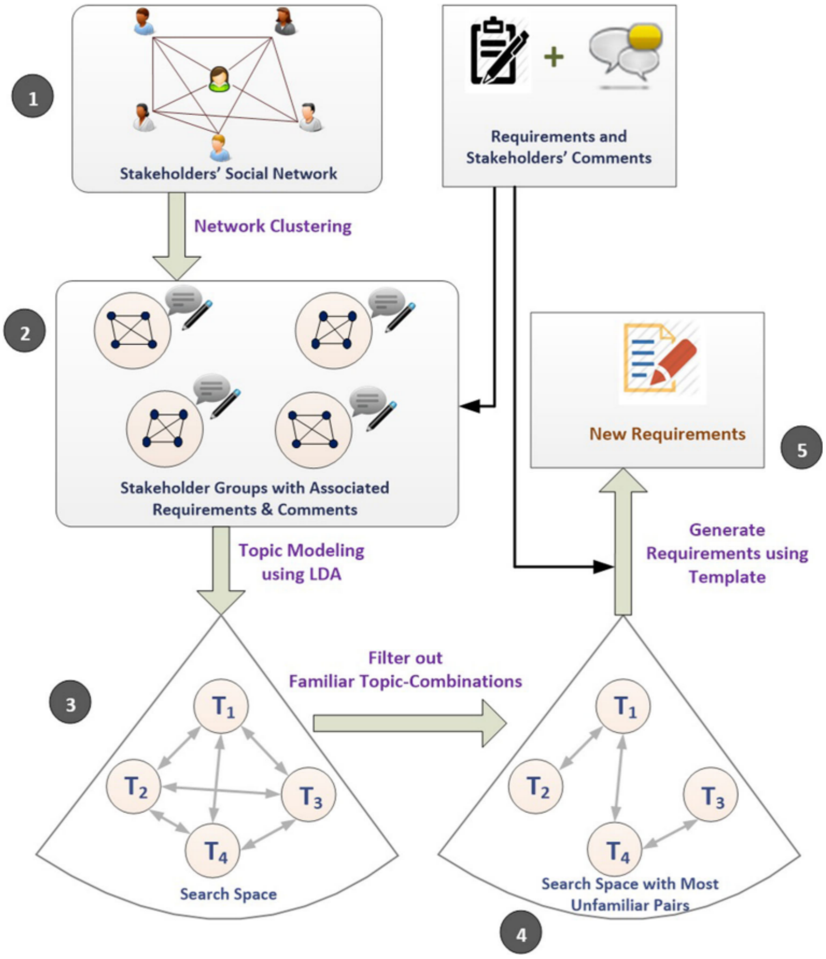


Figure 7: The framework for combinational creativity of Bhowmik, Niu, Savolainen, and Mahmoud (2015, p. 258)

3.3 Creativity Beyond Requirements Engineering

A chapter in a book of Arthur Cropley, a professor in Engineering Innovation, and David Cropley, a professor in Educational Psychology, describe the distinction between aesthetic and functional creativity (Cropley, Cropley, et al., 2005). Aesthetic creativity is the creativity that is involved in art, while functional creativity is the creativity that results from creativity that has a purpose. Only aesthetic creativity was studied in the past, until convergent thinking changed to divergent thinking in the commercial markets. Convergent thinking leads to conventional products, and limits the use of products. When divergent thinking was introduced, this was associated to novelty and creativity. Guilfords' paper in 1950 about divergent thinking was titled *Creativity.* At the time, creativity has become so important because of the rapid change, it was stated that "*societies will stagnate, even perish unless their leaders in all fields become more creative*"(Cropley, Cropley, et al., 2005, p. 170). From this moment on, creativity is not only studied in the aesthetic domain, but in all domains. *Creativity with a purpose*, as Cropley and Cropley call it, is about creating creative products, ideas and processes with the help of creativity, while being usable and useful. This means that a creative product should be functional and be evaluated on relevance and effectiveness. In the non-aesthetic domains a product that does not have a purpose is useless, thus the first criteria in these domains on creative products is the relevance and effectiveness of a product. Next comes novelty, because a product that is not relevant or effective is not a functional but a aesthetic product. Rating a product on these two criteria can be done by experts in the domain. It appears that this is applicable in a constant and reliable way as Cropley, Cropley, et al. found in the research of Besemer and O'Quin (1987).

Three ways of creativity

Boden (2009) defines creativity as "*the ability to come up with ideas or artifacts that are new, surprising, and valuable*". She adds to this definition that you cannot just ask if something is creative or not, but that you should ask how creative something is. Someone who thinks of a new idea by its own can be very creative for example. This creativity is on a personal level, also called psychological creativity (P-creativity), the idea is new to the person itself. However, when something is new and no-one has thought of it before, it is on a different creative level. This level is called historical creativity (H-creativity). H-creativity is a special case of P-creativity. Next to creativity, Boden also describes the different meanings of surprising as this is a part of the definition of creativity. Something may be surprising when it is unfamiliar of unlikely. Next, something is surprising when a certain idea fits in a different context or domain. The last meaning is the feeling you get when you encounter something that is you thought is impossible.

The three ways of creativity described by Boden (2009) are combinational, exploratory and transformational creativity. These ways of creativity are earlier mentioned in the previous subsection. Combinational creativity is the result of combining two different things into a new idea or artifact. This combination should have a link to get some value. Exploratory creativity is the result of thinking out of the box within a conceptual space. New possibilities that have not been thought about become possible. When the limit of a conceptual space is reached transformational creativity comes into place. The conceptual space is changed or radically reformed to make new ideas possible.

With these ways of creativity Boden (2009) discusses the way computer creativity could be achieved. Even in the case that computers cannot be genuinely creative, she states that this does not mean the subject is not interesting anymore. Synthesizing creativity and making computers appear creative is interesting on its own. Combinational creativity is an easy to model in a computer. However, to make the combinations valuable the computer should have extensive knowledge to form links between different ideas and artifacts to make valuable combinations. A good example of computer-based combinational creativity are JAPE or STANDUP, which are automated punt generators (Manurung et al., 2008). Exploratory creativity has also current examples. AARON is a computer-based program that creates drawings and paintings in certain styles while exploring the conceptual space of a certain style (McCorduck & Cohen, 1991). Transformational creativity requires computer programs to alter their own rules to achieve this. Despite all the attempts and research on (computational) creativity, Boden states that there cannot be a consensus of the creative value of a program or its result by definition. Value differs to everyone and we cannot define this into words, especially aesthetic values. For a computer model however, this should be stated really clear.

3.4 Computational Creativity

Ritchie proposed formal criteria to evaluate creativity in 2001, which were revised in 2007. These criteria are used as a basis of the Standardized Procedure for Evaluating Creative Systems (SPECS) created by Jordanous (2012). Evaluating creativity in systems is important to point out strengths and weakness which can be used to improve a system. Jordanous stated the formal criteria in a natural way to achieve a faster understanding of them. These criteria are based on how typical an artefact is in a certain domain and how valuable an output is. Not all of the criteria created by Ritchie are needed to evaluate the creativity of a system. Next to this Jordanous also mentions that the some formal criteria are hard to implement and others could be implemented with implementation decisions of the evaluator, which could be biased.

Next to Ritchie's formal criteria, Jordanous (2012) mentions Colton's Creativity Tripod Framework (2008). This framework concentrates on the behavior of a system and evaluates this on three criteria: skill, imagination and appreciation. According to this framework, a system has to demonstrate that it works in a creative way, rather than have a creative artifact as an output.

The third theory evaluated by Jordanous (2012) are the FACE/IDEA models. This combination of models represent and evaluate creative acts. At the stage of writing her paper these models were still in progress, but the aim of these models is to formalize each term and have a conceptual definition which can be used to assess computational creativity. Unfortunately no further research on these models is found.

Jordanous (2012) also did a literature research among 75 different papers to identify the state of evaluation in the field of computational creativity. The conclusion of this literature review was that less than a third of the creative systems were evaluated on creativity and there is no consensus in the field on how to do this evaluation. There is no standard methodology that is accepted to evaluate creativity in systems. The systems that were evaluated, were mostly evaluated on both quality and creativity with a skewed focus on quality.

Based on her research, Jordanous created SPECS. This standarized procedure consists of the following steps:

1. Identify a definition of creativity that your system should satisfy to be considered creative. Fourteen components are identified by linguistic analysis and using a subset of these is strongly suggested as a base definition. Also take into account which of these components are specifically important for the domain that system is in. The components are shown in Figure 8.
2. Using step 1, clearly state what standards you use to evaluate the creativity of your system.
3. Test your creative system against the standards stated in step 2 and report the

results. Quantitative as well as qualitative tests could be used.

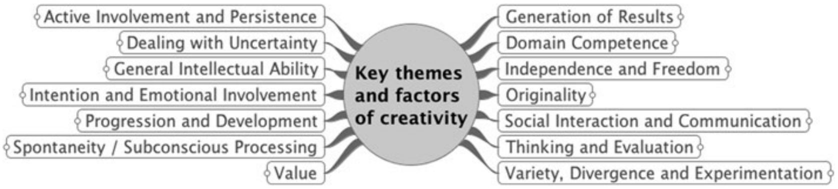

Key themes and factors of creativity
Active Involvement and Persistence
Dealing with Uncertainty
General Intellectual Ability
Intention and Emotional Involvement
Progression and Development
Spontaneity / Subconscious Processing
Value
Generation of Results
Domain Competence
Independence and Freedom
Originality
Social Interaction and Communication
Thinking and Evaluation
Variety, Divergence and Experimentation

Figure 8: The fourteen components of creativity identified by Jordanous (from Jordanous, 2012)

To illustrate the use of the SPECS-framework, Jordanous provided an example. Four different musical improvisation systems are evaluated on creativity in this example. In step 1a, Jordanous identifies the domain independent factors. As given by her own advice, the identified factors of Figure 8 are used. Next, in step 1b, domain specific factors are identified. To identify words that are applicable to the domain of musical improvisation, Jordanous surveyed 34 participants with music experience. These participants were asked about their experience in improvisation and the responses were tagged. The tags are then returned to the participants, who had to rate each tag on importance. Based on this, the most important tags are used for the next steps. In step 2 each system is evaluated on the standards of each identified factor from step 1a. In the last step, step 3, evaluative tests are performed for each identified factor of step 1a and 1b. 6 judges were asked to give a mark between 1 and 10 on questions of each factors. Examples of these questions were:

- How is the system perceived by an audience? (Social Communication and Interaction)
- What musical knowledge does the system have? (Domain Competence)
- Does the system get some reward from improvisation? (Intention and Emotional Involvement)

The results of these ratings are then discussed and the most creative system is picked. Also a general review of each system is provided based on the scores of the judges.

3.5 Natural Language Processing

Natural language processing is the study of computationally understanding and manipulating text with different technologies. To accomplish this understanding different approaches are studied and different techniques are developed. In this section the different approaches are discussed.

Stemming and lemmatization

Stemming is the technique that finds the ‘root’ of a word (Jivani, 2011). The stemmer can find base forms with a lookup tables, but also pre- and suffix striping algorithms can be used. In addition to this, substitution could be used (e.g. substitute -lies with -ly, and then strip -ly to get the base). The root is useful to look up the meaning of a word and the different forms of the word could be used to derive the relations within the sentence. A more advanced way of stemming is lemmatization. Stemming is able to derive the verb *walk* from the past tense *walked*, but is not able see whether the word *meeting* is a verb or a noun. However, lemmatization can be used to do this. This technique uses part of speech tagging and dictionary lookups to determine the lemma of a word. This makes lemmatization slower than stemming, but more accurate. Next to this, lemmatization is able to derive the lemma *go* from word *went*, while a stemmer will not be able to derive a lemma from this word (Jivani, 2011).

Techniques that apply stemming or lemmatization make use of different kind of algorithms to become more accurate. Stochastic algorithms for example, give the probability of having the right 'root' of a word. Current challenge of these algorithms is achieving less false positives (overstemming) and false negatives (understemming).

Parsing

Parsing is the technique that tries to determine the function of a word in a sentence. Different parsing trees can be constructed to determine the possibilities of the categories per word. This method could use the relationships with other words in the sentence or the form in which the word appears. Next to this, the right category or function is chosen based on statistics in most modern parsers. *Shallow parsing*, also called *chunking*, tries to group constituent parts of a sentence and then determine the category or function of a group and is an important part of systems that perform information extraction or summary generation (Daelemans, Buchholz, & Veenstra, 1999). Examples of constituent parts are noun phrases and verb phrases.

Part-of-Speech tagging

Part-of-Speech (POS) tagging is concerned with the tagging of words or a group of words. With this tagging the words are identified as nouns, verbs, adjectives, etc. This is a difficult task as words could have multiple forms while they are spelled the same. A bank could be an institution and thus a noun, but it could also be a form of the verb to bank. POS itself does not solve a particular problem, however it helps to as a first step to resolve other issues. POS tagging could be done rule based or in a stochastic form. A rule-based form would require so many rules that this would not be possible without automatic program. The stochastic approach calculates the frequencies of the options based on a corpus and decides based on these number what tag should be given. Current part-of-speech taggers already have a 97% accuracy in identifying tokens (Manning, 2011). However the accuracy of completely right tagged sentences is much lower, around 55%.

Named Entity Recognition

Named Entity Recognition (NER) is concerned with locating and classifying important nouns and proper nouns such as names, organizations and dates in a sentence (Mohit, 2014). This type of information extraction is challenged with identifying the complete entity and fails when two parts of the entity are recognized seperately. If John Wick is identified as [John] and [Wick], instead of [John Wick] for example, the prediction wrong. Next to this, it is difficult to do the right tagging across different domains. NER is useful for classifying text or to imporove search algorithms.

Semantic Role Labeling

Semantic Role Labeling (SRL) is concerned with labeling relations within sentences. These relations are indicated with arguments that are related to a predicate. Representations that show the abstract role of the arguments are called semantic roles (Jurafsky & Martin, 2018). These can be abstract, specific or somewhere in between. Specific roles are called deep roles. These roles are specific for a certain event. For example in the sentence *John buys a book*, John is a `buyer` and a book is the `thing bought`. The `buyer` and the `thing bought` are deep roles in this case. The roles are called thematic roles when the roles are assigned in a general way. In the previous example, John would be an `Agent` and a book would be a `Theme`. All sets of roles are thus called semantic roles, but depending on the level of role labeling the terms thematic roles and deep roles are introduced.

Research showed that it is difficult to come up with a definitive set of semantic roles to store and use for further research (Jurafsky & Martin, 2018). The definitive list could come short to new cases which have not been identified yet (Màrquez, Carreras, Litkowski, & Stevenson, 2008). Next to this, some SRL programs are more specific, while others are trying to keep the tagging of the arguments general.

Arguments related to a predicate could show in different realizations of a sentence. The following two sentences show different realizations of arguments in the sentence for example:

- John buys a book for Mary.
- John buys Mary a book.

The order of the theme (a book) and goal (Mary) in this sentence are interchangeable, but the meaning of the sentence does not change. The available options of semantic roles related to a verb is called a thematic grid. Levin (1993) created verbclasses based on the thematic grid of 3100 English verbs. These classes are stored in VerbNet.

The Proposition Bank (PropBank) is a resource with annonated sentences. These sentences are annonated with numbered arguments as semantic roles and are saved in frame files. The numbered arguments usually refer to typical thematic roles such a `proto-agent` and `proto-patient`. Next to this, non-numbered arguments are used. These ArgMs represent modifications or adjunct meanings (Jurafsky & Martin, 2018). Examples of these ArgMs are ArgTMP and ArgLOC, which respectively identify time or location. PropBank uses verbs as predicates to relate arguments to.

NomBank is a lexical resource that is similar to VerbNet but based is based on nouns. The nouns are used as predicate and parts of the sentence are related to these nouns.

FrameNet is another widely used lexical resource next to VerbNet, PropBank and NomBank. Instead of focusing on verbs, FrameNet focuses on frames. A frame is the holisitc background knowledge that relates a set of words. Frame elements specify which semantic roles exist within a frame. For example the frame `commerce_buy`[3] is defined as follows: *These are words describing a basic commercial transaction involving a Buyer and a Seller exchanging Money and Goods, taking the perspective of the Buyer. The words vary individually in the patterns of frame element realization they allow. For example, the typical pattern for the verb BUY*: *Buyer buys Goods from Seller for Money.* The frame elements identified in a frame are parts that could exist in such activity. In the case of the frame above the frame elements are `buyer`, `goods`, `seller` and `money`. This frame identifies all the sentences that are related to buying something. So when John buys a car, John is identified as the `buyer` and the car as the `good`. Notice that not all frame elements have to be in the sentence to be part of the frame.

Currently SRL is usually approached in three or four steps according to Màrquez et al. (2008). The first step identifies the different arguments that are related to the predicate. These argument-candidates are then scored based on possible role labels. Finally, arguments are identified and classified with the help of these scores. Multiple scoring methods can be combined to provide a joint scoring to enhance the accuracy of the SRL could be added as additional step between the second and third step.

[3] https://framenet2.icsi.berkeley.edu/fnReports/data/frameIndex.xml?frame=Commerce_buy - visited on 19 October 2018

Natural Language Generation

Natural Language Generation (NLG) is concerned with creating new texts based on input. Reiter and Dale (2000) identified the following stages of NLG:

1. Content Determination: This step is concerned with deciding what to write down. What aspects are important and which aspects could be ignored?
2. Document structuring: How is the document going to be structured? The order of the text is determined and a title, if applicable, is created.
3. Aggregation: To read the text more fluently separate sentences could be combined.
4. Lexical choice and referring: Choose which words are going to be used. This seems odd as the sentences are already placed and structured, however a check is needed to see whether the correct words are used or not. Next to this, if the texts constantly goes about the same, e.g. a person like John, it could be referred back to, e.g. he instead of John.
5. Realization: The text is finished and the output can be created.

3.6 Insights

Based on the literature, requirements synthesis can be placed within the process of requirements engineering. The three dimensions of Pohl (1994) for example, show that synthesizing requirements will have effect on the specification and representation dimensions. The specification of new requirements will be synthesized with the program, and these synthesized requirements are preferably represented in a semi-formal way. Therefore, user stories come in handy. The standard notation makes it easier to apply NLP techniques and synthesize requirements in this standard notation.

Taking creativity into account while synthesizing requirements is a difficult and new problem. Requirements engineering is in most literature approached as a creative process with lots of dimensions, contexts, and factors. These dimensions, contexts and factors all have an influence on the creativity of the RE process. However, it is not studied how the process is influenced, as studying just one dimension, context or factor is difficult without having a change in the others. A part of the RE process in this thesis is synthesized, with the goal to create a creative product. Boden's (2009) ways of creativity are used to do this. With a combinational and exploratory creativity, synthesized creativity is studied and applied earlier in other domains (Manurung et al., 2008; McCorduck & Cohen, 1991), and therefore, these ways are used as well in this thesis. SRL is a promising NLP-technique that could be used to make meaningful combinations, based on thematic roles. The exploratory creativity could be applied by looking into alternatives that are semantically close to the original statements, but do differ enough to make a change.

Assessing the creativity of the synthesized requirements or other creative artefacts is not yet standardized as studied by Jordanous (2012). An approach to do this, is letting experts assess creativity on several dimensions. The three dimensions that are identified for this thesis are usefulness, surprisingness and novelty. The definitions in Table 4 are used for these dimensions and are based on the creative product which is discussed in the paper of Nguyen and Shanks (2009).

Table 4: Creativity dimensions to evaluate

Dimension	**Description**
Usefulness	A requirement is useful when it specifies a workable and effective way to solve a problem.
Novelty	A requirement is novel when it defines a new and original business requirement that helps develop innovative solutions.
Surprisingness	A requirement is surprising when it describes something that is unusual, unexpected or something that ”may shock or amaze us”.

4 Requirements Synthesizer

Based on the insights of the previous section, several steps to synthetically extend a set of requirements are identified. First, the original set of requirements needs to be analyzed. This means that the semantic parts of every requirement are identified and stored, so they can be used later in the process. After analyzing the requirements, new requirements need to be synthesized while taking into account that these need to be creative. The focus in this research lays on combinational and exploratory creativity as these are already studied and applied in research in other domains (Manurung et al., 2008; McCorduck & Cohen, 1991).

Combinational creativity is achieved by combining different parts of requirements in this thesis. This can be done with requirements from one product or requirements from similar products. These combinations could result in new features of existing products, like adding a camera to a phone. Both products existed separately but were combined later to one product.

The exploratory creativity is achieved by looking into the terms that are used in the existing requirements and find related terms that could also fit into the requirement. This change will also extend the set of requirements and could result in unexpected and creative requirements.

The method for synthesizing requirements that we employ is shown in Figure 9 and described as follows:

1. Start with a list of requirements of a specific domain. This requirements in this list are formatted in User Stories as described in Section 3.1
2. Use semantic role labeling to identify semantic parts of every requirement. Each part that is identified is labeled with a thematic role and stored. This process is described in Section 4.2.
3. Identify alternatives of stored parts that are saved in the previous step. Also label these parts with a thematic role and add them to the list of parts that can be used to synthesize requirements.
4. Combine parts from step 2 and 3 based on their thematic roles.
5. Make sure the requirements' notation is correct and a user story as described in Section 3.1 is created.
6. Output a list of synthesized requirements.

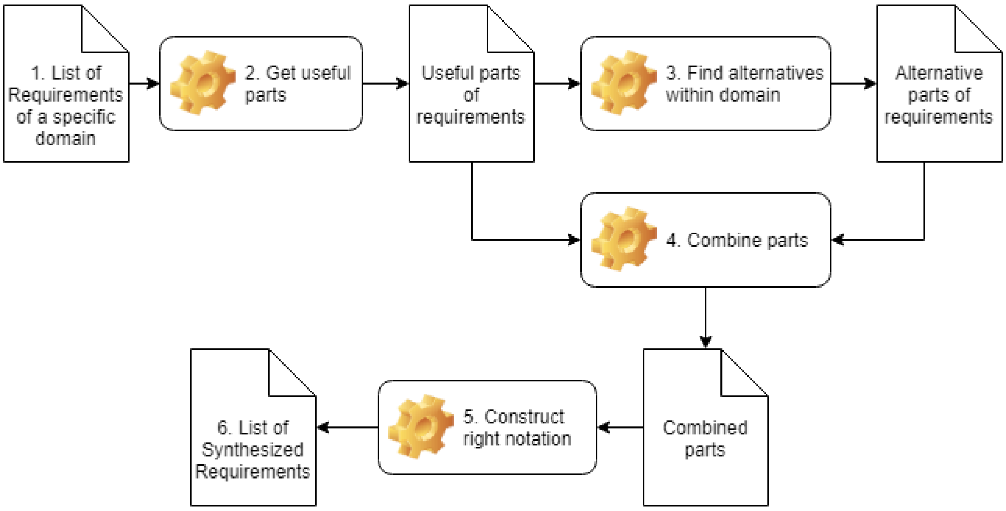

1. List of Requirements of a specific domain
2. Get useful parts
Useful parts of requirements
3. Find alternatives within domain
Alternative parts of requirements
4. Combine parts
Combined parts
5. Construct right notation
6. List of Synthesized Requirements

Figure 9: Conceptual approach of synthezing requirements within a domain

We have developed a program that synthesizes requirements based on this conceptual approach. This program is made in Python, as this is dynamic programming language in which can be coded in fewer steps compared to Java or C#. The libraries that are used for the Requirements Synthesizer are described first in Section 4.1. With these libraries the tool is created, which is described in Section 4.2 and 4.3. At the end of these sections an example is provided to show how the code works. Next to this, the code is publicly available on Github[4].

