Master thesis
Information Sciences

Radboud University

Liquid Requirements Engineering

Author:
Jonathan Seesink
s4143590
`j.seesink@yahoo.nl`

First supervisor/assessor:
Stijn Hoppenbrouwers
`stijnh@cs.ru.nl`

Second assessor:
Erik Barendsen
`E.Barendsen@cs.ru.nl`

February 22, 2017

Abstract

In this case study, an e-democracy approach called liquid democracy is applied for gathering feedback from a large group of software end-users. Liquid democracy aims to leverage the scalability of decision processes to a crowd of people. The outcomes gave an indication of the added value of liquid democracy techniques for gathering end-user requirements.

Traditional requirements engineering approaches such as interviews and workshops only gather requirements based on a small group of users. When eliciting requirements from a few users, lots of needs are not recognized and a user-centered prioritization of requirements is difficult.

A liquid democracy tool was used on a small scale. Interviews were conducted to investigate the added value of this technique for requirements practitioners. In addition, they were used to discover potentials and areas of concern of applying liquid democracy techniques in the field of requirements engineering.

The main liquid democracy property was not used by the users of the liquid democracy tool. The general potential of inspiration for key stakeholders have been confirmed by most of the requirements practitioners.

This research confirmed, albeit not decisively, that online requirements gathering adds value to requirements engineering processes. However, liquid democracy techniques probably did not influence the result. For using a liquid democracy tool in requirements engineering, at least the areas of concern encountered in this study need to be addressed.

Acknowledgments

At first, I want to thank my heavenly Father who gave me hope in life (thank you Jesus), health, talents, a loving family, friends who make life even more enjoyable and an environment with political and legal systems that seem to work. These are altogether important preconditions for a successful master thesis project.

This thesis is one of the last mandatory deliverables during my study of Information Sciences. I want to thank all the lecturers at Radboud University who taught me relevant topics mainly in computer science and business administration, but also in several other research fields.

Special thanks are directed to Stijn Hoppenbrouwers, Marcel de Wit, and Erik Barendsen. They provided me with valuable feedback and helped me to make decisions during the whole research process.

I also would like to thank Altagracia Solvabe and Babs Jacobs who reviewed this thesis report on mistakes in writing, my sister Annechien who encouraged and inspired me during this project, my dad who lent me a car that I used to drive to the internship company, all respondents who participated in my research (including the members of the development team of the software application that is called 'SA' in this report) and of course all employees of the internship company that facilitated my internship, helped me with my research and accompanied me during my internship.

Contents

List of Figures

List of Tables

Acronyms

CC Consultancy company

[CC] Placeholder for the real name of the CC

SA Software application

[SA] Placeholder for the real name of the SA

ORG Online requirements gathering

RE Requirements engineering

LD Liquid democracy

LDT Liquid democracy tool

Chapter 1

Introduction

This report is written in the context of a Master's thesis that is part of the Information Sciences curriculum of Radboud University, Nijmegen, the Netherlands. The research described in this report is conducted during an internship at an IT consultancy company located in the Netherlands (**CC**).

A case study is performed which investigates how *liquid democracy* (**LD**) techniques can contribute to the requirements engineering (**RE**) processes of the CC for existing software applications.

In this research, a tool that implemented LD techniques is used for gathering feedback from end-users about an existing software application. The results of using the tool, additional areas of concern of using such a tool for RE and the motivation of users to participate in the tool are investigated.

The title "*Liquid Requirements Engineering*" of this report is chosen to emphasize using LD techniques within the field of RE.

For readability reasons, words such as 'he', or 'she' are used, but often the expressions also apply for women.

In the following, the problem statement, the aim of the study, and the research questions are described. Furthermore, a contextual entity diagram and an overview about the report is provided.

1.1 Problem statement

End-users are important stakeholders of software applications (Bano & Zowghi 2015). Traditional RE approaches such as interviews and workshops only gather requirements based on a small group of users (Johann & Maalej 2015). When eliciting requirements from a few users, lots of needs are not recognized and a user-centered prioritization of requirements is difficult.

User involvement can be a big success factor in building software applications (Bano & Zowghi 2015). User involvement on online platforms indicate that users are intrinsically willing to contribute to the product's success (Maalej & Pagano 2011).

This could be improved by letting an unlimited number of people participate in the decision making process. This concept is not new. The field of e-democracy aims to communicate individuals' opinions to politicians relating to a public issue (Thomas & Streib 2005). Massive individual involvement is the foundation of democratic systems (Johann & Maalej 2015). Therefore, this research applies a specific democratic voting system for receiving feedback from end-users.

Direct democracy means that every person can vote for any proposal (Johann & Maalej 2015). Indirect democracy means that people vote for a representative who makes direct decisions (Johann & Maalej 2015). In LD, individuals can vote directly for proposals, but they can also choose to (partly) delegate their voting power to others simultaneously. They can adjust their voting anytime if they want to (Johann & Maalej 2