4.1 Libraries

Different libraries are needed to make the program in Python, as the experience and time is not available to construct every part manually. Libraries are needed to handle the SRL, to explore the domain and to construct the requirements. By using a set of libraries, expertise from different resources can be used, without having to study and create the necessary elements of the program yourself. Choosing the right resources is important, to make sure the program uses state of the art solutions and works correctly.

Semantic Role Labeling

To do the SRL of the program, different options are evaluated. A promising option was the CogComp pipeline (Punyakanok, Roth, & Yih, 2008) with a demo that shows arguments with assigned roles related to a verb, which is useful for this thesis[5]. This SRL-module is programmed in Java, however a pipeline is created to use a version of the module in python. A pipeline in this case makes it able to use Python commands to communicate with the original code in Java, so that commands in Python are executed the same as the commands in Java. Unfortunately, this module is resource intensive to perform SRL on a local computer (approximately 5 minutes per sentence[6]). Next to this, the module only gives numbered arguments that are related to the predicate without the thematic roles. The demo did have a feature that assigned roles to the arguments. Unfortunately, the module responsible for assigning these thematic roles, VerbSense, is not available in the python pipeline. Because of these reasons, the CogComp pipeline is not used in this thesis.

PractNLPTools is a python wrapper for SENNA and the Stanford Dependency Extractor, and is able to identify the arguments related to the verbs in the sentence. These arguments are numbered from A0 to A5. SENNA is a tool written C and has different functionalities such as POS tagging and SRL (Collobert et al., 2011). The Stanford Dependency Extractor is one component from the Stanford Parser and is used to perform the syntactic parsing of the PractNLPTool (biplab-iitb, 2016). Since this function is not used in the created program, the Stanford Parser is not relevant for this thesis. However, the accuracy of the SENNA SRL tool is relevant. According to it's website (Collobert, n.d.) the F1-score of this tool is 75.49%, which is measured with a CoNLL 2005 benchmark (Carreras & Màrquez, 2005). The role labeling of practNLPTools is split up verb, so the arguments are identified related to a single verb. Also the results are returned in a structured dictionary in Python, which can easily be processed in next steps. Another pro for this library is that it is easy to set up and does not have many dependencies on other python libraries, which makes the installation of the library easy.

Another state of the art solution that is available is AllenNLP[7]. This library requires

[4] https://github.com/RELabUU/Requirements-Synthesizer

[5] http://cogcomp.org/page/demo_view/srl - Demo visited on 15 October 2018

[6] Computer specifications are: Intel Core i7 4500U @ 1.80GHz CPU; 8,00GB Single-Channel DDR3 @ 798MHz RAM; 2048MB NVIDIA GeForce GT 730M GPU; 119GB SAMSUNG MZ7TD128HAFV-000L1 SSD

[7] https://allennlp.org/ visited on 15 October 2018

python 3.6 or 3.7 and is built onto the PyTorch library. When it was tried to install this library multiple errors showed, while a fresh python 3.7 install was used as advised. The first error shows that the right version of the torch library could not be found. A suggestion was to update the pip module, that is used to automatically install libraries. After a pip upgrade this error could be resolved. Unfortunately, this was not the case. So a manual install of the torch library was tried to fix this. However, the torch, or pytorch, seemed to have little support for a Windows machine on which the program was tried to install. After multiple tries with and without conda installations -conda is a tool that helps with the installation of non-python libraries too-, it was decided that it was to much effort to get this library running as well. Therefore the efforts to use this library are stopped and practNLPTools is chosen to use.

PractNLPTools is a semantic role labeler that identifies the arguments related to verbs. These numbered arguments however, will only make sense when they can be identified with a thematic role. SemLink is used to accomplish this. SemLink combines different lexical resources such as VerbNet, PropBank, FrameNet and OntoNotes and provides a mapping so that strengths of these resources can be used (Bonial, Stowe, & Palmer, 2013). Based on the mappings of these resources, the numbered arguments that are identified by practNLPTools can be linked to thematic roles with the help of the verb that is used to identify the numbered arguments. These thematic roles provide a general meaning to the part text that is identified, which can be used to construct new requirements.

Exploratory creativity

Not only combinational creativity is tried to be applied, but also exploratory creativity. In order to apply this, semantic similar alternatives to the thematic parts that are identified with SRL need to be found. Sense2vec identifies the semantic relationship between two words based on a training corpus. Two words are used as input and a ratio between 0 and 1 is given to show the semantic relationship. This process makes use of word clustering and is shown in Figure 10.

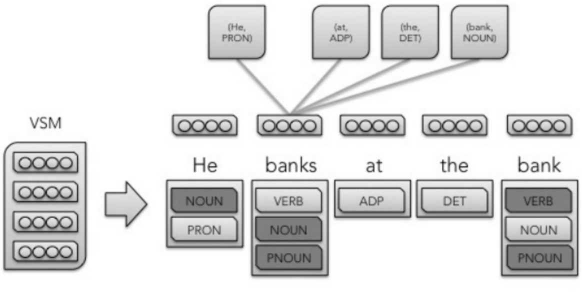


Figure 10: sense2vec word clustering (Trask, Michalak, & Liu, 2015, p. 3)

Based on word clustering, words that have semantic similarity can be found with context-sensitive word vectors. The library uses the pipeline English model and Word2Vec component of Spacy[8] to do this. A function of Sense2Vec makes it possible to find words that are semantically related to an input word. When a word or part of text is searched for, Sense2Vec shows a result of all the semantic similar words with a percentage that indicates how related the words are.

[8] https://spacy.io/ - visited on 15 October 2018

Constructing the requirement

To construct the requirement the different parts need to be 'glued' together. This combining of parts is just combining the different strings in python to one string: the requirement. However, it has to be checked whether this requirement is grammatically correct or not. To do this, the library grammar-check[9] is used. This library is originally build for python 3.2, but with the lib3to2 library it is also usuable in python 2.7. Grammar check identifies the errors in the sentence and is able to suggest ways to solve the errors. It can also automatically solve the grammatical errors by applying the most likely solution.

Unfortunately no libraries were available that could check whether a sentence is complete or not as this is difficult to determine for a computer. Therefore, this problem remains open and is not solved in this thesis.

4.2 Analyzing Requirements

As discussed the requirements should be analyzed first. All the analyzed requirements need to be saved in a text file with a new requirement on every line of the file. Next to this, every requirement is structured as an user story:

As a [Agent], I want to [requirement].
or
As a [Agent], I want to [requirement], so that [reason]

In Figure 11 the process of analyzing the requirements of a text file is shown visually. The [Agent] in the user story is always known and saved. If a [reason] part exists in the user story, this part is removed to simplify the analysis of the requirements and get more useful thematic roles for the generation of requirements. This limits the usefulness of the application, as the reason behind a requirement is an important part of the requirement. However, to make the program more accurate it is needed to remove the [reason] part.

PractNLPTools is used to find the different semantic parts of the [requirement] part in the user story. To assign the thematic roles to the identified numbered arguments the VerbNet and PropBank link of SemLink is used. These thematic roles and their corresponding semantic parts are stored in a file. This file can be used to synthesize new requirements. A copy of the full code is available on Github[10].

Example of analyzing a requirement

In this example, analyzing the requirement "*As a Publisher, I want to publish a dataset, so that I can view just the dataset with a few people.*" is shown. Figure 11 shows that this user story is taken from the file with requirements that is taken as input. With this requirement the following steps are executed:

1. The [Agent] is taken from the requirement first and stored in the file with semantic parts. In this example *Publisher* is stored as an Agent. The *As a Publisher,* part is removed from the requirement before moving on to the next step.
2. To simplify the process of analyzing requirements and have useful parts without reasoning available in the file with semantic parts the reason is removed from the requirement as well. Thus "*, so that I can view just the dataset with a few people.*" is removed and the remaining requirement is "*I want to publish a dataset*".
3. To make sure the SRL is performed on the non-standard part of the requirement, the *I want* part of the remains are removed too. By doing this, SRL can be applied on "*to publish a dataset*".

[9] https://github.com/viraja1/grammar-check - visited on 19 October 2018

[10] https://github.com/RELabUU/Requirements-Synthesizer

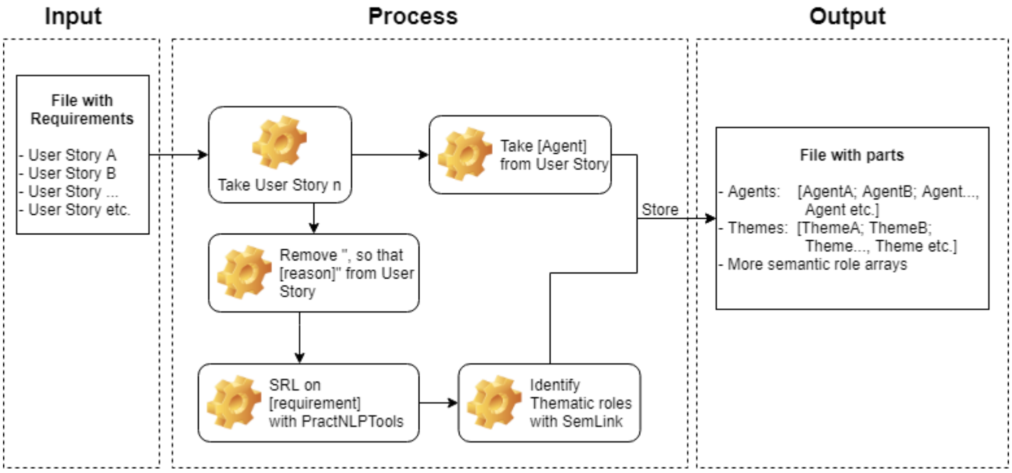
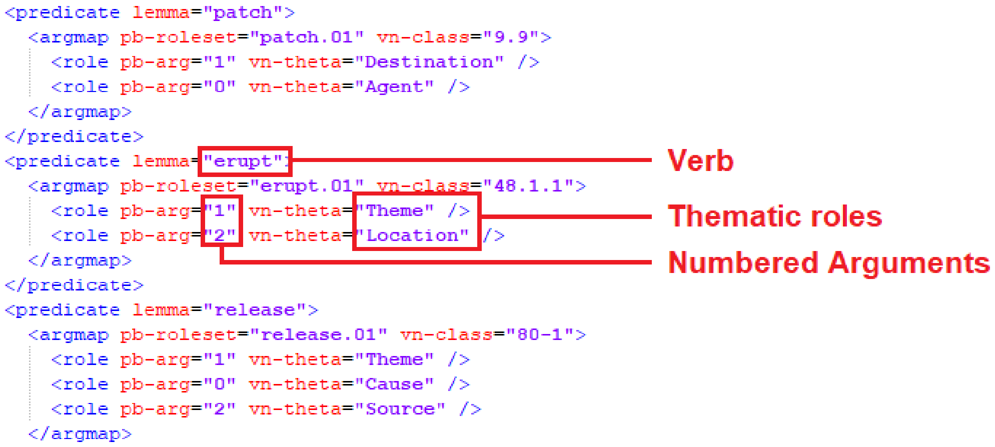

Input
Process
Output
File with Requirements
- User Story A
- User Story B
- User Story ...
- User Story etc.
Take User Story n
Take [Agent] from User Story
Remove ", so that [reason]" from User Story
SRL on [requirement] with PractNLPTools
Identify Thematic roles with SemLink
Store
File with parts
- Agents: [AgentA; AgentB; Agent..., Agent etc.]
- Themes: [ThemeA; ThemeB; Theme..., Theme etc.]
- More semantic role arrays

Figure 11: Process of analyzing requirements

4. PractNLPTools labels the verb and the arguments that are available. This results in the labeled verb and an argument: *to* $[publish]_V$ $[a\ dataset]_{A1}$
5. With SemLink the right thematic role is assigned to the numbered argument based on the verb. In this case *a dataset* is identified as a theme. The verb *publish* and theme *a dataset* are stored in the file with semantic parts as such. This process works as follows:
   (a) The lemma of the verb is searched in the SemLink file that maps VerbNet to PropBank.
   (b) The numbered arguments are mapped on the thematic roles in the file as shown in Figure 12.
   (c) Every numbered argument is saved as the thematic role that is identified in the SemLink file.



```
<predicate lemma="patch">
  <argmap pb-roleset="patch.01" vn-class="9.9">
    <role pb-arg="1" vn-theta="Destination" />
    <role pb-arg="0" vn-theta="Agent" />
  </argmap>
</predicate>
<predicate lemma="erupt">
  <argmap pb-roleset="erupt.01" vn-class="48.1.1">
    <role pb-arg="1" vn-theta="Theme" />
    <role pb-arg="2" vn-theta="Location" />
  </argmap>
</predicate>
<predicate lemma="release">
  <argmap pb-roleset="release.01" vn-class="80-1">
    <role pb-arg="1" vn-theta="Theme" />
    <role pb-arg="0" vn-theta="Cause" />
    <role pb-arg="2" vn-theta="Source" />
  </argmap>
```

Figure 12: Snapshot of a part of the used SemLink file that maps the numbered arguments on thematic roles

4.3 Synthesizing Requirements

New requirements can be synthesized after analyzing the original requirements. This process is shown in Figure 13. To enhance the creativity of these requirements, also exploratory creativity is used to extend the list of saved parts for requirements. For each semantic part alternatives are identified with the sense2vec library, by replacing the nouns in the different parts with alternative nouns. Only nouns were replaced as different parts stopped to make any logical sense when verbs, organizations, or other kind of words were replaced.

The alternative that is chosen is the first alternative with a score below .75. This score is chosen because of two reasons. To prevent that the alternative word is too close related to the original word, which would not result in an exploratory change, the score should be preferably below .75. However, to prevent that the alternative has too little semantic similarity, a score as close as possible to .75 is chosen. The score is determined with a manual exploration. Examples of nouns and related nouns are shown in Table 5. As shown in the table, the first semantic similarity given by Sense2Vec is often a variation on the input. The alternative around the the .85 score is sometimes sufficient but is in a lot of cases still similar to the input. The first alternative below the .75 score show different nouns in most cases. If there is no alternative within the first 250 alternatives with a score lower than .75, the 250th alternative is chosen, because searching for more alternatives would decrease the performance of the application too much.

Table 5: Nouns and their semantic similarities as identified with Sense2Vec

Noun	**First alternative**	**.85 alternative**	**first alternative below .75**
dataset	data set (0.905)	source data (0.852)	database table (0.749)
client	other client (0.836)	other client (0.836)	server admin (0.749)
data package	data plan (0.851)	data plan (0.851)	basic phone (0.749)
data	other data (0.930)	most data (0.849)	good data (0.749)
world	entire world (0.905)	world- (0.841)	civilization (0.749)
requirement	required (0.866)	other requirements (0.851)	special circumstances (0.749)
thesis	dissertation (0.927)	PhD thesis (0.857)	academic discipline (0.749)
public information	public info (0.890)	public record (0.844)	personal data (0.747)
dashboard	dash board (0.796)	dash board (0.796)	little screen (0.749)

These alternatives are added to the list of viable parts for the requirements. Based on the list with these parts new requirements are created. This starts by picking a random verb from the list with the standard random library of Python. Based on this verb, the verbfamily is identified via the SemLink files that were used while analyzing the requirements. This verbfamily is then looked into, to construct requirements based on the given example structures that are provided by the VerbNet xml files. A requirement is built like a formal user story again and starts with *As a*. The user story is only valid if the an agent is provided next. Therefore, user stories based on verbs or structures that do not start with a agent will not be created. If the agent is part of the structure, the requirement is build up like *As a [Agent], I want to*. After this part the rest of the structure is completed with the verb and the remaining thematic roles that are needed by the provided structure. This results in a complete user story without reasoning:

As a [Agent], I want to [requirement].

At last the grammar of the requirement is checked with the grammar_check library. If there are any grammar mistakes found, these are automatically fixed with the first suggestion of the library. After this grammar check the new requirement is written to a text file, until the file contains a minimum of 50 requirements. This is a minimum, as more requirements could be constructed based on the random picked verb that is used.

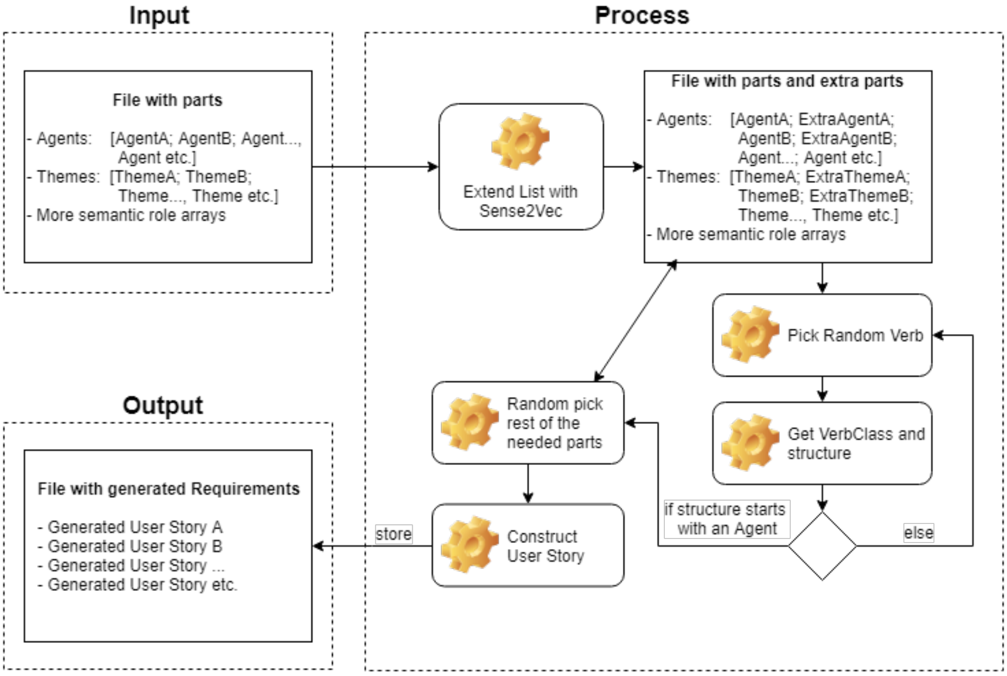

Input
Process
File with parts
- Agents: [AgentA; AgentB; Agent..., Agent etc.]
- Themes: [ThemeA; ThemeB; Theme..., Theme etc.]
- More semantic role arrays
Extend List with Sense2Vec
File with parts and extra parts
- Agents: [AgentA; ExtraAgentA; AgentB; ExtraAgentB; Agent...; Agent etc.]
- Themes: [ThemeA; ExtraThemeA; ThemeB; ExtraThemeB; Theme..., Theme etc.]
- More semantic role arrays
Pick Random Verb
Get VerbClass and structure
if structure starts with an Agent
else
Random pick rest of the needed parts
Construct User Story
store
Output
File with generated Requirements
- Generated User Story A
- Generated User Story B
- Generated User Story ...
- Generated User Story etc.

Figure 13: Process of synthesizing requirements

Example of synthesizing a requirement

In this example, synthesizing a requirement is shown. Figure 13 shows that an input file with semantic parts is used to do this. This input file is the output file of analyzing the requirements. More semantic parts are identified with the existing semantic parts with Sense2Vec. For example, similarities of the agent *Software Engineer* are identified. The first similar noun that comes up with a score lower than .75 is *Games Developer*. This noun is added to the file with semantic parts. Consider table 6 as the available semantic parts.

Table 6: Available semantic parts

Verb	Agent	Theme	etc.
publish	publisher	a dataset	
provide	software engineer	any database	
include	platform administrator	the properties	
	games developer	various properties	
etc.	etc.	etc.	etc.

With the available semantic parts a requirement is synthesized in the following way:

1. First a random verb from the list is picked. In this case *provide* is taken.
2. The verbfamiliy of provide is identified and the corresponding VerbNet file is looked up. *Provide* belongs to the verbfamily 13.4.1 and the details of this verbfamily are shown in an xml file provided by VerbNet, in this case `fulfilling-13.4.1.xml`. In this file the a structure to make a sentence with this verb is provided. The structure that is given in the file is: `Agent - Verb - Theme - Preposition "to" - Agent`.
3. The structure starts with an Agent, so it is usable to create a requirement. For each part of the structure a corresponding semantic part is taken from the file with available parts: Agent, Software Engineer; Theme, the properties; Agent, Platform Administrator.
4. The parts are taken and combined to create a complete requirement: *As a Software Engineer, I want to provide the properties to the Platform Administrator*
5. Finally a grammar check is done on the requirement before it is saved into the output file.

5 Evaluation of the Requirements Synthesizer

In this section the evaluation of the artefacts is explained. The evaluation design is described first. This starts with a description of the applications that are used as input for the Requirements Synthesizer. Then, the questionnaires are described which contain human generated and synthesized requirements based on these applications. Hypotheses are given next, based on the dimensions that are evaluated with the questionnaire. The results are discussed after the description of this evaluation design.

5.1 Evaluation Design

Two applications are used to synthesize requirements. The first supervisor provided a list of available requirements of applications (Dalpiaz, 2018). This list is used in earlier research. Based on this list two applications that are publicly available are chosen to use for synthesizing requirements. One application with similar applications from the list provided by the first supervisor, Datahub, and one application from the list provided by the first supervisor without similar applications, Recycling 101. Synthesizing requirements based on two applications is done to study whether requirements or information from other applications result in different results or not. This could help to determine whether requirements from one application are sufficient to synthesize creative requirements or it is more useful to use requirements from applications from the same domain as well.

The input requirements of both applications can be found in Appendix A.1 and A.2. The Datahub requirements are the first 67 lines of the 250 requirements that are used as input. The other 183 requirements, are requirements of the other three applications. The number of requirements of the recycling app is 51. Both files are used as input for analyzing and then synthesizing requirements. The output of the synthesized requirements is shown in Appendix B.1 and B.2.

Datahub

Datahub allows to store, share and look up data. The Datahub application provides tools to do so, aiming to improve the quality of the available data. The goal is to have a place to have high quality datasets that can be shared with others. One of the main features of Datahub is creating a datapackage of a set of data files. Basic info such as the author, license, list of files, data structure, etc. can be added to the package. Datahub supports this to have a ”very simple, web friendly, standardized and extensible” approach (https://datahub.io/docs/data-packages). While uploading data in data packages to Datahub, Datahub is able to validate whether the uploaded data is stored in the way as described in the meta-data of of the data package. For example, this validation includes an automatic check of whether all columns are filled in. Next to creating these data packages, Datahub supports a way to publish and deploy them. Applications are available for Windows, MacOS and Linux and can be used to publish the data. Next to this, users can configure with whom you want to share the data. This could be public or within a specified group. The data available can be downloaded and used in multiple ways. A script in python, R or JavaScript for example could retrieve data packages from Datahub and use them in an application. Datahub can be visited via https://datahub.io. Applications similar to datahub based on the dataset of Dalpiaz (2018) are OpenSpending, Frictionless and CASK. These applications are chosen based on their short description. The short description of all four applications can be found in Table 7.

Table 7: Applications and their short description

Application	Short Description
Datahub	Platform to find, share and publish data online
OpenSpending	Website for viewing fiscal data submitted by others
Frictionless	Repository management for data insights
CASK	Dataset management system

Recycling 101

The other application is Recycling 101, an application that helps you to find the closest destination to recycle your used products in New York. The user will use the application to look up where they can get rid of their used products. They will enter the category of their product, their preferred drop-off date and their zipcode. As a result they get the nearest recycling center that handles the goods they offer. This application is made to improve the communication of the recycling centers within New York. Recycling centers themselves can add their opening hours, contact details and recycling possibilities on the application so they can be found by users of the application. Next to this, users can make a profile and add their favorite recycling centers to their own profile. The application can be visited via: https://warm-beach-37724.herokuapp.com/.

Questionnaires

Two questionnaires are constructed with the requirements. These questionnaires firstly gather some demographic information. Because most of the respondents were expected to be students, the age groups are around the twenties. Next to this, a question is asked to check if the respondents have any experience with requirements engineering to make sure the responses are valid. Also a difference in gender is tried to measure, but as many people of the available target group are male, a measurable difference is not expected.

After the the introduction questions, the participants are informed about the application and user stories. This info is presented in separate sections to make sure all the text is read and a respondent does not skip to the questions without reading the needed context. After this information the respondents are asked to asses the requirements on three dimensions.

To assess the three dimensions, twelve requirements are selected. Three original requirements are incorporated in the questionnaire to have a basis of the assessment of the requirements by the respondent. Next to the three original requirements, nine synthesized requirements are selected based on two criteria. More synthesized are used, because the results of the out the Requirements Synthesizer can go either way, while human made requirements are made by experts and do not differ much in quality The first one is that the requirements should be full sentences. Next to this the first requirement of every random verb is used. The created program makes several requirements based on the structures of one verb. Of these structures the first requirement is picked and the rest is skipped until the next random verb in the requirements. The total of twelve requirements are rated on three dimensions of creativity. A four point likert scale is used to asses the dimensions of creativity of the different requirements to avoid 'neutral' responses. Also a motivation of the ratings is asked to discover the way the respondents assessed creativity. The three dimensions with their description are shown in Table 8 and their corresponding mark labels of the likert scales are shown in Table 9. The questionnaires can be found in Appendix C.

The questionnaires are distributed within the master Business Informatics Whatsapp group, by personal approach of known fellow students and via a small mailing list of

Table 8: Definitions of the three dimensions of creativity in the questionnaire

Dimension	**Description**
Usefulness	A requirement is useful when it specifies a workable and effective way to solve a problem.
Novelty	A requirement is novel when it defines a new and original business requirement that helps develop innovative solutions.
Surprisingness	A requirement is surprising when it describes something that is unusual, unexpected or something that "may shock or amaze us".

Table 9: Three dimensions of creativity in the questionnaire and their likert mark labels

Dimension	**1**	**2**	**3**	**4**
Usefulness	Not useful	Somewhat useful	Useful	Extremely useful
Novelty	Not novel	Somewhat novel	Novel	Extremely novel
Surprisingness	Not surprising	Somewhat surprising	Surprising	Extremely surprising

thesis students of the first supervisor. As a result, all respondents were students within the field of information science.

Based on dimensions and the applications that are used to synthesize requirements, the following hypothesis are created which are tested after the results were in:

H1. The human generated requirements are the requirements that were used in the requirements document and thus useful for the application. Therefore, these requirements are more likely to score better on usefulness than the synthesized requirements, which have not be accepted by the developers. This results in the following hypothesis: *The human generated requirements are more useful than the synthesized requirements.*

H2. Synthesized requirements based on multiple applications have influences from other applications that could not be relevant or useful for the application where the requirements are synthesized for. This problem is not present at the synthesized requirements based on one application. This results in the following hypothesis: *The synthesized requirements based on one application are more useful than the synthesized requirements based on multiple applications.*

H3. The requirements made by humans are usually focused on the functionality of the application. New and innovative solutions often have a secondary focus. The synthesized requirements are made with a focus on novelty. This results in the following hypothesis: *The synthesized requirements are more novel than the human generated requirements.*

H4. Combining requirements from different applications could lead to new and innovative ideas. Synthesized requirements based on one application has a lower chance on these new innovative ideas, because of the smaller input. This results in the following hypothesis: *The synthesized requirements based on multiple applications are more novel than the synthesized requirements based on one application.*

H5. The synthesized requirements could come up with unexpected combinations, while human made requirements are made based on the logic of the requirements engineer. This results in the following hypothesis: *The synthesized requirements are more surprising than the human generated requirements.*

H6. Combinations and similarities from other applications from a domain could lead to requirements that were not based on requirements of the original application. The synthesized requirements of one application are expected to be more related to this application and therefore less surprising than synthesized requirements from multiple applications. This results in the following hypothesis: *The synthesized requirements based on multiple applications are more surprising than the synthesized requirements based on one application.*

5.2 Results and Analysis

In this section the results of the questionnaires are discussed. Each questionnaire is filled out by 10 people, of which 10% of the respondents is female and 90% is male.

Of the Recycling app questionnaire 1 respondent is aged between 18 and 21 years old, 7 respondents are aged between 22 and 25 years old, 1 respondent is aged between 26 and 29 years old and 1 respondent is older than 30 years old. The highest completed level of education of 2 respondents is their first year bachelor, 7 respondents completed their bachelor's and 1 respondent completed a master's degree. 30% of the respondents assessed themselves as somewhat experienced with software requirements and 70% were assessed as experienced in software requirements.

The questionnaire of the Datahub requirement has slightly different demographics. 7 respondents are between 22 and 25 years old and 3 respondents are between 26 and 29 years old. 1 of the respondents finished its first year bachelor, 6 respondents finished their bachelor's degree and 3 respondents finished their master's degree. Although the average education level of the Datahub respondents is higher, they asses themselves a bit less experienced with software requirements on average. 2 respondents state that they have no experience with software requirements, 4 are somewhat experienced with software requirements and 4 assessed themselves experienced on software requirements.

The requirements used in the Recycling and Datahub questionnaire are stated in respectively Table 10 and 11. The human made requirements are indicated with a HR label and the synthesized requirements, made with the program described in Section 4, are indicated with a SR label. The requirements are ordered based on the way they were shown in the questionnaire.

Table 10: Requirements used in the questionnaire and their labels of the Recycling 101 application

Label	**Requirement**
HR1	As a user, I want to click on the address.
SR2	As an employee from the HR department, I want to keep usage stats and locations
SR3	As a superuser, I want to handle usage stats and immediate area
HR4	As a user, I want to add donation centers as favorites on my profile.
SR5	As a company, I want to keep specific users based on IP address
SR6	As a bad employee from the HR department, I want to handle all locations of landfills
HR7	As an admin, I want to be able to block specific users based on IP address.
SR8	As an employee from the HR department, I want to keep usage stats and locations
SR9	As a senior executive, I want to keep all locations of landfills
SR10	As an employee from the HR department, I want to handle my private info
SR11	As a company, I want to keep specific users based on vpn client
SR12	As an admin, I want to keep usage statistics and locations

Table 11: Requirements used in the questionnaire and their labels of the Datahub application

Label	Requirement
HR13	As a Visitor, I want to sign up via Github or Google.
SR14	As a proprietor, I want to take key metrics about usage such as users and API usage
SR15	As a Developer, I want to set the dataset code
HR16	As a Publisher, I want to see real examples of published packages.
SR17	As a API user, I want to exist on the computer monitor of all monetary measures
SR18	As a Data Analyst, I want to inject more examples of published packages at runtime
HR19	As a Publisher, I want to data to be validated when I publish it.
SR20	As a hydrator user, I want to search for a database table that was not configured for explore initially
SR21	As a DeveloperDataWrangler, I want to inject few units tests for an app that depends on the interface of a dataset type a Data Package
SR22	As an Industry, I want to start an existing dataset instance
SR23	As a technical sales, I want to search for a database table that was not configured for explore initially
SR24	As a Repository Manager, I want to load the code of a dataset type in my app artifact and create a dataset of that type when deploying the app. at runtime

In Figure 14 the assessment of usefulness for the Recycling 101 application is shown. The hypothesis of this result was that the human made requirements would be assessed more useful than the synthesized requirements. The results indicate that this hypothesis is true. Respondents argued that they assessed usefulness on how actionable the requirement was and if they could imagine the requirement in the application. The synthesized requirements were found to be more vague and therefore less useful. However, as shown in Figure 15 SR2, SR8 and SR12 are rated more useful than the other synthesized requirements and it could be argued that they are also found to be more useful than the human made requirements as they do not have any ”Not useful” marks.

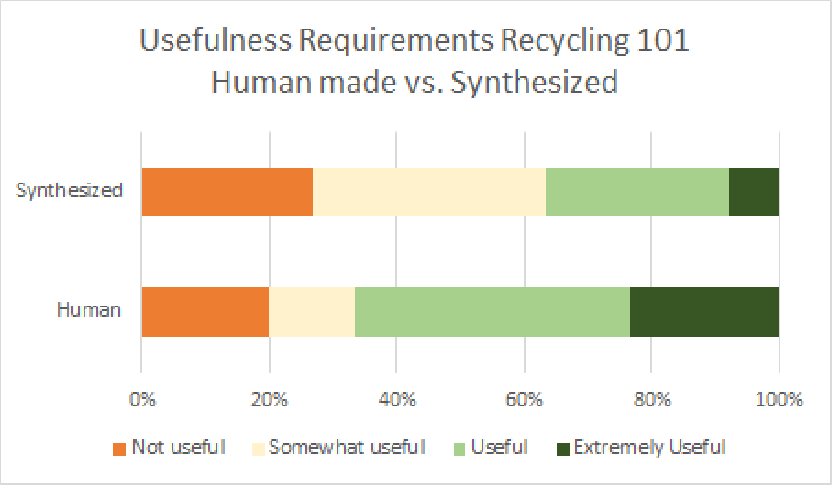


Figure 14: Assessment of usefulness of synthesized and human made requirements for the Recycling 101 application

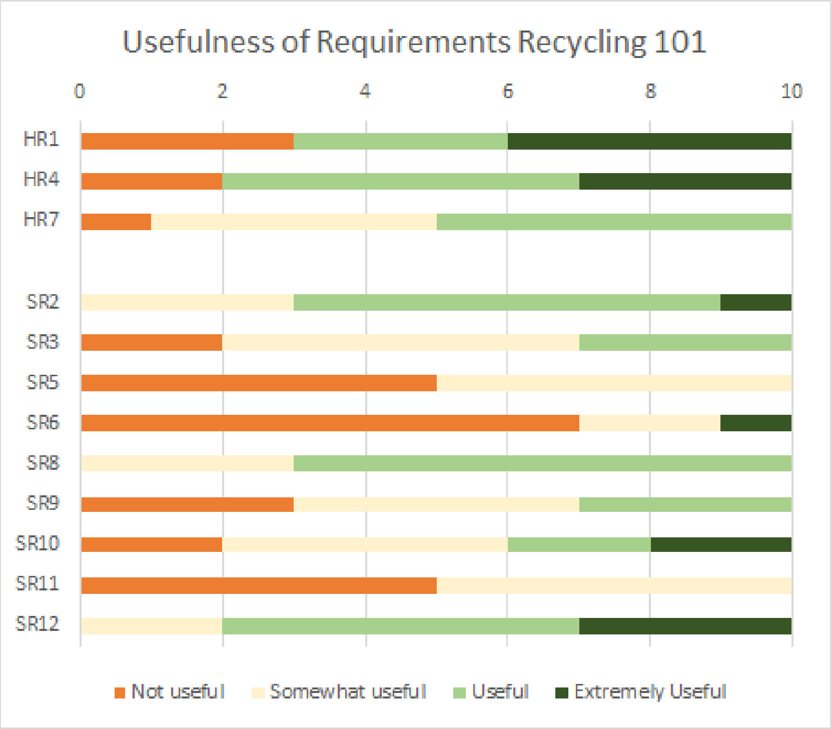


Figure 15: Assessment of usefulness of Recycling 101 application requirements

The assessment of requirements for the Datahub application shows even better that human made requirements are assessed more useful than synthesized requirements (Figure 16). Respondents rated high priority requirements higher on usefulness than requirements that should have a lower priority according to them. Next to this, the requirements are also assessed on how clear the requirement was, a clearer requirement is found to be more useful. The fifth respondent for example described it as follows:

> *"In my understanding, user stories should be atomic, specific, and free of ambiguity (amongst other things). The first user story satisfies all of these aspects: a visitor wants to sign up through Google or Github, making it very useful. Some user stories are specific as well, but not atomic, and some are not that well described or not understandable (like user story 9 and 12)."*

The clear requirement stated by the respondent is the human made requirement HR13, while synthesized requirements SR21 and SR24 are described as not understandable. An outlier of the results is SR14, which is found to be very useful while it is a synthesized requirement. SR17 and SR21 are found to be the least useful by the respondents.

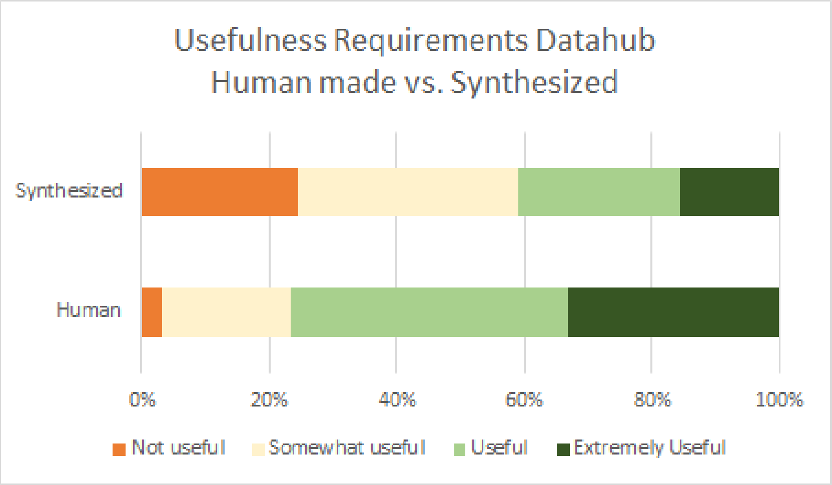


Figure 16: Assessment of usefulness of synthesized and human made requirements for the Datahub application

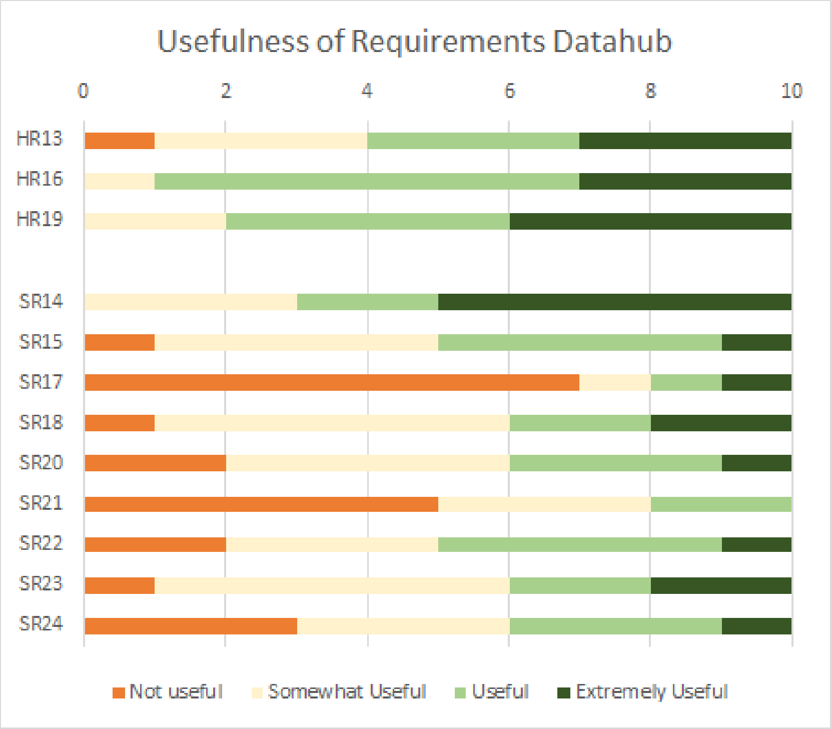


Figure 17: Assessment of usefulness of Datahub application requirements

The novelty of the synthesized requirements was expected to be higher than the novelty of the human made requirements. However, both Figure 19 and 21 show that the there is almost no difference in between those categories. Novelty itself was assessed by focusing on how innovative and new they think a functionality was, a lot of the respondents used their previous experiences with similar applications to see whether they had encountered the functionality before or not. The respondents could not agree on novelty for the human made and synthesized requirements of the Recycling 101 and Datahub application, with the exception for SR3, which is assessed very low on novelty.

When comparing the novelty of the requirements for the Datahub and Recycling 101 application, it is shown that the requirements for the Datahub application are more novel assessed on average than the requirements for the Recycling 101 application for both human made and synthesized requirements. One of the respondents of the Recycling 101 questionnaire described the application as follows:

> *"This whole application does not sound innovative to me. There are many of these kind of apps for other sectors. For example the food sector. You have tons of apps where you can look for specific restaurants in your neighbourhood."*

The fact that the application was found to be not novel to the respondent could have influenced the assessment of the novelty of the requirements.

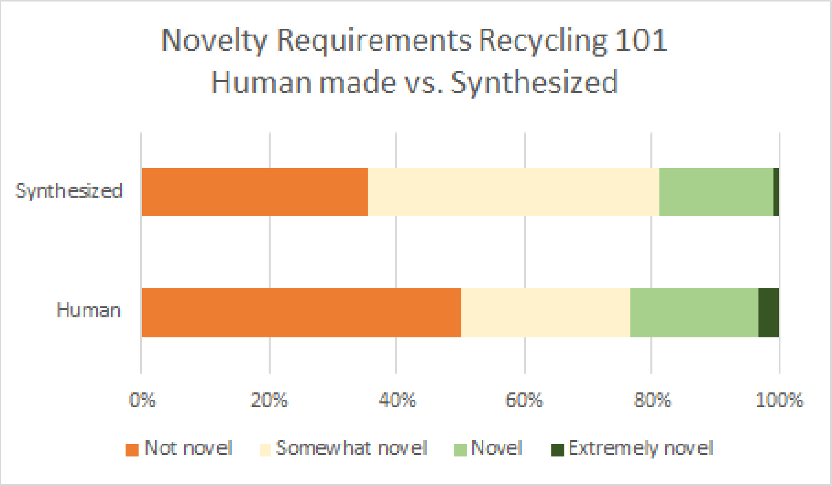


Figure 18: Assessment of novelty of synthesized and human made requirements for the Recycling 101 application

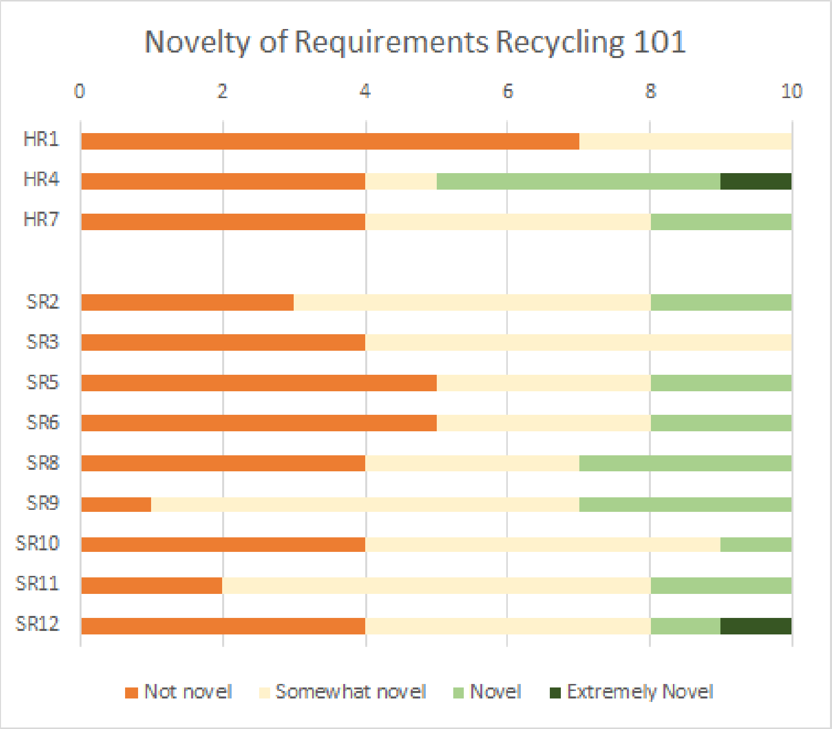


Figure 19: Assessment of novelty of Recycling 101 application requirements

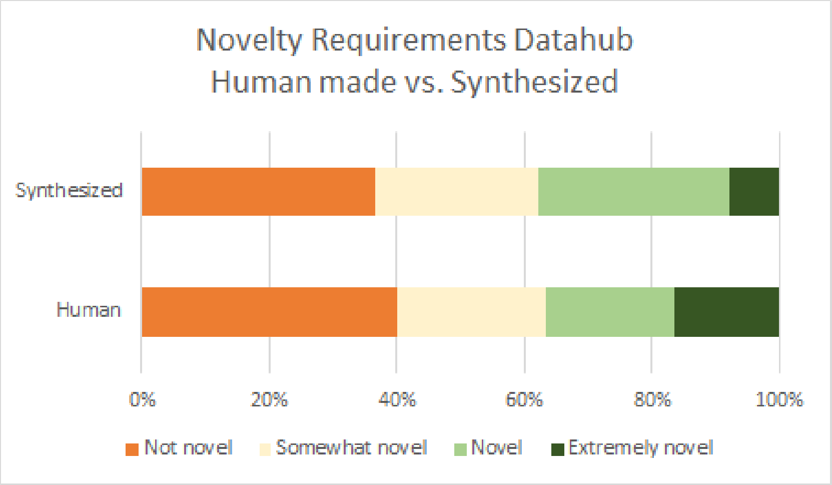


Figure 20: Assessment of novelty of synthesized and human made requirements for the Datahub application

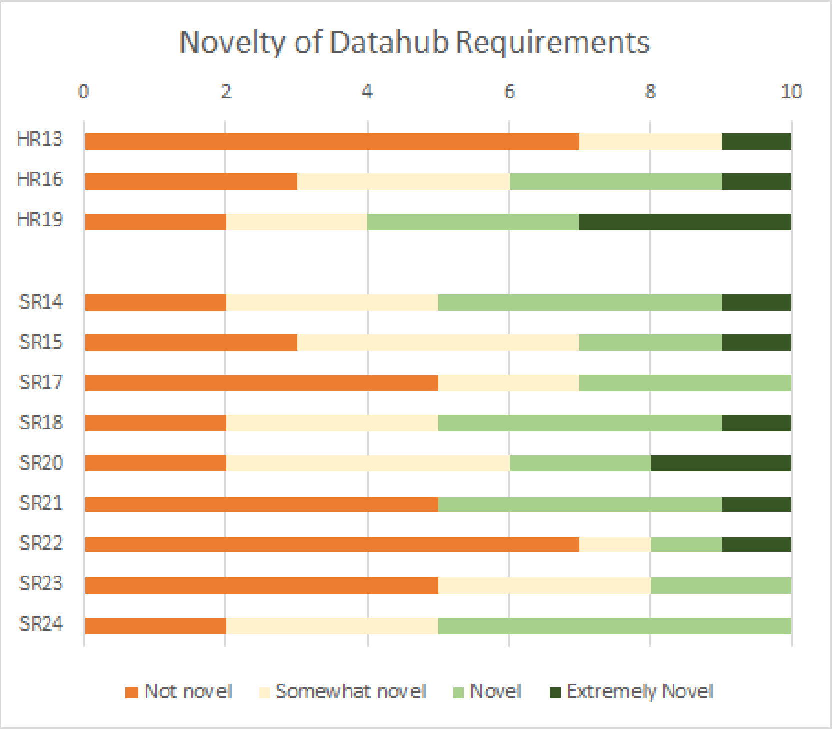


Figure 21: Assessment of novelty of Datahub application requirements

The surprisingness of the synthesized requirements of the Recycling 101 application are assessed a bit more surprising than the human made requirements for the same application, as shown in Figure 22. Especially when looking to the individual cases some synthesized requirements are assessed really surprising, such as SR6 an SR11 (Figure 23), which does not occur for the human made requirements. Another interesting requirement is SR5. The respondents answered conflicted about this requirement as some assessed it as not useful while others do think the requirement is useful.

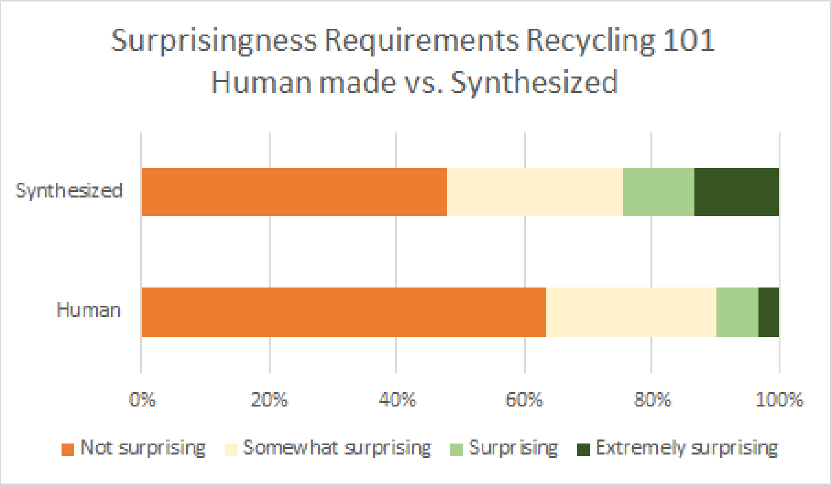


Figure 22: Assessment of surprisingness of synthesized and human made requirements for the Recycling 101 application

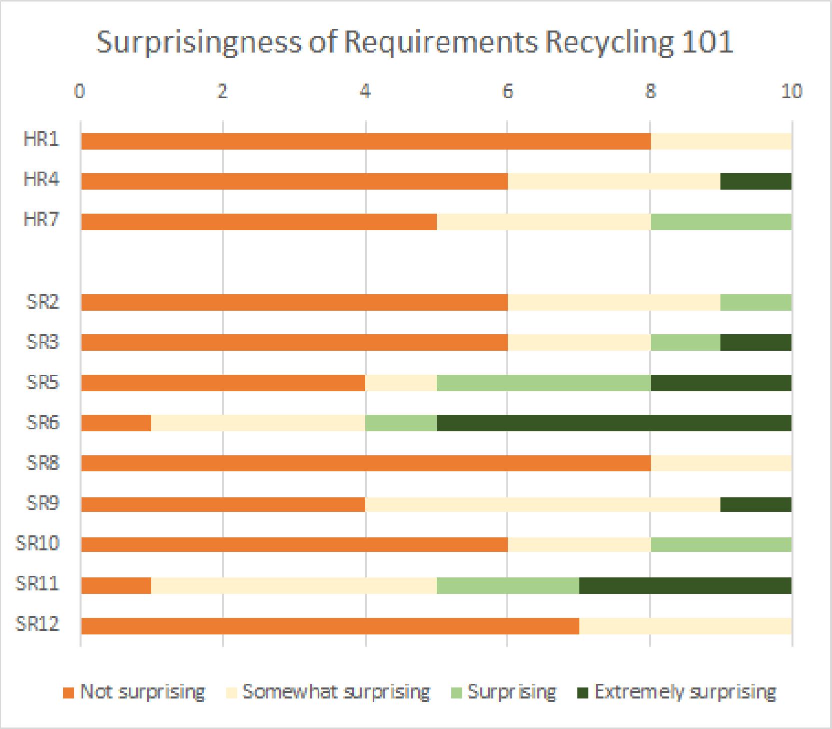


Figure 23: Assessment of surprisingness of Recycling 101 application requirements

The syntesized requirements of the Datahub application also look a bit more surprising than the human made requirements (Figure 24). The human made requirement HR13 is assessed least surprising by the respondents. Some of the respondents mention that an unexpected combination of user and goal of a requirement results in a surprising requirement as shown by the following quotes:

"I looked at how well the goal fits the user. If it is a good match, it's not surprising."

"Compared them to my expectancy of requirements for a specific role"

"I marked a requirement as surprising when it contained some weird or seemingly out of place concepts. For example: "As an Industry, I want to start an existing dataset instance". It would be surprising to me if an entire industry could own one account and start a dataset. It is surprising in the way that it does not make sense to me."

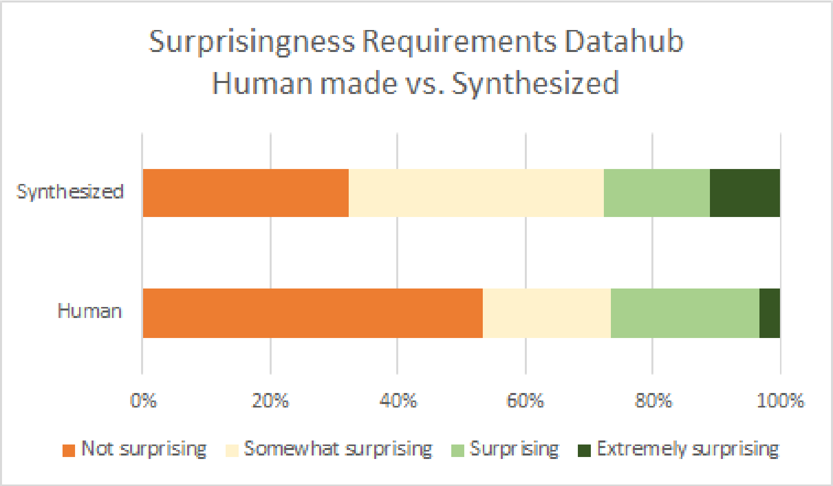


Figure 24: Assessment of surprisingness of synthesized and human made requirements for the Datahub application

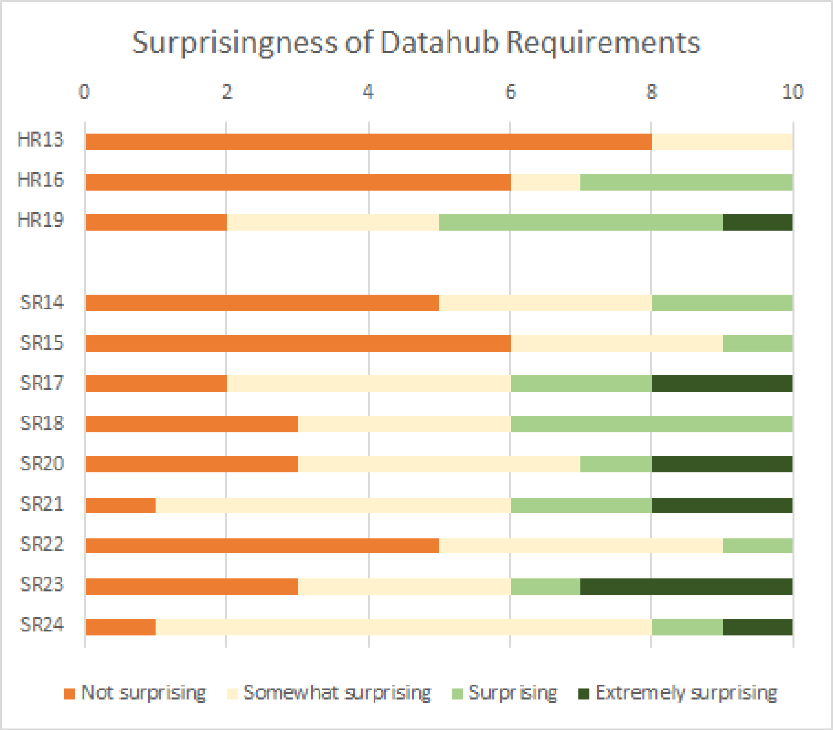


Figure 25: Assessment of surprisingness of Datahub application requirements

6 Discussion and Conclusion

In this section the different findings and results of this thesis are discussed and concluded. Main findings of this research are given and each research question is answered separately. Also the limitations and threats of this thesis are discussed and directions for future work are proposed.

6.1 Research Questions

The research questions introduced in the first section are as follows:

RQ1 How is creativity conceived in requirements engineering?
RQ2 How to extend a set of requirements with synthesized creativity?
RQ3 Are synthesized requirements, as made in RQ2, found to be creative as human made requirements?

The research questions are answered in the subsections below. After these answers the lesson learned for the main research question is discussed. The main research question was stated as follows: *MRQ. How to automatically synthesize creative requirements?*

Creativity in Requirements Engineering

The insights in Section 3.6 show creativity in the field of requirements engineering is conceived as a collaborative creative process. This process is influenced by multiple contexts, dimensions and factors. However this does not show us how to identify a creative requirement. To do this the creative product is discussed and looked into. Three dimensions are identified which are used to asses the creativity of a requirement: usefulness, novelty and surprisingness. These dimensions are shown in Table 8.

Synthesizing Creativity

In this thesis a set of requirements is extended with a program made in python. This program uses semantic role labeling to identify semantic parts that can be reused for new requirements. By reusing semantic parts from other requirements combinational creativity is used. Also semantic similarities to the semantic parts that are available are identified and used to create new requirements. This enhances the creativity of the program with exploratory creativity. With these two methods, new requirements are successfully synthesized based on input requirements from a domain. The method is applicable for any set of requirements if they are documented as user stories.

Assessed creativity

The synthesized requirements are assessed on three dimensions with a survey among information science students. With the survey six hypotheses are tested with a qualitative approach. The hypotheses that were as follows:

H1. The human generated requirements are more useful than the synthesized requirements.
H2. The synthesized requirements based on one application are more useful than the synthesized requirements based on multiple applications.
H3. The synthesized requirements are more novel than the human generated requirements.
H4. The synthesized requirements based on multiple applications are more novel than the synthesized requirements based on one application.

H5. The synthesized requirements are more surprising than the human generated requirements.

H6. The synthesized requirements based on multiple applications are more surprising than the synthesized requirements based on one application.

The conclusions stated in this section are preliminary results, as the questionnaires had only twenty respondents. Next to this, only a small sample of the human generated and synthesized requirements are surveyed to keep the focus of the respondents on the questionnaire. Based on the results the first hypothesis is accepted. Human generated requirements are found to be more useful than synthesized requirements. The difference between usefulness is clearly visible in the Figures 14 and 16. Also on the level of individual requirements most synthesized requirements are assessed less useful than the human made requirements. Only requirement SR2, SR8, SR12 and SR14 show that they are useful according to the respondents. Of these four requirements, three requirements are synthesized for the Recycling 101 application. However, on average there is not much difference found between synthesized requirements based on one application or based on multiple applications, which rejects H2. Next to the useful assessed requirements, some requirements are clearly assessed as not useful by the respondents. SR5, SR6, SR11 and SR17 are found to be not useful by most respondents. Also of these four requirements, three are synthesized requirements for the Recycling 101 application. This shows us that the usefulness of synthesized requirements based on one application has more extremes than the synthesized requirements based on multiple applications.

The novelty of synthesized requirements is not very different assessed from the novelty of human made requirements, which rejects the third hypothesis. H4 can be accepted as the novelty of the synthesized requirements based on multiple applications are assessed more novel than the synthesized requirements based on one application. However, this result is a bit flawed as all requirements of the Datahub application are assessed more novel than the requirements of the Recycling 101 application. This shows that the application or application domain could influence the results of the different assessments that are made.

In the surprisingness dimension little difference is shown between the different groups. The synthesized requirements are slightly more surprising assessed than the human made requirements, which results in a weak acceptance of the fifth hypothesis. Between the synthesized requirements based on one application and based on multiple applications also little difference is shown. The synthesized requirements of the datahub application are assessed higher on the somewhat more surprising mark than the synthesized requirements of the Recycling 101 application. This small difference is also a weak acceptance of the sixth hypothesis.

Lessons learned

The approach of extending a set of requirements used in this thesis showed that it is possible to extend a set of requirements. An approach of semantic role labeling is used to help the process of synthesizing requirements with the correctness of the requirements. With combining different parts of requirements combinational creativity is applied in an optimal way. The application of exploratory creativity could be improved as in this thesis only nouns are replaced to guarantee the semantic meaning of the requirements. Nevertheless, the output shows that synthesizing requirements with this approach is possible and shows promising results.

Assessing creativity of requirements however, is a challenge that will stay. Respondents repeatedly stated that they found it difficult to assess requirements on the novelty and surprisingness dimensions.

6.2 Limitations and Threats

This thesis had some limitations and some of these limitations represent threats to the validity of this thesis. These threats and limitations are discussed in this section.

Creating the program

During the creation of the program multiple challenges needed to be tackled. An approach with semantic role labeling was chosen and it was chosen to make the program in Python. The semantic role labeling is, due to setbacks with other libraries as discussed in section 4.1, performed with PractNLPTools. This tool has a F1-score of 75,49%, measured with the CoNLL 2005 benchmark. This score shows that the tool is not completely accurate, which could lead to a wrong identification of the numbered arguments. A library with a higher F1-score helps to increase the accuracy of the developed tool such as the SRL-tool of the Cognitive Computation Group (Punyakanok et al., 2008). Next to this, the library originally chosen to assign the thematic roles of the numbered arguments was not available for Python. Therefore a approach with SemLink was made to do this. However, this approach cannot take multiple meanings of one verb into account and can therefore be inaccurate. These two limitations can cause wrongly labeled thematic roles to parts of requirements.

Synthesizing the requirements with the created program also had a limitation. To make requirements that made sense only verb structures that started with an Agent were used. This limited the number of available verbs for creating requirements. An alternative way to build up requirements could help to resolve this.

The last limitation of this research during the creation of the program is related to the reasoning of requirements. The reason in user stories could not be used or synthesized due to the difficulty of relating a reasoning to an user story and learning the program how to use this. Therefore, it is chosen to not use any reasoning of user stories

Assessing creativity

Unfortunately, there was no method available to assess the creative product of a program. The framework proposed by Jordanous (2012), discussed in Section 3.4, evaluates a creative system instead of the creative product that is created with the system and could not be used in this thesis. Without a standard framework to assess creativity a survey was constructed to do this. Two of the three dimensions were hard to assess according to the respondents of the survey. Therefore, the results of these assessments could be flawed. Next to this, the likert scales used in the questionnaire to assess the dimensions could be interpreted in two ways. The likert scale was designed to have two negative answers, not [dimension] and somewhat [dimension], and two positive answers, [dimension] and extremely [dimension]. However, the respondents could also interpret the likert scale as one negative answer, not [dimension] and three positive answers, somewhat [dimension], [dimension], and extremely [dimension]. This leads to an unbalanced likert scale on which the respondents could assess the dimensions in another way than with a balanced likert scale.

6.3 Future Work

This thesis shows a lot of opportunities for future work. A first approach of synthesizing requirements with taking creativity into account is made and this could be further explored. A set of directions to do this will be given in this section.

Firstly, a standardized way of assessing a creative product could help in future work to asses requirements better on creativity. Ritchie (2001) already proposed formal rules to assess creativity of a product, but these do not seem to be actionable and applicable on requirements. Defining a right assessment method of the dimensions of creativity could help to define the scope of creative requirements.

As described in the previous section, the created program has some limitations on semantic role labeling and synthesizing requirements based on thematic roles. Research into other approaches of synthesizing requirements should clarify if the approach used in this thesis is the optimal approach. Next to this, the approach used in this thesis could be improved by using libraries with a higher accuracy of semantic role labeling. Also the exploratory creativity that is used in the created program could be extended more. Not only alternative semantic parts for nouns, but also for other kind of words could be found. Including requirements from other domains is another way to increase the exploratory creativity of the program. However, when other domains are included in the input of the program, a way to make sure the requirements are still relevant to the application domain should be developed.

The synthesized programs are clearly not ready to be automatically added the process, without human check. However, the requirements could help to increase the collaborative creative process of requirements engineering. Studying what output is needed to help this process is something that could be really interesting for future work.

References

Berry, D., Gacitua, R., Sawyer, P., & Tjong, S. F. (2012). The case for dumb requirements engineering tools. In B. Regnell & D. Damian (Eds.), *Requirements engineering: Foundation for software quality* (pp. 211–217). Berlin, Heidelberg: Springer Berlin Heidelberg.

Besemer, S. P. & O'Quin, K. (1987). Creative product analysis: Testing a model by developing a judging instrument. *Frontiers of creativity research: Beyond the basics*, 367–389.

Bhowmik, T., Niu, N., Savolainen, J., & Mahmoud, A. (2015). Leveraging topic modeling and part-of-speech tagging to support combinational creativity in requirements engineering. *Requir. Eng. 20*(3), 253–280. doi:10.1007/s00766-015-0226-2

biplab-iitb. (2016). Practnlptools. https://github.com/biplab-iitb/practNLPTools. GitHub.

Boden, M. (2009). Creativity in a nutshell. *5*, 83–96.

Bonial, C., Stowe, K., & Palmer, M. (2013). Renewing and revising semlink. In *Proceedings of the 2nd workshop on linked data in linguistics (ldl-2013): Representing and linking lexicons, terminologies and other language data* (pp. 9–17).

Carreras, X. & Màrquez, L. (2005). Semantic role labeling. Visited on 01 October 2018. TALP Research Center, Technical University of Catalonia (UPC). Retrieved from http://www.lsi.upc.edu/~srlconll/

Collobert, R. (n.d.). Senna. Visited on 01 October 2018. Ronan Collobert. Retrieved from https://ronan.collobert.com/senna/

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*(Aug), 2493–2537.

Colton, S. (2008). Creative intelligent systems, papers from the 2008 AAAI spring symposium, technical report ss-08-03, stanford, california, usa, march 26-28, 2008, AAAI. Retrieved from http://www.aaai.org/Library/Symposia/Spring/ss08-03.php

Cropley, D. H., Cropley, A. et al. (2005). Engineering creativity: A systems concept of functional creativity. *Creativity across domains: Faces of the muse*, 169–185.

Cybulski, J. L., Nguyen, L., Thanasankit, T., & Lichtenstein, S. (2003). Understanding problem solving in requirements engineering: Debating creativity with IS practitioners. In *The seventh pacific asia conference on information systems, PACIS 2003, adelaide, australia, july 10-13, 2003* (p. 32). Retrieved from http://aisel.aisnet.org/pacis2003/32

Daelemans, W., Buchholz, S., & Veenstra, J. (1999). Memory-based shallow parsing. *arXiv preprint cs/9906005.*

Dallman, S., Nguyen, L., Lamp, J. W., & Cybulski, J. L. (2005). Contextual factors which influence creativity in requirements engineering. In *Proceedings of the 13th european conference on information systems, information systems in a rapidly changing economy, ECIS 2005, regensburg, germany, may 26-28, 2005* (pp. 1734–1745). Retrieved from http://aisel.aisnet.org/ecis2005/107

Dalpiaz, F. (2018). Requirements data sets (user stories). Mendeley Data, v1. doi:10.17632/7zbk8zsd8y.1

Jivani, A. (2011). A comparative study of stemming algorithms. *Int. J. Comp. Tech. Appl. 2*, 1930–1938.

Jordanous, A. (2012). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, *4*(3), 246–279. doi:10.1007/s12559-012-9156-1

Jurafsky, D. & Martin, J. H. (2018). *Semantic role labeling* (3rd ed draft). Draft of September 23, 2018. Retrieved from https://web.stanford.edu/~jurafsky/slp3/

Kametani, T., Nishina, K., & Suzuki, K. (2010). Attractive quality and must-be quality from the viewpoint of environmental lifestyle in japan. In H.-J. Lenz, P.-T. Wilrich, & W. Schmid (Eds.), *Frontiers in statistical quality control 9* (pp. 315–327). Heidelberg: Physica-Verlag HD. doi:10.1007/978-3-7908-2380-6_20

Levin, B. (1993). *English verb classes and alternations.* Chicago: University of Chicago Press.

Lucassen, G., Dalpiaz, F., Werf, J. M. E. M. v. d., & Brinkkemper, S. (2016). The use and effectiveness of user stories in practice. In M. Daneva & O. Pastor (Eds.), *Requirements engineering: Foundation for software quality* (pp. 205–222). Cham: Springer International Publishing.

Mahaux, M., Mavin, A., & Heymans, P. (2012). Choose your creativity: Why and how creativity in requirements engineering means different things to different people. In *Requirements engineering: Foundation for software quality - 18th international working conference, REFSQ 2012, essen, germany, march 19-22, 2012. proceedings* (pp. 101–116). doi:10.1007/978-3-642-28714-5_9

Mahaux, M., Nguyen, L., Gotel, O., Mich, L., Mavin, A., & Schmid, K. (2013). Collaborative creativity in requirements engineering: Analysis and practical advice. In *IEEE 7th international conference on research challenges in information science, RCIS 2013, paris, france, may 29-31, 2013* (pp. 1–10). doi:10.1109/RCIS.2013.6577678

Maiden, N. A. M., Jones, S., Karlsen, I. K., Neill, R., Zachos, K., & Milne, A. (2010). Requirements engineering as creative problem solving: A research agenda for idea finding. In *RE 2010, 18th IEEE international requirements engineering conference, sydney, new south wales, australia, september 27 - october 1, 2010* (pp. 57–66). doi:10.1109/RE.2010.16

Manning, C. D. (2011). Part-of-speech tagging from 97% to 100%: Is it time for some linguistics? In A. F. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (pp. 171–189). Berlin, Heidelberg: Springer Berlin Heidelberg.

Manurung, R., Ritchie, G., Pain, H., Waller, A., O'Mara, D., & Black, R. (2008). The construction of a pun generator for language skills development. *Applied Artificial Intelligence*, *22*(9), 841–869. doi:10.1080/08839510802295962

Màrquez, L., Carreras, X., Litkowski, K. C., & Stevenson, S. (2008). Semantic role labeling: An introduction to the special issue. *Computational Linguistics*, *34*(2), 145–159. doi:10.1162/coli.2008.34.2.145

McCorduck, P. & Cohen, H. (1991). *Aarons code: Meta-art, artificial intelligence, and the work of harold cohen.* W.H. Freeman.

Mohit, B. (2014). Named entity recognition. In *Natural language processing of semitic languages* (pp. 221–245). Springer.

Mustasfa, B. A. (2014). Classifying software requirements using kano's model to optimize customer satisfaction. In *New trends in software methodologies, tools and techniques - proceedings of the thirteenth somet '14, langkawi, malaysia, september 22-24, 2014* (pp. 271–279). doi:10.3233/978-1-61499-434-3-271

Nguyen, L. & Shanks, G. G. (2009). A framework for understanding creativity in requirements engineering. *Information & Software Technology*, *51*(3), 655–662. doi:10.1016/j.infsof.2008.09.002

Nuseibeh, B. & Easterbrook, S. M. (2000). Requirements engineering: A roadmap. In *22nd international conference on on software engineering, future of software engineering track, ICSE 2000, limerick ireland, june 4-11, 2000.* (pp. 35–46). doi:10.1145/336512.336523

Pohl, K. (1994). The three dimensions of requirements engineering: A framework and its applications. *Inf. Syst. 19*(3), 243–258. doi:10.1016/0306-4379(94)90044-2

Pohl, K. (1999). Requirements engineering : An overview.

Punyakanok, V., Roth, D., & Yih, W.-t. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, *34*(2). Retrieved from http://cogcomp.org/papers/PunyakanokRoYi07.pdf

Reiter, E. & Dale, R. (2000). *Building natural language generation systems*. Cambridge university press.

Ritchie, G. (2001). Assessing creativity. In *Proc. of aisb'01 symposium*. Citeseer.

Ritchie, G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds and Machines*, *17*(1), 67–99. doi:10.1007/s11023-007-9066-2

Sawyer, R. K. (2006). *Explaining creativity: The science of human innovation* (1st ed.). Oxford University Press.

Trask, A., Michalak, P., & Liu, J. (2015). Sense2vec - A fast and accurate method for word sense disambiguation in neural word embeddings. *CoRR*, *abs/1511.06388*. arXiv: 1511.06388. Retrieved from http://arxiv.org/abs/1511.06388

van Lamsweerde, A. (2009). *Requirements engineering - from system goals to UML models to software specifications*. Wiley. Retrieved from http://eu.wiley.com/WileyCDA/WileyTitle/productCd-EHEP000863.html

A Requirements Input

A.1 Recycling 101

1 As a user, I want to click on the address, so that it takes me to a new tab with Google Maps.

2 As a user, I want to be able to anonymously view public information, so that I know about recycling centers near me before creating an account.

3 As a user, I want to be able to enter my zip code and get a list of nearby recycling facilities, so that I can determine which ones I should consider.

4 As a user, I want to be able to get the hours of each recycling facility, so that I can arrange drop-offs on my off days or during after-work hours.

5 As a user, I want to have a flexible pick up time, so that I can more conveniently use the website.

6 As a user, I want to be able to select different types of recyclable waste, so I have and get a list of facilities that accept each type and their opening hours, so that I can find an optimal route and schedule.

7 As a user, I want to add donation centers as favorites on my profile, so that I can view them later.

8 As a user, I want to be able to give my email ID, so that I can receive notifications for new events as they are posted.

9 As a user, I want to be able to view a map display of the public recycling bins around my area.

10 As a user, I want to be able to view a map display of the special waste drop off sites around my area.

11 As a user, I want to be able to view the safe disposal events currently being organised around my area.

12 As a user, I want to choose a flexible pick up time, so that I can more conveniently use the website.

13 As a user, I want to view user documentation for the website, so that I know how to use the web app.

14 As a user, I want to get feedback when I enter an invalid zip code.

15 As a user, I want to be able to create an acocunt, so that I can create my own profile.

16 As an admin, I want to be able to add or remove recycling facilities' information, so that users get the most recent information.

17 As an admin, I want to be able to read users' feedback and complaints, so that we can add more features and keep improving the service we provide to them.

18 As a user, I want to be able to check transaction history and keep a record of it, so that I can go back when needed.

19 As a user, I want to have a great UI and UX from the sites, so that I have a pleasant experience when navigating through them.

20 As a user, I want to be able to access the site and do all the other stuffs on all of my electronic devices.

21 As an admin, I want to be able to block specific users based on IP address, so that I can prevent spamming on the websites.

22 As an admin, I want to view a dashboard that monitors all the sites' statuses, so that I can have a sense of what people are doing on our sites and know the service status.

23 As an admin, I want to have all data encrypted, so that important information will not be stolen during a server breach or an attack.

24 As an executive, I want to have full access to data related to my company, so that I can have a sense of my company's performance.

25 As an employee, I want to access the route planning system during work, so that I can be guided through the neighbourhood.

26 As an employee from the HR department, I want to have access to the full information of all employees working for this business.

27 As a developer, I want to access an API from the website, so that I can integrate it and implement certain features in my own iOS application.

28 As a user, I want to be able to receive tempting rewards, so that I have a reason to use the website.

29 As a user, I want to have my personal information kept securely in the database of the website, so that I will not suffer from identity theft or telephone harassment.

30 As an admin, I want to handle all users' activities, so that I can manage more efficiently.

31 As a company, I want to have a website that is easy to use, so that I can upload or delete stuff step by step.

32 As an employee, I want to get quick notifications, so that I can process cases the first time.

33 As a company accountant, I want to view all available activity fees online, so that I can easily create a bill statement.

34 As a developer, I want to use bootstrap in the process of developing, so that I can easily design my website.

35 As a developer, I want to attend some UI/UX lessons, so that I can develop an awesome and beautiful features website.

36 As a user, I want to view all locations of recycling centers on a map, so that I can check which routes to take to drop off waste.

37 As a user, I want to upload my week's schedule, so that I can get recommendations for recycling centers that best fit my availability.

38 As a user, I want to link my email account to my profile, so that I can get a temporary password in case I forget my own one.

39 As a user, I want to contact the administrators, so that I can give feedback or ask for help.

40 As an admin, I want to add recycling center information, so that I can keep the database up-to-date over time.

41 As an admin, I want to view user error logs, so that I can fix or review any issues that are being faced by users of the

system.

42 As an admin, I want to onboard recycling centers on the platform, so that I can increase information accuracy.

43 As a superuser, I want to update the recycling center information, so that I can provide the latest information about the recycling center.

44 As a superuser, I want to view users' stats, so that I can view in real-time how many users have visited my recycling center information and their recyclable waste.

45 As a superuser, I want to reply to user questions, so that I can answer any questions about my recycling center.

46 As an admin, I want to be able to have a dashboard that shows usage stats and locations, so that I can identify the neighbourhoods with the largest number of drop-offs and to try getting more facilities involved.

47 As an admin, I want to be able to communicate directly with facilities, so that I can keep them updated about features we have on our website.

48 As a user, I want to be able to browse through the list of facolities and see which ones are environment-friendly, so that I can know for sure my waste is not going to leave a negative ecological footprint.

49 As a recyclingfacility representative, I want to be able to update my information and the type of material I accept, so that I can avoid any miscommunication with users.

50 As a recyclingfacility representative, I want to have access to user stats and schedules, so that I can adjust my hours and/or upgrade equipment and capacity in order to be able to accomodate larger amounts of recyclable materials.

51 As a recyclingfacility, I want to be able to communicate directly with the site admin and convey any issues or concerns I have, so that they fix them.

A.2 Datahub

1 As a Publisher, I want to publish a dataset, so that I can view just the dataset with a few people.

2 As a Publisher, I want to publish a dataset, so that I can share the dataset publicly with everyone.

3 As a Publisher, I want to sign up for an account, so that that I can publish my data package to the registry and to have a publisher account to publish my data package under.

4 As a Visitor, I want to sign up via github or google, so that that I don't have to enter lots of information and remember my password for yet another website.

5 As a Publisher, I want to know what do next after signing up, so that that I can get going quickly.

6 As an Admin, I want to invite someone to join the platform, so that that they can start contributing or using data.

7 As a Publisher, I want to import my data package into the registry, so that my data has a permanent online home to access.

8 As a Publisher, I want to configure my client, so that I can

```
     start publishing data packages.
 9  As a Publisher , I want to use a publish command to update a
     data package that is already in the registry , so that it
     appears there.
10  As a Publisher , I want to unpublish a data package, so that it
     is no longer visible to anyone.
11  As a Publisher , I want to permanently delete a data package, so
      that that it no longer takes up storage space.
12  As a Publisher , I want to validate the data I am about to
     publish to the registry , so that that I publish good data
     and know that I am doing.
13  As a Publisher , I want to data to be validated when I publish
     it , so that that I know immediately if I have accidentally
     broken my data or have bugs and can take action to correct.
14  As a Consumer, I want to know that the data I am downloading is
      good and can be relied on, so that that I don't have to
     check it myself or run into annoying bugs later on.
15  As a Publisher , I want to publish a data package where its
     resource data is stored on my servers but the registry
     caches a copy of that data, so that that if my data is lost
     or gets broken I still have a copy people can use.
16  As a Consumer, I want to be able to get the data for a data
     package even if the original data has been moved or removed,
      so that that I can still use is and my app or analysis
     keeps working.
17  As a Publisher , I want to publish a data package in the UI, so
     that that it is available and published.
18  As a Publisher , I want to create a data package in the UI so
     that it is available and published.
19  As a Publisher , I want to be able to restore the deleted data
     package via cli , so that that it is back visible and
     available to view, download.
20  As a Publisher , I want to undelete the deleted data packages,
     so that that the deleted data packages is now visible again.
21  As a Publisher , I want to be able to preview the views of the
     current data package using cli prior to publishing, so that
     that I can refine the json declarations of datapackage view
     section to achieve a great looking result.
22  As a Consumer, I want to view a data package online, so that I
     can get a sense of whether this is the dataset I want.
23  As a publisher , I want to show the world how my published data
     is , so that that it immediately catches consumer's attention
     .
24  As a consumer, I want to view the data package, so that that I
     can get a sense of whether I want this dataset or not.
25  As a Publisher , I want to preview a datapackage I have prepared
     , so that that I can check it works and share the results.
26  As a Consumer, I want to see how much the data has been
     downloaded, so that that I can choose most popular in the
     case when there are several alternatives for my use case.
27  As a Publisher , I want to see real examples of published
     packages, so that I can understand how useful and simple the
```

datapackage format and the registry is.

28 As a Consumer, I want to see some example data packages quickly , so that I get a sense of what is on this site and if it is useful to look further.

29 As a Consumer, I want to search data packages, so that that I can find the ones I want.

30 As a Consumer, I want to search based on description of data package, so that that I can find package which related to some key words.

31 As a Consumer, I want to download the data package in one file , so that that I don't have to download descriptor and each resource by hand.

32 As a Developer, I want to use data package as a node lib in my project, so that that I can depend on it using my normal dependency framework.

33 As a Consumer, I want to load a Data Package from R, so that that I can immediately start playing with it.

34 As a Data Analyst, I want to download a data package, so that that I can study it and wrangle with it to infer new data or generate new insights.

35 As a Data Analyst, I want to update previously downloaded data package, so that that I can work with the most recent data.

36 As a Consumer, I want to download a DataPackage's data one coherent SQLite database, so that that I can get it easily in one form.

37 As a Data Analyst, I want to compare different versions of some datapackage locally, so that that I can see schema changes clearly and adjust my analytics code to the desired schema version.

38 As a Web Developer, I want to be able to install multiple versions of the same datapackage separately, so that that all my projects could be developed independently and deployed locally.

39 As a Developer, I want to list all DataPackages requirements for my project in the file and pin the exact versions of any DataPackage that my project depends on, so that that the project can be deterministically deployed locally and won't break because of the DataPackage schema changes.

40 As a Publisher, I want to tag datapackage to create a snapshot of data on the registry server, so that that consumers can refer to it.

41 As a Publisher, I want to be warned that a tag exists when I try to overwrite it, so that that I don't accidentally overwrite stable tagged data which is relied on by consumers .

42 As a Publisher, I want to be able to overwrite the previously tagged datapackage, so that that I can fix it if I mess up.

43 As a Publisher, I want to version my Data Package and keep multiple versions around including older versions, so that that I do not break consumer systems when I change my datapackage.

44 As a Publisher, I want to be able to get access to a previous

version I tagged, so that that I can return to it and review it.

45 As a Consumer, I want to know full details of how the data package schema has changed, so that that I can adjust my scripts to handle it.

46 As a Consumer, I want to get a sense of the age of the datapackage that I have downloaded before, so that that I can decide if I should update or not.

47 As a Consumer, I want to view a Datapackage at a particular version online, so that that I can present/discuss the particular data timeslice of interest with other people.

48 As a Consumer, I want to download a Data package at a particular version, so that that I know it is compatible with my scripts and system.

49 As a Consumer, I want to be notified of changes to a package I care about, so that that I can check out what has changed and take action.

50 As a Consumer, I want to see how active the site is, so that I can decide if I should get involved.

51 As a Consumer, I want to browse and find publishers, so that that I can find interesting publishers and their packages.

52 As a Consumer, I want to see a publisher's profile, so that that I can discover their packages and get a sense of how active and good they are.

53 As a Consumer, I want to view a publisher's profile, so that that I can see who is behind a particular package or to see what other packages they produce.

54 As a Consumer, I want to search among all data packages owned by a publisher, so that that I can easily find one data package amongst all the data packages by this publisher.

55 As an Owner, I want to edit my profile, so that that it is updated with new information.

56 As an Owner, I want to invite an existing user, so that the user can become a member of my publisher.

57 As an owner, I want to invite someone using their email to sign up and become a member of my Publisher, so that that they are authorized to publish data packages under my Publisher.

58 As an owner, I want to remove someone from membership in my publisher, so that they no longer have ability to publish or modify my data packages.

59 As an owner, I want to view all the people in my organization and what roles they have, so that that I can change these if I want.

60 As an owner, I want to make a user an owner, so that they have full control.

61 As an owner, I want to remove a user as an owner, so that they are just a member and no longer have full control.

62 As an Admin, I want to set key configuration parameters for my site deployment, so that that I can change key information like the site title.

63 As an Admin, I want to see key metrics about usage such as users, API usage, downloads, so that that I know how things

are going .

64 As an Admin, I want to have a pricing plan and billing system , so that that I can charge users and make my platform sustainable .

65 As a Publisher , I want to know if this site has a pricing plan and what the prices are , so that that I can work out what this will cost me in the future and have a sense that these guys are sustainable .

66 As a Publisher , I want to sign up for a given pricing plan , so that that I am entitled to what it allows .

67 As a Publisher , I want to have private data packages , so that I can share just with my team.

68 As a Data Publishing User , I want to be able to edit a dataset I have published , so that I can correct or enhance existing data .

69 As a Data Publishing User , I want to be able to edit the model of data I have already imported , so that I can fix bugs or make enhancements in the API built for my data .

70 As a Data Publishing User , I want to be able to delete a dataset I have published , so that I can remove unwanted data from OpenSpending .

71 As a Platform Administrator , I want to be able to Hide any dataset already added as Public , so that I can maintain Public/Hidden status for other users .

72 As a Platform Administrator , I want to have a view on all datasets published by all users , so that I can perform management actions on any dataset .

73 As a Platform Administrator , I want to be able to delete any dataset published , so that I can deal with takedown requests , or clean up test datasets .

74 As a Data Publishing User , I want to be able to edit the data source of data I have already imported , so that I can fix bugs or make enhancements in the API built for my data .

75 As a Data Publishing User , I want to have the Packager support Constants , so that I can model dimensions that may not exist in the source file .

76 As a Data Publishing User , I want to be able to import data in Excel , so that I do not have to convert data formats in order to use the data packager .

77 As a Data Publishing User , I want to know what my data needs to be able to be visualised on a map, so that I can visualise it on a map.

78 As a Data Publishing User , I want to be able to import data in JSON, so that I do not have to convert data formats in order to use the data packager .

79 As a Data Publishing User , I want to be able to import data from a Google Spreadsheet , so that I do not have to convert data formats in order to use the data packager .

80 As a Data Publishing User , I want to be able to import data from Fiscal Data Package descriptor file , so that I do not have to convert data formats in order to use the data packager .

81 As a Data Publishing User, I want to be able to provide the Platform Administrator with additional GeoJSON sources, so that I can improve the map-based visualisations of my data.

82 As a Data Consuming User, I want to be able to filter, sort and aggregate data by multiple dimensions and measures, so that I can get more granular views on the data.

83 As a Data Consuming User, I want to be able to download a CSV of the data that is used in any visualisation I am viewing, so that I can use the data in other tools.

84 As a Data Consuming User, I want to be able to change the display of all monetary measures across a set of currencies, so that I can understand localised amounts in non-localised figures.

85 As a Data Consuming User, I want to see textual descriptions that accompany embedded visualisations, so that I can more easily understand what I am viewing.

86 As a Data Consuming User, I want to be able to share a view state as a URL to social networks, so that I can share data that I have found with others.

87 As a Data Consuming User, I want to be able to download an image ofa particular view state, so that I can use it offline.

88 As a Data Consuming User, I want to be able to share an image of a particular view state to the social networks that support this, so that I can provide richer context in those communication channels for data I am sharing.

89 As a Data Consuming User, I want to be able to have stepped zoom on map visualisations, so that I can have better control over the navigation experience inside a map view.

90 As a Data Consuming User, I want to have consistent use of colour on map visualisations, so that I can better understand the visual logic of the map view.

91 As a Developer, I want to be able to customise the Brand Name and Icon, and Primary Color of all frontend Javascript apps, so that I can customise the branding for my own needs.

92 As an API User, I want to be able to understand if a user is a Publisher, so that I can offer functionality based on Dataset Publisher privileges.

93 As an API User, I want to be able to understand if a user is an Administrator, so that I can offer functionality based on Platform Administration privileges.

94 As an API User, I want to be able to get bordering regions| cities when I query a region|city, so that I can provider wider visual context for mapping visualisations.

95 As an API User, I want to be able to dynamically request polygons based on the query made, so that I can provide maps that match the query.

96 As an API User, I want to have a flexible API using HASC codes for countries, regions and cities, so that I can visualise budget data on maps.

97 As an API User, I want to be able to get a CSV output of any cube-based query, so that I can use work with tools that

read CSV.

98 As an API User, I want to be able to get a set of monetary measures transferred to different currencies, so that I can use this in scenarios that might enable comparison by normalisation.

99 As an API User, I want to be able to use metadata to get results from multiple datasets, so that I can build user experiences based on more than one dataset more easily.

100 As an API User, I want to be able to use data to get results from multiple datasets, so that I can build user experiences based on more than one dataset more easily.

101 As an API User, I want to be able to normalise measures by population, so that I work with datasets in reference to their contextual constraints.

102 As an API User, I want to be able to normalise measures by geographical area, so that I work with datasets in reference to their contextual constraints.

103 As an API User, I want to be able to normalise measures by GDP, so I work with datasets in reference to their contextual constraints.

104 As an API User, I want to be able to normalise measures by GINI and related socioeconomic indexes, so that I work with datasets in reference to their contextual constraints.

105 As an API User, I want to be able to get a relative percentage of a measure to the total of the dataset it comes from, so that I can build alternative displays of the data.

106 As an API User, I want to be able to persistently store visualisation state in the database, so that such can be shared more easily and contribute to a visualisation gallery.

107 As a Data Publishing User, I want to have my dataset update automatically as the source file/files changes, so that OpenSpending always shows current data.

108 As an OpenSpending Community Member, I want to have a blog that highlights any and all projects in the open fiscal space, so that I can relate to openspending.org as the central hub of fiscal openness.

109 As a User, I want to be able to set my own username, so that my data is more easily discoverable.

110 As a Data Publishing User, I want to be able to add a dataset in a Hidden state, so that I can work on a dataset before having it discoverable via OpenSpending user interfaces.

111 As a Data Publishing User, I want to be able to Hide a dataset that I have already added as Public, so that I can fix my mistakes or have a dataset primarily for my own use.

112 As a Data Publishing User, I want to have a view on all the datasets I have published, so that I can perform management actions on my own datasets.

113 As a Data Publishing User, I want to have a functioning Python Client, so that I can add data to the datastore in bulk from the command line or my own programs.

114 As an OpenSpending Community Member, I want to have an app

where I can find examples of use of fiscal data visualisations, so that I can find guidance in creating my own with Open Spending.

115 As a Data Publishing User, I want to know if my CSV file is valid, so that I can fix possible data issues before publishing it on Open Spending.

116 As a Data Consuming User, I want to be able to search any dataset published and publicly accessible by their title and metadata, so that I can find the datasets I'm interested in .

117 As a Data Consuming User, I want to visualize by default in treemap, bubble tree, map and pivot table the most recent year when my dataset contain multiple years, so that I'm not confused with the amounts.

118 As an API user, I want to be able to change the colors of the embedded visualisations in my own platform, so that I can customize the visualisations.

119 As an API user, I want to be able to change some of the styling of the embedded Viewer in my own platform, so that I can brand it to my own organisation's color scheme.

120 As a Platform administrator, I want to be able to translate the data types hierarchies of the Viewer while in embed mode, so that my users can understand the interface in their native language.

121 As a Developer, I want to get a Data Package into Node, so that I can start using the data for doing analysis and visualizations.

122 As a Researcher, I want to get a Data Package into Julia in seconds, so that I can start using the data for doing analysis and visualizations.

123 As a Publisher, I want to add type information to my data, so that it is more useful to others and can be used better with tools like visualization programs.

124 As a Publisher, I want to be able to provide a visualization of data in the Data Package, so that I can provide my analysis and show my work to users of the data.

125 As a Researcher, I want to be able to save new visualizations, so that I can share them with others or include them in the Data Package.

126 As a ResearcherPublisher, I want to know that my data conforms to its Data Package profile, so that I can feel trust in the validity and usefulness of the data.

127 As a ResearcherPublisher, I want to understand the ways in which my data is invalid, so that I can know how to fix it.

128 As a Researcher, I want to get a Data Package into R in seconds , so that I can start using the data for doing analysis and visualizations.

129 As a Researcher, I want to get a Data Package into Excel in seconds, so that I can start using the data for doing analysis and visualizations.

130 As a Researcher, I want to get a Data Package into SPSS in seconds, so that I can start using the data for doing

analysis and visualizations.

131 As a Researcher, I want to get a Data Package into STATA in seconds, so that I can start using the data for doing analysis and visualizations.

132 As a Researcher, I want to be able to translate my EML dataset to a Data Package, so that I can benefit from the wide array of tools available for Data Packages.

133 As a Researcher, I want to get a Data Package into LibreOffice/OpenOffice in seconds, so that I can start using the data for doing analysis and visualizations.

134 As a Developer, I want to get a Data Package into Python in seconds, so that I can start using the data for doing analysis and visualizations.

135 As a Developer, I want a jQuery plugin for Core Data Packages, so that I can use it to apply to form control that uses a core dataset for autocompletion.

136 As a Researcher, I want to get my Excel spreadsheet into a Data Package, so that I can benefit from better tooling and standardization.

137 As a Developer, I want to do exploratory data analysis in R and operationalize that analysis in Python, so that I can use the best tool for the job.

138 As a Developer, I want to get a Data Package into Clojure in seconds, so that I can start using the data in doing analysis and visualizations.

139 As a Developer, I want to get a Data Package into Julia in seconds, so that I can start using the data in doing analysis and visualizations.

140 As a Developer, I want to get a Data Package into C++ in seconds, so that I can start using the data in doing analysis and visualizations.

141 As a Machine Learning expert, I want to package ML datasets as data packages, so that I can easily import them into my ML platform, so that I can start using the data in doing analysis.

142 As a Developer, I want an Elasticsearch integration, so that I can integrate data-packaged data with pipelines that use Elasticsearch.

143 As a Developer, I want an SPSS integration, so that I can integrate data-packaged data with pipelines that use SPSS.

144 As a Developer, I want an EPrints integration, so that I can integrate data-packaged data with pipelines that use EPrints.

145 As a Developer, I want a Mongo integration, so that I can integrate data-packaged data with pipelines that use Mongo.

146 As a Developer, I want a DAT integration, so that I can integrate data-packaged data with pipelines that use DAT.

147 As a ResearcherGovernment Publisher, I want to add general reference data to my narrow dataset, so that my dataset is more useful.

148 As a ResearcherGovernment Publisher, I want to add general country names to my dataset that only contains country codes

, so that my dataset is more readable.

149 As a ResearcherGovernment Publisher, I want to add reference data on inflation to my spending dataset, so that the spending lines in my dataset is more understandable.

150 As a ResearcherGovernment Publisher, I want to map lines in my dataset using geographic data in my dataset, so that my dataset is more engaging for non-technical users.

151 As a Researcher, I want to be able to reference a remote-controlled vocabulary for my dataset, so that I can be sure that column of my dataset are valid against a single shard list of terms.

152 As a developer, I want an DSpace integration, so that I can integrate data-packaged data with pipelines that use Dspace.

153 As a Developer, I want Feather integration, so that I can integrate data-packaged data with pipelines that use Feather .

154 As a Developer, I want HDF5 integration, so that I can integrate data-packaged data with pipelines that use HDF5.

155 As a Researcher, working with data, I want an Microsoft Power BI integration, so that I can import datasets without downloading them locally.

156 As a ResearcherPublisher, I want an integration with Zenodo, so that when I store my dataset in GitHub, I don't have to retype descriptive information about my dataset.

157 As a Publisher, I want an integration with Open Refine, so that I can output cleaned Data Packages.

158 As a ResearcherPublisher, I want to publish Data Packages to CKAN, so that my data is findable, and I can have a data API .

159 As a ResearcherDeveloper, I want the ability import/export from MS-SQL, so that I can use Data Packages in workflows that involve MS-SQL.

160 As a Researcher, working with data in NetCDF, I want NetCDF integration, so that I can store my data in plaintext while still retaining its metadata.

161 As a Researcher, I want an integration with https://data.mendeley.com/, so that I can validate my data upon ingest to the service.

162 As a Publisher, I want an integration with Excel, so that I can output cleaned Data Packages.

163 As a Publisher, I want to store my data quickly and easily online.

164 As a Repository Manager, I want a tool that makes it easy for researchers/ users to add basic metadata to their research data, so that it is more findable and therefore useful.

165 As a ResearcherPublisher, I want validate my data with a minimum of clicks, so that I can feel trust in the validity and usefulness of the data.

166 As a publisher, I want to be able to check that every time I update my data it is still good, so that I can catch errors early and store reliable data.

167 As a DeveloperWrangler, I want to use a command line tool that

allows met to validate data , so that I can feel trust in the validity and usefulness of the data quickly and in the context of my command line workflow .

168 As a developer , I want an online service that is connected to my data repository that validates data on update , so that I can delegate data validation to a third party .

169 As a government Publisher , I want to make it easy to prove that our published data is valid , so that I can claim that we are living up to our transparency commitments .

170 As a Civic Tech Activist , I want to make it easy to assess the quality of data stored by the government , so that I can make sure that government is living up to its transparency commitments .

171 As a publisher , I want to embed an interactive preview of my data on my site , so that users can be encouraged that this is the correct data for them .

172 As a publisher , I want to embed a preview button on my site , so that users can preview the data and be encouraged that this is the correct data for them .

173 As a Publisher , I want to know how many users have previewed a dataset , so that I know how interest in a dataset relates to its actual download numbers .

174 As a Developer , I want to customize an existing wizard for my specific type of data , so that I can give my users a great user experience .

175 As a Publisher , I want to add useful metadata or add in new data columns to make the dataset more useful .

176 As a publisher , I want to package reproducible steps to get a certain data state , so my methodology is transparent and can be rerun by others .

177 As a DeveloperDataWrangler , I want to store my Data Package in GitHub and have it automatically stored into CKAN, so that I get a data API and my dataset is listed in a relevant catalog .

178 As a Researcher , I want a tool that can generate basic statistics about a dataset , so that I can get a quick preview of the data .

179 As a DeveloperPublisher , I want a tool to create an embeddable data summary via iframe , so that I can embed data summaries across sites .

180 As a Researcher , I want an app that generates an OpenRefine reconciliation API endpoint from a Tabular Data Package , so that I can use it to clean messy data .

181 As a Researcher , I want an app that create proxy Data Packages for well know and reliable data , sources , so that I can load high quality data using Data Package tooling .

182 As a RepositoryManagerResearcher , I want an app that acts as a match−making service for packaging data , so that owners are paired with data packagers .

183 As a developer , I want to create a web socket protocol for Frictionless data tools , so that I can easily use data packages with additional data analysis tools .

184 As a Publisher, I want a tool to check data availability persistence after publication.

185 As a ResearcherPublisher, I want to specify the funding that contributed to the creation of a given dataset, so that funding agencies can identify the funding, source for a given dataset.

186 As a ResearcherPublisher, I want to add a DOI to a dataset, so that I can cite it in papers stored with the data.

187 As an app developer, I want to include the code of a dataset type in my app artifact and create a dataset of that type when deploying the app.

188 As an app developer, I want to deploy a new version of a dataset type as part of deploying a new version of the app that includes it and I expect that all dataset instances of that type that were created as part of the app deployment start using the new code.

189 As an app developer, I want to deploy a new version of a dataset type as part of an app artifact, without affecting other datasets of this type.

190 As an app developer, I want to explore a dataset instance of a type that was deployed as part of an app.

191 As an app developer, I want to ensure that when I deploy an artifact without creating an app this will not create any dataset types or instances.

192 As an app developer, I want to share a dataset type across multiple applications that include the dataset type's code in their artifacts.

193 As an app developer, I want to ensure that when I deploy a new version of an app that includes a shared dataset type that all dataset instances created by this app start using the new code but all dataset instances created by other apps remain unchanged.

194 As an app developer, I want to ensure that when I deploy a new version of an app that includes an older version of a dataset type deployed by another app and I expect that the dataset instances created by this app use the dataset type code included in this app.

195 As an app developer, I want to ensure that when I deploy a new version of an app that includes a different version of a dataset type deployed by another app and this app shares a dataset instance of this type with the other app the deployment will fail with a version conflict error.

196 As an app developer, I want to share a dataset type that I had previously deployed as part of an app.

197 As a dataset developer, I want to deploy a dataset type independent from any app and allow apps to create and use dataset instances of that type.

198 As a dataset developer, I want to have the option of forcing applications to have the dataset code injected at runtime.

199 As a dataset developer, I want to have an archetype that helps me package my dataset type properly.

200 As a dataset developer, I want to separate the interface from

the implementation of a dataset type.

201 As an app developer, I want to only depend on the interface of a dataset type in my app and have the system inject the implementation at runtime.

202 As an app developer, I want to write unit tests for an app that depends on the interface of a dataset type.

203 As a dataset developer, I want to assign explicit versions to the code of a dataset type.

204 As a dataset developer, I want to deploy a new version of a dataset type without affecting the dataset instances of that type.

205 As an app developer, I want to create a dataset instance with a specific version of a dataset type.

206 As a dataset developer, I want to explore a dataset instance created from a dataset type that was deployed by itself.

207 As a dataset developer, I want to delete outdated versions of a dataset type and I expect this to fail if there are any dataset instances with that version of the type.

208 As a dataset developer, I want to list all dataset instances that use a dataset type or a specific version of a type.

209 As a data scientist, I want to be able to create a dataset instance of an existing dataset type without writing code.

210 As a data scientist, I want to be able to upgrade a dataset instance to a new version of its code.

211 As a hydrator user, I want to create a pipeline that reads or writes an existing dataset instance.

212 As a hydrator user, I want to create a pipeline that reads or writes a new dataset instance and I want to create that dataset instance as part of pipeline creation.

213 As a hydrator user, I want to specify an explicit version of the dataset types of the dataset instances created by my pipeline and I expect pipeline creation to fail if that results in incompatible upgrade of an existing dataset instance that is shared with other apps or pipelines.

214 As a hydrator user, I want to explore the datasets created by my pipeline.

215 As a hydrator user, I want to ensure that all dataset instances created by apps are available as sinks and sources for pipelines.

216 As an app developer, I want to ensure that all dataset instances created by Hydrator pipelines are accessible to the app.

217 As a plugin developer, I want to include the code for a dataset type in the plugin artifact, so that when a pipeline using this plugin is created a dataset instance of that type is created and it is explorable and available to apps.

218 As a plugin developer, I want to use a custom dataset type that was deployed independently or as part of an app inside the plugin.

219 As a plugin developer, I want to upgrade the code of a dataset type used by a dataset instance created by that plugin when I deploy a new version of the plugin and update the pipeline

to use that version.

220 As a pipeline developer, I want to upgrade a dataset instance to a newer version of the code after the pipeline was created.

221 As a dataset developer, I want to have the option of implementing an upgrade step for when a dataset instance is upgraded to a new version of the dataset type.

222 As a dataset developer, I want to have a way to reject an upgrade of a dataset instance to a newer version of it type if the upgrade is not compatible.

223 As a dataset developer, I want to have the option of implementing a migration procedure that can be run after an upgrade of a dataset instance to a new version of it type.

224 As a developer, I want to take a dataset offline, so that I can perform a long−running maintenance or migration procedure.

225 As a dataset developer, I want to implement custom administrative operations such as ”compaction” or ”rebalance” that are no common to all dataset types.

226 As an app developer, I want to perform custom administrative operations on dataset instances from my app and the CLI and REST or the UI.

227 As a user, I want to find out what properties are supported by the dataset type what values are allowed and what the defaults are when creating a dataset instance.

228 As a user, I want to specify the schema of a dataset in a uniform way across all dataset types.

229 As a user, I want to specify schema as a JSON string.

230 As a user, I want to specify schema as a SQL schema string.

231 As a user, I want to configure time−to−live in a uniform way across all dataset types.

232 As a user, I want to see the properties that were used to configure a dataset instance.

233 As a user, I want to find out what properties of a dataset can be updated.

234 As a user, I want to update the properties of a dataset instance and I expect this to fail if the new properties are not compatible with a meaningful error message.

235 As a user, I want to update a single property of a dataset instance without knowing all other properties.

236 As a user, I want to remove a single property of a dataset instance without knowing all other properties.

237 As a user, I want to trigger a migration process for a dataset if updating its properties requires that.

238 As a user, I want to ensure that if reconfiguration of a dataset fails then no changes have taken effect, so that all steps required to reconfigure a dataset must be done as a single atomic action.

239 As an app developer, I want to ensure that application creation fails if any of its datasets cannot be created.

240 As an app developer, I want to ensure that application redeployment fails if any of its datasets cannot be reconfigured.

241 As an app developer , I want to tolerate existing datasets if their properties are different but compatible when creating a dataset as part of app deployment .
242 As a pipeline designer , I want to get a meaningful error message when pipeline creation fails when I use an existing dataset as a sink or source , so that I know that the schema or any other property of the dataset is incompatible with what the pipeline requires .
243 As a user , I want to specify as part of dataset configuration whether it is explorable .
244 As a user , I want to specify the explore schema separately .
245 As a user , I want to ensure that dataset creation fails if the dataset cannot be enabled for explore .
246 As a user , I want to ensure that dataset reconfiguration fails if the corresponding update of the explore table fails .
247 As a user , I want to ensure that a dataset operation fails if it fails to make its required changes to explore .
248 As a user , I want to ensure that an update of explore never leads to silent loss of data or data available for explore .
249 As a user , I want to enable explore for a dataset that was not configured for explore initially .
250 As a user , I want to disable explore for a dataset that was configured for explore initially .

B Requirements Output

B.1 Recycling 101

1 As an employee from the HR department , I want to keep usage stats and locations
2 As an admin account , I want to keep all power users ' activities
3 As an admin account , I want to keep my personal information
4 As a superuser , I want to handle a record of it
5 As a superuser , I want to handle usage stats and immediate area
6 As a company , I want to keep specific users based on IP address
7 As a senior executive , I want to keep all users ' certain activity
8 As an admin , I want to keep all immediate area of recycling centers
9 As an admin , I want to keep all users ' certain activity
10 As a recyclingfacility , I want to keep I
11 As a bad employee , I want to keep all users ' activities
12 As an admin account , I want to keep a record of it
13 As a user , I want to keep specific users based on vpn client
14 As a developer , I want to keep usage stats and locations
15 As a bad employee from the HR department , I want to handle all locations of landfills
16 As an employee , I want to handle usage stats and immediate area
17 As an employee from the HR department , I want to keep usage stats and locations
18 As an employee from the HR department , I want to keep all power

```
     users ' activities
19 As a user interaction , I want to keep all power users '
     activities
20 As a bad employee from the HR department , I want to handle
     usage stats and locations
21 As a superuser , I want to handle I
22 As an outside company, I want to keep I
23 As a bad employee from the HR department , I want to keep a
     record of it
24 As a senior executive , I want to keep all locations of
     landfills
25 As an employee from the HR department , I want to handle my
     private info
26 As an outside company, I want to handle all users ' certain
     activity
27 As a bad employee , I want to keep specific users based on vpn
     client
28 As a company, I want to keep specific users based on vpn client
29 As a developer , I want to keep usage statistics and locations
30 As a recyclingfacility representative , I want to keep which
     ones
31 As a developer , I want to keep my personal information
32 As an admin account , I want to keep random users based on IP
     address
33 As a bad employee from the HR department , I want to keep all
     power users ' activities
34 As a superuser , I want to keep all users ' activities
35 As a user interaction , I want to keep all power users '
     activities
36 As an admin, I want to keep usage statistics and locations
37 As a developer , I want to keep which ones
38 As a main developer , I want to keep a map display
39 As a bad employee , I want to keep all locations of recycling
     centers
40 As an executive , I want to keep all users ' activities
41 As a user , I want to keep all power users ' activities
42 As a company accountant , I want to keep my private info
43 As a user interaction , I want to keep my personal information
44 As an outside company, I want to keep usage stats and immediate
     area
45 As a company, I want to keep a record of it
46 As an employee from the HR department , I want to keep usage
     stats and immediate area
47 As an admin account , I want to keep a map display
48 As an executive , I want to keep specific users based on IP
     address
49 As a bad employee , I want to keep specific users based on vpn
     client
50 As a user , I want to keep I
```

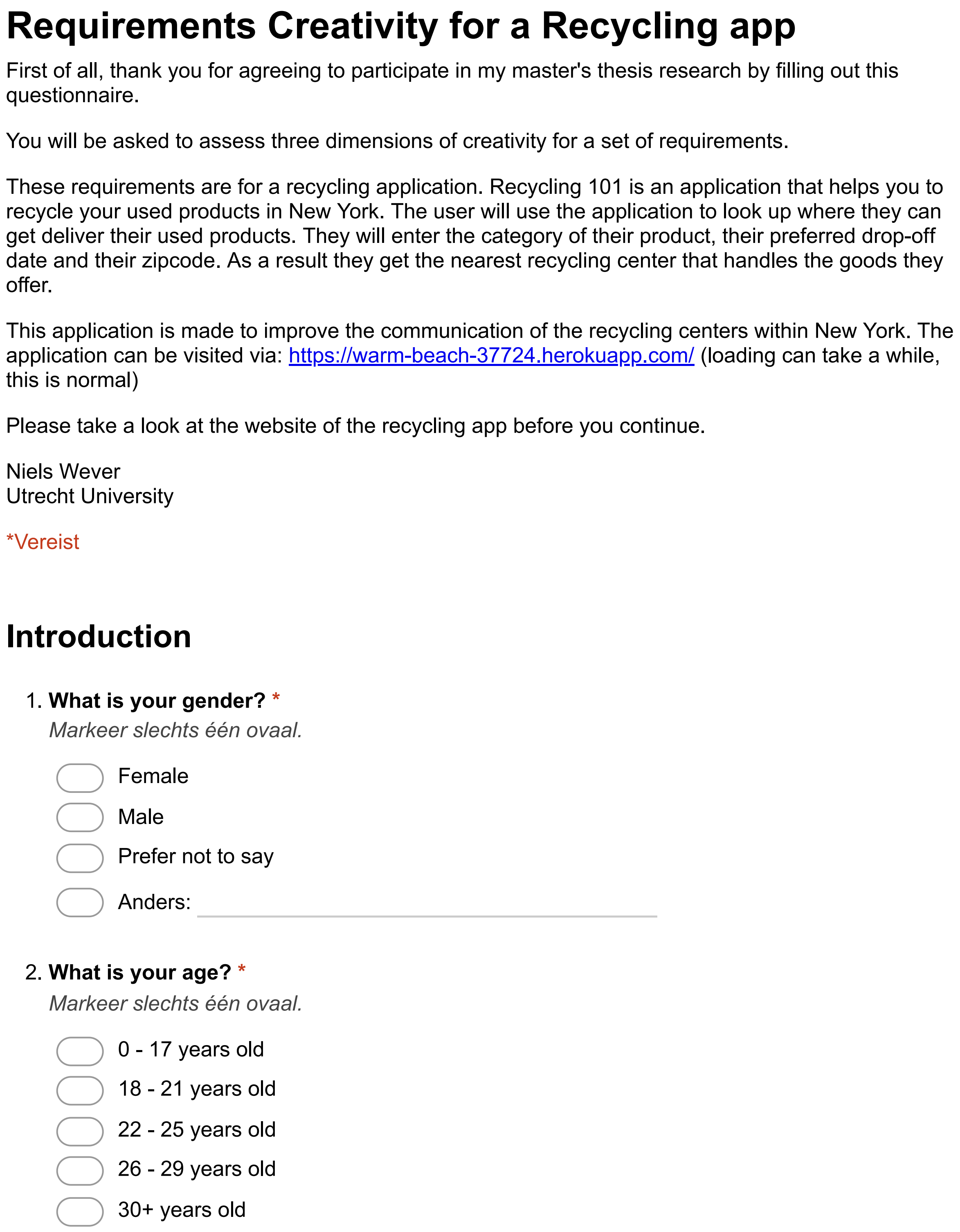
B.2 Datahub

```
 1 As a proprietor , I want to take key metrics about usage such as
```

```
      users, API usage
 2  As a Consumer, I want to take key configuration parameters for
      my site deployment from membership in my Zenimax
 3  As an Owner, I want to write all the Normal people
 4  As a API user, I want to write
 5  As a main developer, I want to set the code of a dataset type
      in my how many apps' artifact and create a dataset of that
      type when deploying the app. a Data Package
 6  As a user interaction, I want to set as part of dataset
      configuration whether it is explorable at runtime
 7  As a Developer, I want to set the dataset code
 8  As a user interaction, I want to set on upon a Data Package no
      appropriate changes
 9  As a API user, I want to exist on the computer monitor of all
      monetary measures
10  As a Data Analyst, I want to inject more examples of published
      packages at runtime
11  As a ResearcherPublisher, I want to inject a Data Package with
      the code of a dataset type in my app artifact and create a
      dataset of that type when deploying the app.
12  As a developer, I want to inject few units tests for an app
      that depends on the interface of a dataset type
13  As a Repository Manager, I want to inject at runtime
14  As a Researcher, working with good data in NetCDF, I want to
      search particular guy for this to fail
15  As a proprietor, I want to search for how active the sites iOS
16  As a Software Engineer, I want to search a dataset instance of
      an only one type that was deployed as part of an app. for
      this to fail if there are any dataset instances with that
      older version of the type
17  As a hydrator user, I want to search for a database table that
      was not configured for explore initially
18  As a DeveloperDataWrangler, I want to inject few units tests
      for an app that depends on the interface of a dataset type a
       Data Package
19  As a Data Analyst, I want to inject at optimized code with all
      the Normal people
20  As a Publisher, I want to inject with additional GeoJSON
      sources
21  As a Data Analyst, I want to inject a Data Package
22  As an Industry, I want to start an existing dataset instance
23  As a Researcher, I want to start unit tests for an app that
      depends on the ncurses of a dataset type
24  As an Analyst, I want to start multiple versions around
      including older versions
25  As a RepositoryManagerResearcher, I want to search for a
      dataset that was not configured for explore initially for
      key metrics about own usage such as users, API usage
26  As a proprietor, I want to search for this to fail iOS
27  As a Civic Tech Activist, I want to search for a dataset that
      was configured for explore initially for with additional
      GeoJSON sources
```

28 As a technical sales , I want to search for a database table that was not configured for explore initially
29 As a Repository Manager, I want to load the code of a dataset type in my app artifact and create a dataset of that type when deploying the app. at runtime
30 As a Repository Manager, I want to load at runtime with that the dataset instances created by this app use the dataset type code included in this app.
31 As a ResearcherGovernment Publisher , I want to load from R
32 As a Researcher , working with data in NetCDF, I want to load at optimized code
33 As a Researcher , I want to load the app deployment at optimized code
34 As a technical sales , I want to load at optimized code with a set of monetary measures
35 As a Researcher , I want to load the schema of a dataset in a uniformly across all dataset types
36 As a Researcher , working with data in NetCDF, I want to load a Data Package
37 As a Platform administrator , I want to exist on across all dataset types
38 As a Developer , I want to write a user
39 As a publisher , I want to write
40 As a Civic Tech Activist , I want to keep examples of intended use of fiscal data visualisations
41 As a API User, I want to keep visualisation state law
42 As a Data Consuming User , I want to keep shitty business practices
43 As a Platform administrator , I want to accompany the implementation
44 As a API User, I want to accompany a user at optimized code
45 As a Publisher , I want to inject a user at optimized code
46 As a DeveloperPublisher , I want to inject a Data Package with the explore schema
47 As a pipeline developer , I want to inject the civilization
48 As a Software Engineer , I want to inject a Data Package
49 As a OpenSpending Community Member, I want to inject that the dataset instances created by this how many apps use the dataset type code included in this app. at runtime
50 As a consumer, I want to inject a Data Package with all the people
51 As a main developer , I want to inject how active the sites
52 As a user interaction , I want to inject at optimized code

C Questionnaires



Requirements Creativity for a Recycling app

First of all, thank you for agreeing to participate in my master's thesis research by filling out this questionnaire.

You will be asked to assess three dimensions of creativity for a set of requirements.

These requirements are for a recycling application. Recycling 101 is an application that helps you to recycle your used products in New York. The user will use the application to look up where they can get deliver their used products. They will enter the category of their product, their preferred drop-off date and their zipcode. As a result they get the nearest recycling center that handles the goods they offer.

This application is made to improve the communication of the recycling centers within New York. The application can be visited via: https://warm-beach-37724.herokuapp.com/ (loading can take a while, this is normal)

Please take a look at the website of the recycling app before you continue.

Niels Wever
Utrecht University

*Vereist

Introduction

1. **What is your gender?** *

Markeer slechts één ovaal.

○ Female
○ Male
○ Prefer not to say
○ Anders: ______________________________

2. **What is your age?** *

Markeer slechts één ovaal.

○ 0 - 17 years old
○ 18 - 21 years old
○ 22 - 25 years old
○ 26 - 29 years old
○ 30+ years old

3. **What is the highest degree or level of school you have completed?** *

Markeer slechts één ovaal.

○ High School
○ First year bachelor
○ Bachelor’s degree
○ Master's degree
○ Professional degree
○ Doctorate degree
○ Anders: ______________________

4. **How experienced are you in identifying or writing software requirements?** *

Markeer slechts één ovaal.

○ I have no experience with software requirements
○ I am somewhat experienced with software requirements
○ I am experienced with software requirements
○ I am an expert on software requirements

User stories

User stories are used to express requirements and communicate them in a structured way, while capturing the goals of different kinds of users and their reason behind it. The structure of a user story is as follows:

As a [user], I want to [goal], so that I [reason].

Example:
"As a user, I want to be able to enter my zip code and get a list of nearby recycling facilities, so that I can determine which ones I should consider."
Where:
- [user] = user
- [goal] = be able to enter my zip code and get a list of nearby recycling facilities
- [reason] = can determine which ones I should consider

The reason is an optional component of a user story. Therefore, another example without a reason is the following:
"As a user, I want to be able to enter my zip code and get a list of nearby recycling facilities"

Other examples of user stories are:
- As a user, I want to get feedback when I enter an invalid zip code. ([user] = user; [goal] = get feedback when I enter an invalid zip code)
- As an admin, I want to have all data encrypted. ([user] = admin; [goal] = have all data encrypted)
- As an employee, I want to get quick notifications. ([user] = employee; [goal] = get quick notifications)

Recycling app

Recycling 101 is an application that helps you to find the closest destination to recycle your used products in New York. The user will use the application to look up where they can get rid of their used products. They will enter the category of their product, their preferred drop-off date and their zipcode. As a result they get the nearest recycling center that handles the goods they offer.

This application is made to improve the communication of the recycling centers within New York. Recycling centers themselves can add their opening hours, contact details and recycling possibilities on the application so they can be found by users of the application.

Next to this, users can make a profile and add their favorite recycling centers to their own profile.

The application can be visited via: https://warm-beach-37724.herokuapp.com/ (loading can take a while, this is normal)

Requirements - Usefulness

With the next questions, you will assess the usefulness of a set of requirements. A requirement is useful when it specifies a workable and effective way to solve a problem. Keep the Recycling application and domain in mind, while assessing usefulness. A short domain description is displayed below to recap.

Recycling app

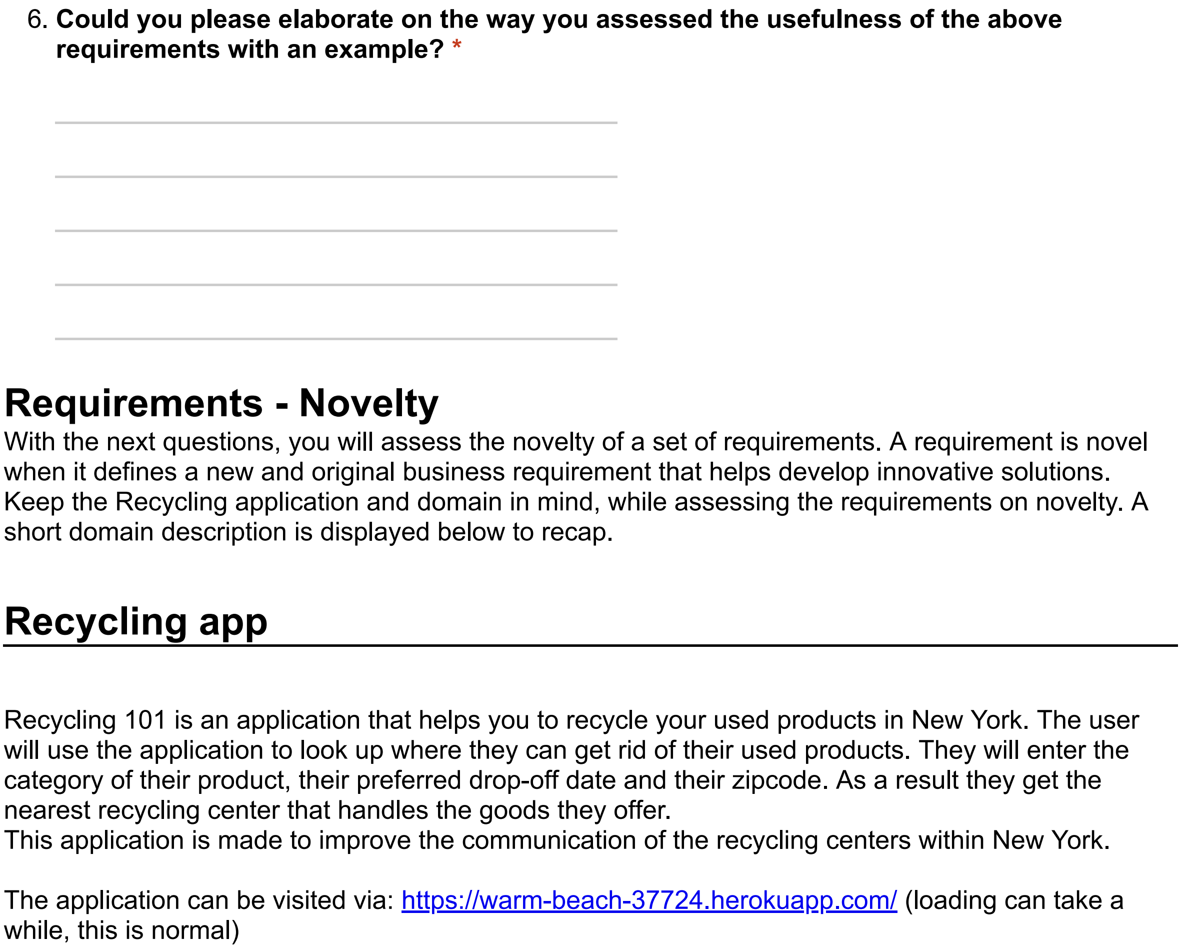
Recycling 101 is an application that helps you to recycle your used products in New York. The user will use the application to look up where they can get rid of their used products. They will enter the category of their product, their preferred drop-off date and their zipcode. As a result they get the nearest recycling center that handles the goods they offer.
This application is made to improve the communication of the recycling centers within New York.

The application can be visited via: https://warm-beach-37724.herokuapp.com/ (loading can take a while, this is normal)

5. **On a scale from 1 to 4 would you rate the following requirements on usefulness?** *

Markeer slechts één ovaal per rij.

	1. Not useful	2. Somewhat useful	3. Useful	4. Extremely useful
As a user, I want to click on the address.	○	○	○	○
As an employee from the HR department, I want to keep usage stats and locations	○	○	○	○
As a superuser, I want to handle usage stats and immediate area	○	○	○	○
As a user, I want to add donation centers as favorites on my profile.	○	○	○	○
As a company, I want to keep specific users based on IP address	○	○	○	○
As a bad employee from the HR department, I want to handle all locations of landfills	○	○	○	○
As an admin, I want to be able to block specific users based on IP address.	○	○	○	○
As an employee from the HR department, I want to keep usage stats and locations	○	○	○	○
As a senior executive, I want to keep all locations of landfills	○	○	○	○
As an employee from the HR department, I want to handle my private info	○	○	○	○
As a company, I want to keep specific users based on vpn client	○	○	○	○
As an admin, I want to keep usage statistics and locations	○	○	○	○



6. **Could you please elaborate on the way you assessed the usefulness of the above requirements with an example?** *

Requirements - Novelty

With the next questions, you will assess the novelty of a set of requirements. A requirement is novel when it defines a new and original business requirement that helps develop innovative solutions. Keep the Recycling application and domain in mind, while assessing the requirements on novelty. A short domain description is displayed below to recap.

Recycling app

Recycling 101 is an application that helps you to recycle your used products in New York. The user will use the application to look up where they can get rid of their used products. They will enter the category of their product, their preferred drop-off date and their zipcode. As a result they get the nearest recycling center that handles the goods they offer.
This application is made to improve the communication of the recycling centers within New York.

The application can be visited via: https://warm-beach-37724.herokuapp.com/ (loading can take a while, this is normal)

7. **On a scale from 1 to 4 would you rate the following requirements on novelty?** *

Markeer slechts één ovaal per rij.

	1. Not novel	2. Somewhat novel	3. Novel	4. Extremely Novel
As a user, I want to click on the address.	○	○	○	○
As an employee from the HR department, I want to keep usage stats and locations	○	○	○	○
As a superuser, I want to handle usage stats and immediate area	○	○	○	○
As a user, I want to add donation centers as favorites on my profile.	○	○	○	○
As a company, I want to keep specific users based on IP address	○	○	○	○
As a bad employee from the HR department, I want to handle all locations of landfills	○	○	○	○
As an admin, I want to be able to block specific users based on IP address.	○	○	○	○
As an employee from the HR department, I want to keep usage stats and locations	○	○	○	○
As a senior executive, I want to keep all locations of landfills	○	○	○	○
As an employee from the HR department, I want to handle my private info	○	○	○	○
As a company, I want to keep specific users based on vpn client	○	○	○	○
As an admin, I want to keep usage statistics and locations	○	○	○	○

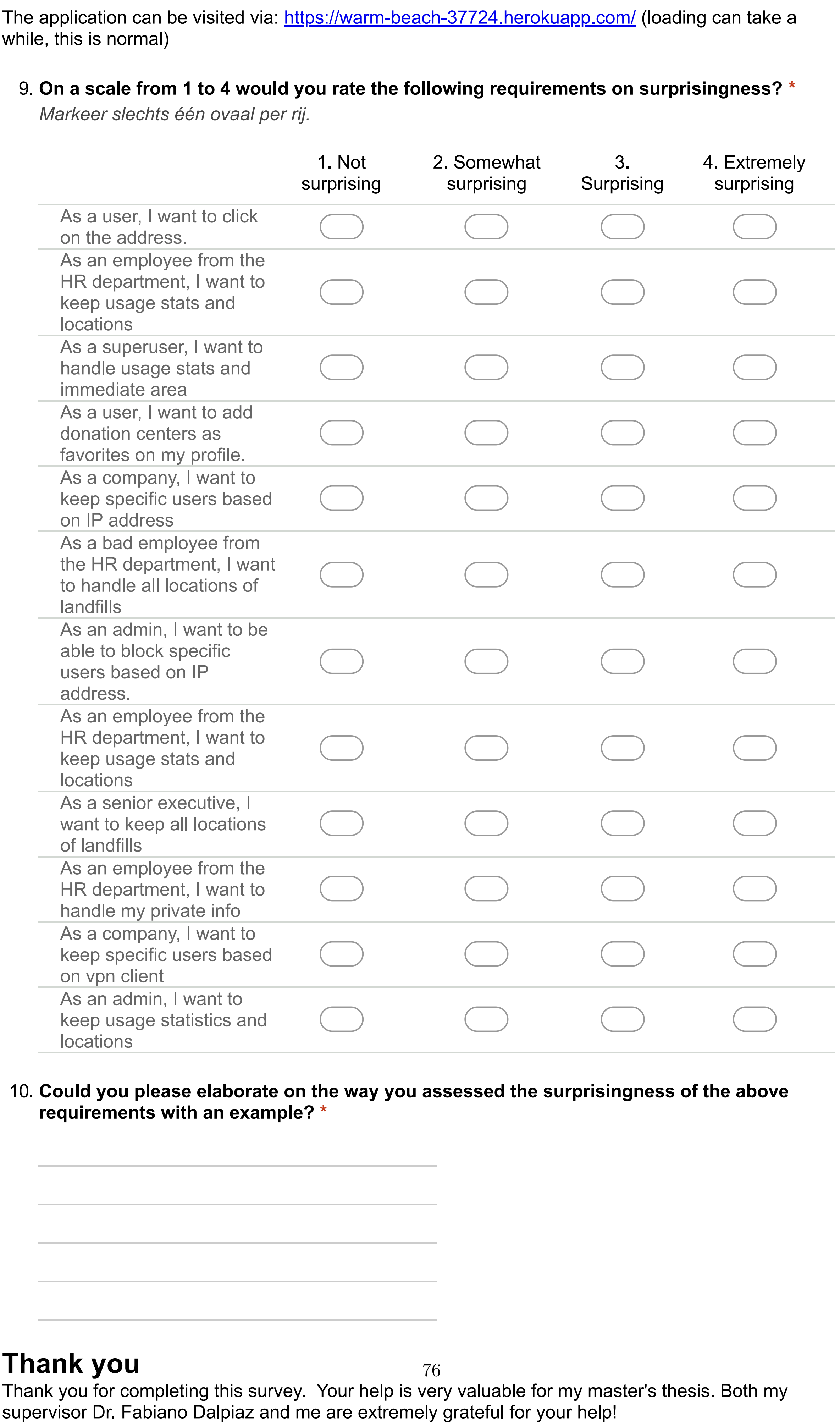
8. **Could you please elaborate on the way you assessed the novelty of the above requirements with an example?** *

Requirements - Surprisingness

With the next questions, you will assess the surprisingness of a set of requirements. A requirement is surprising when it describes something that is unusual, unexpected or something that "may shock or amaze us". Keep the Recycling application and domain in mind, while assessing the requirements on surprisingness. A short domain description is displayed below to recap.

Recycling app

Recycling 101 is an application that helps you to recycle your used products in New York. The user will use the application to look up where they can get rid of their used products. They will enter the category of their product, their preferred drop-off date and their zipcode. As a result they get the nearest recycling center that handles the goods they offer.
This application is made to improve the communication of the recycling centers within New York.



The application can be visited via: https://warm-beach-37724.herokuapp.com/ (loading can take a while, this is normal)

9. **On a scale from 1 to 4 would you rate the following requirements on surprisingness?** *

Markeer slechts één ovaal per rij.

	1. Not surprising	2. Somewhat surprising	3. Surprising	4. Extremely surprising
As a user, I want to click on the address.	○	○	○	○
As an employee from the HR department, I want to keep usage stats and locations	○	○	○	○
As a superuser, I want to handle usage stats and immediate area	○	○	○	○
As a user, I want to add donation centers as favorites on my profile.	○	○	○	○
As a company, I want to keep specific users based on IP address	○	○	○	○
As a bad employee from the HR department, I want to handle all locations of landfills	○	○	○	○
As an admin, I want to be able to block specific users based on IP address.	○	○	○	○
As an employee from the HR department, I want to keep usage stats and locations	○	○	○	○
As a senior executive, I want to keep all locations of landfills	○	○	○	○
As an employee from the HR department, I want to handle my private info	○	○	○	○
As a company, I want to keep specific users based on vpn client	○	○	○	○
As an admin, I want to keep usage statistics and locations	○	○	○	○

10. **Could you please elaborate on the way you assessed the surprisingness of the above requirements with an example?** *

____

____

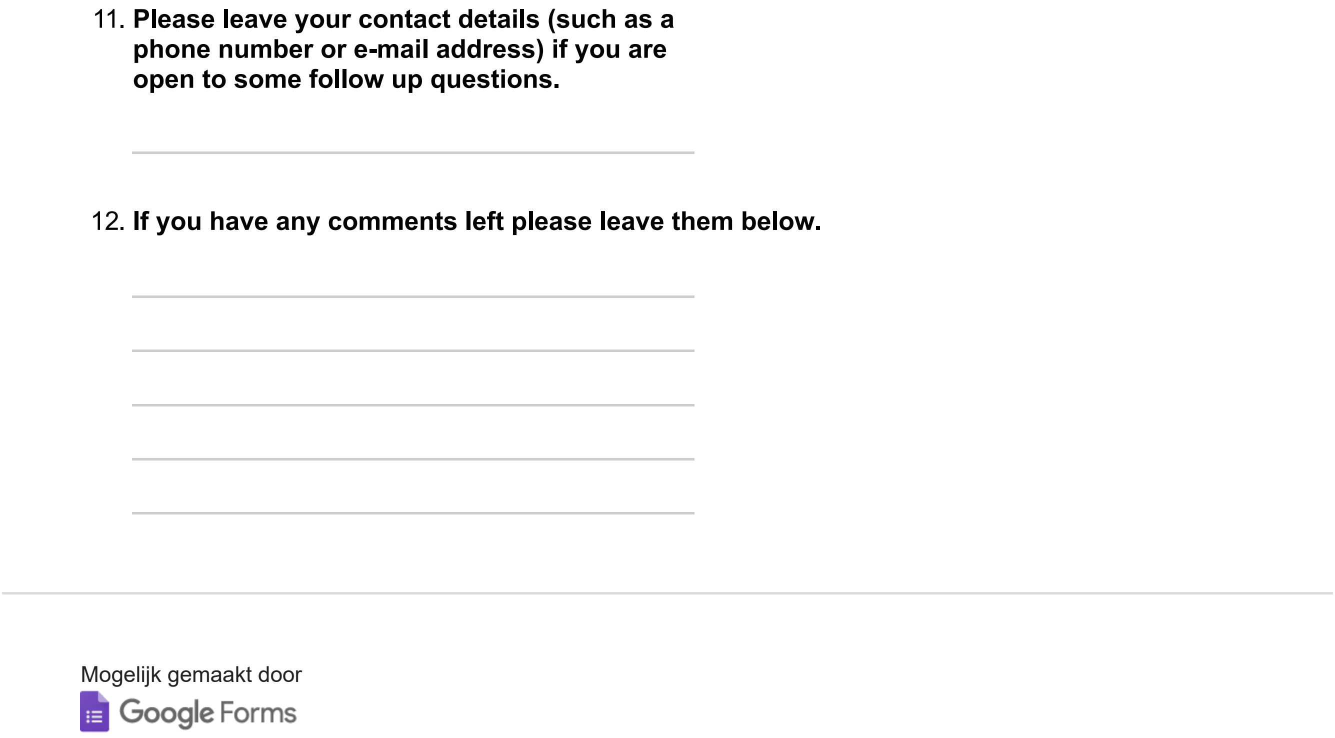
____

____

____

Thank you

Thank you for completing this survey. Your help is very valuable for my master's thesis. Both my supervisor Dr. Fabiano Dalpiaz and me are extremely grateful for your help!



11. **Please leave your contact details (such as a phone number or e-mail address) if you are open to some follow up questions.**

12. **If you have any comments left please leave them below.**

Mogelijk gemaakt door Google Forms

Requirements Creativity for Datahub

First of all, thank you for agreeing to participate in my master's thesis research by filling out this questionnaire.

You will be asked to assess three dimensions of creativity for a set of requirements.

These requirements are for the Datahub application. Datahub enables its users to store, share and look up data. It does so by providing different functions/tools to improve the quality of the available data. The goal is to have a place for storing high-quality datasets that can be shared with others. Datahub can be visited at: https://datahub.io

Please take a look at the Datahub website before you continue.

Niels Wever
Utrecht University

*Vereist

Introduction

1. **What is your gender?** *

Markeer slechts één ovaal.

○ Female
○ Male
○ Prefer not to say
○ Anders: ____________________

2. **What is your age?** *

Markeer slechts één ovaal.

○ 0 - 17 years old
○ 18 - 21 years old
○ 22 - 25 years old
○ 26 - 29 years old
○ 30+ years old

3. **What is the highest degree or level of school you have completed?** *

Markeer slechts één ovaal.

○ High School
○ First year bachelor
○ Bachelor’s degree
○ Master's degree
○ Professional degree
○ Doctorate degree
○ Anders: ____________________

4. **How experienced are you in identifying or writing software requirements?** *

Markeer slechts één ovaal.

○ I have no experience with software requirements
○ I am somewhat experienced with software requirements
○ I am experienced with software requirements
○ I am an expert on software requirements

User stories

User stories are used to express requirements and communicate them in a structured way, while capturing the goals of different kinds of users and their reason behind it. The structure of a user story is as follows:

As a [user], I want to [goal], so that I [reason].

Example:
"As a publisher, I want to publish a dataset, so that I can share the dataset publicly with everyone."
Where:
- [user] = publisher
- [goal] = publish a dataset
- [reason] = can share the dataset publicly with everyone

The reason is an optional component of a user story. Therefore, another example without a reason is the following:
"As a publisher, I want to publish a dataset"

Other examples of user stories are:
- As a consumer, I want to view a data package online. ([user] = consumer; [goal] = view a data package online)
- As a web developer, I want to be able to install multiple versions of the same datapackage separately. ([user] = web developer; [goal] = be able to install multiple versions of the same datapackage separately)
- As a publisher, I want to permanently delete a data package. ([user] = publisher; [goal] = permanently delete a data package)

Datahub.io

Datahub allows to store, share and look up data. The Datahub application provides tools to do so, aiming to improve the quality of the available data. The goal is to have a place to have high quality datasets that can be shared with others.

One of the main features of Datahub is creating a datapackage of a set of data files. Basic info such as the author, license, list of files, data structure, etc. can be added to the package. Datahub supports this to have a "very simple, web friendly, standardized and extensible" approach (https://datahub.io/docs/data-packages).

While uploading data in data packages to Datahub, Datahub is able to validate whether the uploaded data is stored in the way as described in the meta-data of of the data package. For example, this validation includes an automatic check of whether all columns are filled in.

Next to creating these data packages, Datahub supports a way to publish and deploy them. Applications are available for Windows, MacOS and Linux and can be used to publish the data. Next to this, users can configure with whom you want to share the data. This could be public or within a specified group.

The data available can be downloaded and used in multiple ways. A script in python, R or JavaScript for example could retrieve data packages from Datahub and use them in an application.

Datahub can be visited via https://datahub.io

Requirements - Usefulness

With the next questions, you will assess the usefulness of a set of requirements. A requirement is useful when it specifies a workable and effective way to solve a problem. Keep the Datahub application and domain in mind, while assessing usefulness. A short domain description is displayed below to recap.

Datahub

Datahub allows to store, share and look up data. The Datahub application provides tools to do so, aiming to improve the quality of the available data. The goal is to have a place to have high quality datasets that can be shared with others.

Datahub can be visited via https://datahub.io

5. **On a scale from 1 to 4, how would you rate the following requirements on usefulness?** *

   Markeer slechts één ovaal per rij.

	1. Not useful	2. Somewhat useful	3. Useful	4. Extremely useful
As a Visitor, I want to sign up via Github or Google.	○	○	○	○
As a proprietor, I want to take key metrics about usage such as users and API usage	○	○	○	○
As a Developer, I want to set the dataset code	○	○	○	○
As a Publisher, I want to see real examples of published packages.	○	○	○	○
As a API user, I want to exist on the computer monitor of all monetary measures	○	○	○	○
As a Data Analyst, I want to inject more examples of published packages at runtime	○	○	○	○
As a Publisher, I want to data to be validated when I publish it.	○	○	○	○
As a hydrator user, I want to search for a database table that was not configured for explore initially	○	○	○	○
As a DeveloperDataWrangler, I want to inject few units tests for an app that depends on the interface of a dataset type a Data Package	○	○	○	○
As an Industry, I want to start an existing dataset instance	○	○	○	○
As a technical sales, I want to search for a database table that was not configured for explore initially	○	○	○	○
As a Repository Manager, I want to load the code of a dataset type in my app artifact and create a dataset of that type when deploying the app. at runtime	○	○	○	○

6. **Could you please elaborate on the way you assessed the usefulness of the above requirements with an example?** *

Requirements - Novelty

With the next questions, you will assess the novelty of a set of requirements. A requirement is novel when it defines a new and original business requirement that helps develop innovative solutions. Keep the Datahub application and domain in mind, while assessing the requirements on novelty. A short domain description is displayed below to recap.

Datahub

Datahub allows to store, share and look up data. The Datahub application provides tools to do so, aiming to improve the quality of the available data. The goal is to have a place to have high quality datasets that can be shared with others.

Datahub can be visited via https://datahub.io

7. **On a scale from 1 to 4, how would you rate the following requirements on novelty?** *

Markeer slechts één ovaal per rij.

	1. Not novel	2. Somewhat novel	3. Novel	4. Extremely novel
As a Visitor, I want to sign up via Github or Google.	○	○	○	○
As a proprietor, I want to take key metrics about usage such as users and API usage	○	○	○	○
As a Developer, I want to set the dataset code	○	○	○	○
As a Publisher, I want to see real examples of published packages.	○	○	○	○
As a API user, I want to exist on the computer monitor of all monetary measures	○	○	○	○
As a Data Analyst, I want to inject more examples of published packages at runtime	○	○	○	○
As a Publisher, I want to data to be validated when I publish it.	○	○	○	○
As a hydrator user, I want to search for a database table that was not configured for explore initially	○	○	○	○
As a DeveloperDataWrangler, I want to inject few units tests for an app that depends on the interface of a dataset type a Data Package	○	○	○	○
As an Industry, I want to start an existing dataset instance	○	○	○	○
As a technical sales, I want to search for a database table that was not configured for explore initially	○	○	○	○
As a Repository Manager, I want to load the code of a dataset type in my app artifact and create a dataset of that type when deploying the app. at runtime	○	○	○	○

8. **Could you please elaborate on the way you assessed the novelty of the above requirements with an example?** *

Requirements - Surprisingness

With the next questions, you will assess the surprisingness of a set of requirements. A requirement is surprising when it describes something that is unusual, unexpected or something that "may shock or amaze us". Keep the Datahub application and domain in mind, while assessing the requirements on surprisingness. A short domain description is displayed below to recap.

Datahub

Datahub allows to store, share and look up data. The Datahub application provides tools to do so, aiming to improve the quality of the available data. The goal is to have a place to have high quality datasets that can be shared with others.

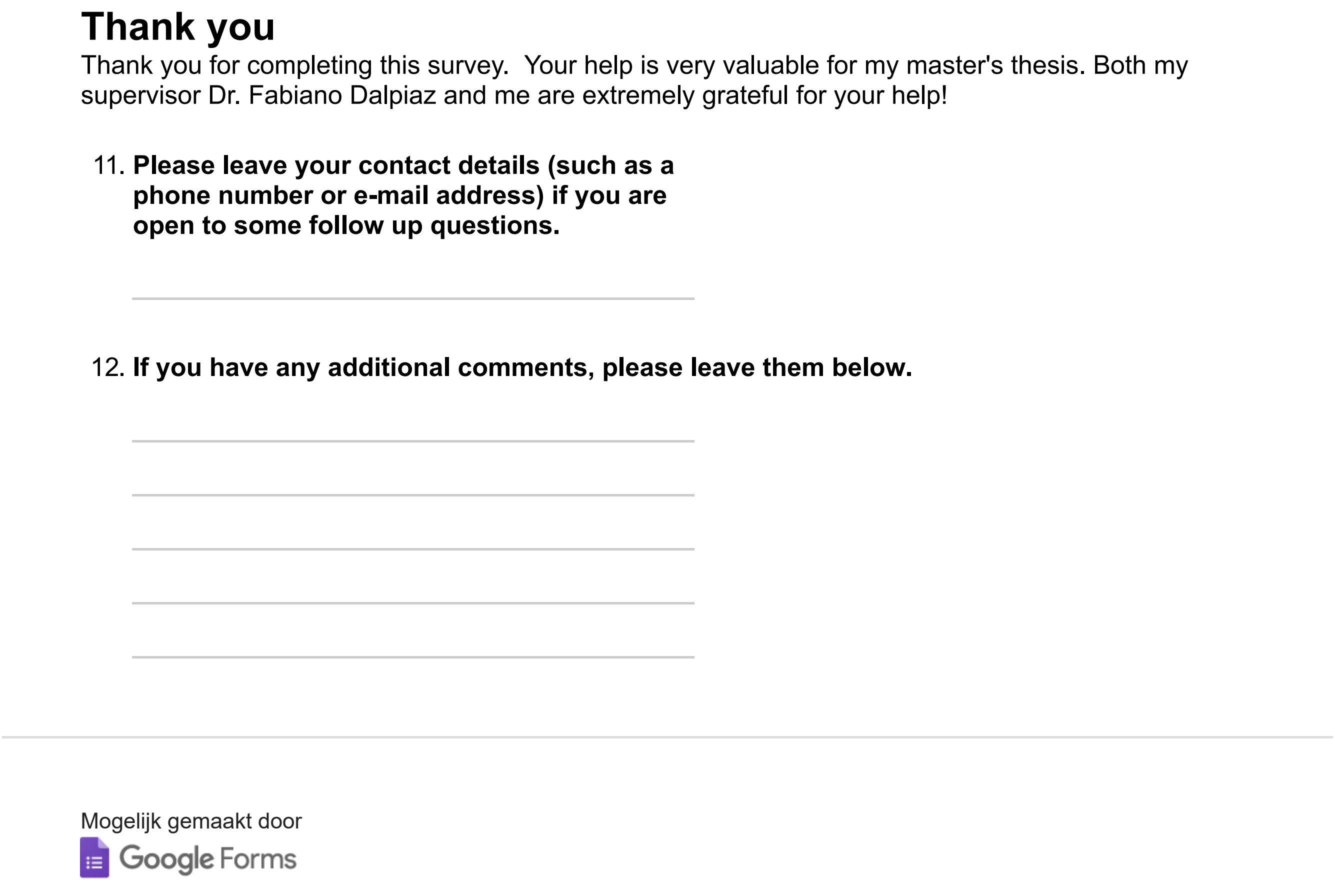
Datahub can be visited via https://datahub.io

9. **On a scale from 1 to 4, how would you rate the following requirements on surprsingness?** *

Markeer slechts één ovaal per rij.

	1. Not surprising	2. Somewhat surprising	3. Surprising	4. Extremely surprising
As a Visitor, I want to sign up via Github or Google.	○	○	○	○
As a proprietor, I want to take key metrics about usage such as users and API usage	○	○	○	○
As a Developer, I want to set the dataset code	○	○	○	○
As a Publisher, I want to see real examples of published packages.	○	○	○	○
As a API user, I want to exist on the computer monitor of all monetary measures	○	○	○	○
As a Data Analyst, I want to inject more examples of published packages at runtime	○	○	○	○
As a Publisher, I want to data to be validated when I publish it.	○	○	○	○
As a hydrator user, I want to search for a database table that was not configured for explore initially	○	○	○	○
As a DeveloperDataWrangler, I want to inject few units tests for an app that depends on the interface of a dataset type a Data Package	○	○	○	○
As an Industry, I want to start an existing dataset instance	○	○	○	○
As a technical sales, I want to search for a database table that was not configured for explore initially	○	○	○	○
As a Repository Manager, I want to load the code of a dataset type in my app artifact and create a dataset of that type when deploying the app. at runtime	○	○	○	○

10. **Could you please elaborate on the way you assessed the surprisingness of the above requirements with an example?** *



Thank you

Thank you for completing this survey. Your help is very valuable for my master's thesis. Both my supervisor Dr. Fabiano Dalpiaz and me are extremely grateful for your help!

11. **Please leave your contact details (such as a phone number or e-mail address) if you are open to some follow up questions.**

    ____

12. **If you have any additional comments, please leave them below.**

    ____

    ____

    ____

    ____

    ____

Mogelijk gemaakt door
Google Forms

D Questionnaire Results

D.1 Recycling 101

Table 12: Demographics of the Recycling 101 questionnaire respondents

Respondent	Gender	Age	Highest completed degree	Level of experience with software requirements
R01	Male	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R02	Male	26 - 29 years old	Master's degree	I am experienced with software requirements
R03	Male	22 - 25 years old	Bachelor's degree	I am somewhat experienced with software requirements
R04	Male	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R05	Male	18 - 21 years old	First year bachelor	I am somewhat experienced with software requirements
R06	Male	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R07	Male	22 - 25 years old	Bachelor's degree	I am somewhat experienced with software requirements
R08	Male	22 - 25 years old	First year bachelor	I am experienced with software requirements
R09	Female	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R10	Male	30+ years old	Bachelor's degree	I am experienced with software requirements

The requirement labels used in Table 13, 15, and 17 correspond to the labels shown in Table 10.

Table 13: Assessment on usefulness of the respondents of the Recycling 101 questionnaire. Likert scale: 1. Not useful, 2. Somewhat useful, 3. Useful, and 4. Extremely useful

	HR1	**SR2**	**SR3**	**HR4**	**SR5**	**SR6**	**HR7**	**SR8**	**SR9**	**SR10**	**SR11**	**SR12**
R01	4	3	1	3	1	1	2	3	1	2	1	3
R02	3	3	2	4	2	4	2	3	2	4	2	4
R03	1	2	1	4	2	1	3	3	2	3	1	2
R04	1	2	2	3	1	1	3	2	3	2	2	2
R05	3	3	2	3	1	1	1	3	2	3	1	3
R06	4	3	3	3	2	2	3	2	1	1	1	3
R07	4	3	2	3	1	2	3	3	2	2	2	3
R08	1	3	2	4	2	1	3	3	1	1	1	3
R09	4	2	3	1	1	1	2	2	3	4	2	4
R10	3	4	3	1	2	1	2	3	3	2	2	4

Table 14: Motivation of the assessment on usefulness of the respondents of the Recycling 101 questionnaire.

	Motivation
R01	Logic, internal structure and applicability
R02	usefullness as in that it would improve application or make to application more successful/popular for a certain role.
R03	I tried to imagine what the resulting functionality would look like. Words such as 'keep' don't describe a functionality and are very vague, therefore I found those user stories not useful.
R04	I assessed the user stories both on the merit of the story and how actionable they are. For example: keeping usage stats an locations sounds useful but it isn't very clear which stats should be kept. That's why I haven't marked those stories higher.
R05	I though about what the application does and how the requirements applies on this application. For example: While people cannot post anything on this app there is no need to be apple to block certain users. Furthermore I though about the ethics of the requimerments and/or privacy involved and ways such that one person can break the application. (bad HR employee)
R06	The user has the most import role regarding the website. The website is build to support de user. Therefore, all of the requirements of the user should be considered as most useful/important.
R07	"As a user, I want to add donation centers as favorites on my profile". This makes sense as it eases lookup of the location and contacting, though it isn't strictly app-breaking if it couldn't be done.
R08	The usefulness of the above requirements were assessed by looking at the role of the user in combination with the desired output.
R09	I try to imagine for each case and user whether or not it you'd want to have this function, and if so if it's something you really need (like I can't imagine the app working as well as it should if the address is not automatically clickable as it adds so much ease of use, whereas I can't really see why hr employees should be able to handle customer ip's).
R10	I consider the requirements useful if it meet the system objective, there is only one interpretation for it. cover one of the main system functionality. i.e:the last user story in the list: "As an admin, I want to keep usage statistics and locations"

Table 15: Assessment on novelty of the respondents of the Recycling 101 questionnaire. Likert scale: 1. Not novel, 2. Somewhat novel, 3. Novel, and 4. Extremely novel

	HR1	**SR2**	**SR3**	**HR4**	**SR5**	**SR6**	**HR7**	**SR8**	**SR9**	**SR10**	**SR11**	**SR12**
R01	2	3	1	3	1	1	3	3	3	3	1	3
R02	1	2	2	3	1	1	1	2	2	1	3	2
R03	1	2	2	3	2	3	3	2	2	2	2	2
R04	1	1	1	1	3	2	1	1	2	1	2	1
R05	1	1	1	1	1	1	1	1	1	1	1	1
R06	2	3	2	2	3	2	2	3	3	2	2	2
R07	1	1	1	1	2	1	2	1	2	2	2	1
R08	2	2	2	4	2	2	2	2	2	2	2	2
R09	1	2	2	3	1	3	1	1	2	1	3	1
R10	1	2	2	1	1	1	2	3	3	2	2	4

Table 16: Motivation of the assessment on novelty of the respondents of the Recycling 101 questionnaire.

	Motivation
R01	Again on logic and internal structure but instead of applicability, I focused more on novelty
R02	How new a requirement is, so if I have ever seen a similar requirement in these kind of applications
R03	I tried to determine if the user story would result in innovative functionality.
R04	I assessed these user stories by checking whether I have seen these functionalities in other applications. For example adding users by ip address is something I have not heard of before while blocking them using ip addresses is very common.
R05	This whole application does not sound innovative to me. There are many of these kind of apps for other sectors. For example the food sector. You have tons of apps where you can look for specific restaurants in your neighbourhood.
R06	Businesses could use usage stats to determine which locations have the most traffic. This could be valuable information for new business strategies.
R07	”As a user, I want to click on the address” isn’t a novel requirement; nothing of this elicits creative thinking in my opinion
R08	Through looking at the different functionailities when keeping the domain in mind.
R09	Just thinking along the lines of ”is this really standard or have I not really heard/seen it before”. I’m somewhat confused by the ”as a *bad* employee” one though
R10	my assessment is based on your definition ” requirement that helps to develop innovative solutions”

Table 17: Assessment on surprisingness of the respondents of the Recycling 101 questionnaire. Likert scale: 1. Not surprising, 2. Somewhat surprising, 3. Surprising, and 4. Extremely surprising

	HR1	**SR2**	**SR3**	**HR4**	**SR5**	**SR6**	**HR7**	**SR8**	**SR9**	**SR10**	**SR11**	**SR12**
R01	1	2	4	2	3	4	3	1	2	3	3	2
R02	1	2	1	2	1	1	1	1	1	1	2	1
R03	1	1	3	1	4	4	2	1	2	3	4	2
R04	1	1	1	1	3	2	1	1	1	1	2	1
R05	1	1	2	1	3	4	3	2	2	1	4	1
R06	2	3	2	1	1	2	1	2	2	1	1	2
R07	2	1	1	1	1	2	2	1	2	2	2	1
R08	1	2	1	2	4	4	2	1	4	2	4	1
R09	1	1	1	1	1	3	1	1	1	1	2	1
R10	1	1	1	4	2	4	1	1	1	1	3	1

Table 18: Motivation of the assessment on novelty of the respondents of the Recycling 101 questionnaire.

	Motivation
R01	I judged them by my initial level of suprise
R02	If I would expect a requirement to be there in that application
R03	I rated based on my own surprise i experienced myself
R04	I checked whether I would expect any of these functionalities in this app. For example I think it is surprising to make user stories based on bad employees.
R05	I assessed the above requirements on surprisingness by thinking about if I would find this logical for a recyling app. About who gets to do what and why.
R06	I couldn't really relate to this question, sorry.
R07	"As a senior executive, I want to keep all locations of landfills": is interesting because you wouldn't expect a senior exec to involve him/herself into more menial tasks like this one
R08	Through looking at the relationship between surprisingness of the requirement and the user.
R09	I think gut feeling? Most are not really surprising, apart from the bad employee one and the one about keeping customers based on vpn
R10	RE that's I didn't expected, despite it usefulness

D.2 Datahub

Table 19: Demographics of the Datahub questionnaire respondents

Respondent	**Gender**	**Age**	**Highest completed degree**	**Level of experience with software requirements**
R11	Male	26 - 29 years old	Master's degree	I have no experience with software requirements
R12	Female	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R13	Male	22 - 25 years old	Bachelor's degree	I have no experience with software requirements
R14	Male	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R15	Male	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R16	Male	22 - 25 years old	Bachelor's degree	I am somewhat experienced with software requirements
R17	Male	22 - 25 years old	First year bachelor	I am somewhat experienced with software requirements
R18	Male	26 - 29 years old	Master's degree	I am somewhat experienced with software requirements
R19	Male	22 - 25 years old	Bachelor's degree	I am experienced with software requirements
R20	Male	26 - 29 years old	Master's degree	I am somewhat experienced with software requirements

The requirement labels used in Table 20, 22, and 24 correspond to the labels shown in Table 11.

Table 20: Assessment on usefulness of the respondents of the Datahub questionnaire. Likert scale: 1. Not useful, 2. Somewhat useful, 3. Useful, and 4. Extremely useful

	HR13	**SR14**	**SR15**	**HR16**	**SR17**	**SR18**	**HR19**	**SR20**	**SR21**	**SR22**	**SR23**	**SR24**
R11	2	4	4	4	4	4	3	2	3	3	4	4
R12	3	2	2	3	1	2	3	1	1	2	2	2
R13	3	4	3	3	2	2	2	2	2	3	2	3
R14	4	4	2	3	1	1	4	2	1	2	2	1
R15	2	4	3	2	1	2	4	1	1	1	1	2
R16	4	4	3	3	1	3	4	3	1	3	2	1
R17	2	3	2	4	3	2	4	4	3	4	4	3
R18	3	2	3	3	1	3	3	3	2	3	3	2
R19	4	2	1	3	1	2	2	2	1	1	3	1
R20	1	3	2	4	1	4	3	3	2	2	2	3

Table 21: Motivation of the assessment on usefulness of the respondents of the Datahub questionnaire.

	Motivation
R11	Taking into account the variety of roles that I, as a user, can fulfill, it determines whether a functionality seems useful or as a must have, or not of any importance at all. Examples are, Google or GitHub account sharing is not a must have, only somewhat useful as it does not interferes the way of working with datahub. Another example is the technical sales functionality. Exploring novel datasets for business thrives value, hence most useful.
R12	I looked at how clear the requirement was to me, because I think a requirement is only useful when it is clear what needs to be done. For example "As a hydrator user, I want to search for a database table that was not configured for explore initially". It is not clear to me what a hydrator user is, hydrate means to absorb water? And "for explore initially" is also very vague to me. This is why I marked this requirement as "not useful".
R13	Using data that is publicly available can be very useful for users. The main challenge however, is getting a lot of high-quality data from publishers in my opinion. Giving the publishers insight on how and why their data was used, and making the platform useful for them should therefore be the main priority in my opinion.
R14	In my understanding, user stories should be atomic, specific, and free of ambiguity (amongst other things). The first user story satisfies all of these aspects: a visitor wants to sign up through Google or Github, making it very useful. Some user stories are specific as well, but not atomic, and some are not that well described or not understandable (like user story 9 and 12).
R15	I look at the general usefulness of the requirement, and then look at the connection of all elements. For example, the idea of starting an existing dataset instance might be useful, but I do not think that a requirement for an industry as a whole is useful.
R16	Readability of the questions, complexity
R17	I've assessed the requirement on whether the requirement would be important for main users. For example: a visitor should sign up to be able to get access to the data packages, but to say this should be done with a google or github account is not important. It is important for the user to be able to access the datapackages with an account. Google and github seem more like an extra, thus that is why it is not a useful requirement.
R18	Based on several quality properties of user stories such as atomicity and unambiguity of the stories. E.g. As a visitor, I want to sign up via github or google can be broken down in two requirements: 1. ... signing in via google. 2. And ... signing in via github. Which is why I rated it usefull instead of extremely usefull.
R19	I compared the usefulness of these requirements, took into account that some sentences were not sound, but did not completely discredit them on that
R20	How refined the goal is. Vague goals are not useful

Table 22: Assessment on novelty of the respondents of the Datahub questionnaire. Likert scale: 1. Not novel, 2. Somewhat novel, 3. Novel, and 4. Extremely novel

	HR13	**SR14**	**SR15**	**HR16**	**SR17**	**SR18**	**HR19**	**SR20**	**SR21**	**SR22**	**SR23**	**SR24**
R11	1	1	2	1	2	1	4	4	3	1	3	3
R12	1	2	1	1	1	1	2	1	1	1	1	2
R13	1	3	3	3	3	4	3	4	3	1	3	3
R14	1	1	1	1	1	3	3	1	1	1	1	1
R15	2	3	3	3	1	3	4	2	1	2	2	3
R16	4	4	4	4	2	2	1	3	1	4	1	2
R17	1	3	2	2	3	2	4	3	4	3	1	2
R18	1	2	2	2	1	3	3	2	3	1	1	3
R19	2	3	2	3	1	3	2	2	1	1	2	1
R20	1	2	1	2	3	2	1	2	3	1	2	3

Table 23: Motivation of the assessment on novelty of the respondents of the Datahub questionnaire.

	Motivation
R11	Taking into account solutions the competitors have I've answered it in the following way; extremely novel as it does not, or marginally exists on other platforms, such as data validation.
R12	This was a hard task to me. Most requirements are very vague to me which is why it is difficult to assess their novelty. I marked a requirement as somewhat novel when I understood the requirement and thought it could lead to interesting new functions I have not heard about before. For example "As a proprietor, I want to take key metrics about usage such as users and API usage". Thinking about useful metrics on this platform could lead to some interesting new insights.
R13	Some are not novel, but rather good practices (e.g. sign up with google)
R14	When novelty is defined as a new or original business requirement, there's a lot of requirements that are not novel. The requirement about the validation of a data set after publishing it is novel, since this is usually not the case.
R15	Requirements that did not make much sense could still be novel, although they would have been rated not useful in the previous section. For example, the requirement of the hydrator user did not make much sense to me (because it's not clear what a hydrator user is), but searching for a database table that was not configured for explore initially does seem somewhat novel.
R16	Fluency, spelling mistakes
R17	I've assessed the requirement on if they could help develop innovative solutions. An example would be to check published sets. This develops innovative solutions as it checks the datapackages on correctness. An incorrect datapackage should be fix asap, to prevent the tool (or user) to develop a bad reputation.
R18	Mostly based on my own experience of similar data applications. E.g. signing up via google or github is pretty common but automatic validation of datasets during publishing is something that I do not see a lot of.
R19	I looked at the novelty of them, but kept in mind that they had to make sense
R20	i used my own experience as a developer and thought if I have heard or done something like that before

Table 24: Assessment on surprisingness of the respondents of the Datahub questionnaire. Likert scale: 1. Not surprising, 2. Somewhat surprising, 3. Surprising, and 4. Extremely surprising

	HR13	**SR14**	**SR15**	**HR16**	**SR17**	**SR18**	**HR19**	**SR20**	**SR21**	**SR22**	**SR23**	**SR24**
R11	1	2	2	1	2	2	4	4	3	2	4	4
R12	1	1	1	1	3	1	1	3	3	2	2	2
R13	1	2	2	3	2	2	2	4	2	2	4	2
R14	1	1	1	1	1	1	3	1	1	1	1	1
R15	2	1	1	2	2	3	3	2	2	3	2	3
R16	1	1	1	1	4	3	1	1	4	2	1	2
R17	1	3	3	3	3	3	3	2	4	1	4	2
R18	1	1	1	1	4	2	2	1	2	1	1	2
R19	2	3	2	3	1	3	2	2	2	1	3	2
R20	1	2	1	1	2	1	3	2	2	1	2	2

Table 25: Motivation of the assessment on novelty of the respondents of the Datahub questionnaire.

	Motivation
R11	As in the previous set of questions, surprisingness is measured on what's already widely available and not. Therefore, searching for opportunities through unexplored datasets may be a big winner.
R12	I marked a requirement as surprising when it contained some weird or seemingly out of place concepts. For example: "As an Industry, I want to start an existing dataset instance". It would be surprising to me if an entire industry could own one account and start a dataset. It is surprising in the way that it does not make sense to me.
R13	Some are unexpected
R14	Again, the validation of data sets is surprising, since this is usually not the case when publishing data elsewhere. Most of the other requirements describe common requirements, so they are not that surprising.
R15	This correlated somewhat with novelty of requirements. Requirements that are very novel also are more likely to surprise me. The same goes for requirements that do not make much sense. For example, starting an existing dataset instance by a complete industry surprises me, although it would not make much sense.
R16	Compared them to my expectancy of requirements for a specific role
R17	For the roles I have assessed the requirements on if the requirement is unusual and would surprise them if it were to be implemented. The tool is made to store, share and lookup data. That's the industry requirement shouldn't be a surprise. The real examples requirement depends on how it is done. If some packages are featured, that would be a surprise. If they are standard requirement made by the publisher, it is less surprising.
R18	As a API user, I want to exist on the computer monitor of all monetary measures. – is surprising because its hard to figure out what the requirement entails. For the rest most of them are not so surprising and some pique an interest with having functionalities during runtime.
R19	I judged them on how much they surprised me initially, but once again the requirements had to make sense
R20	I looked at how well the goal fits the user. If it is a good match, it's not surprising.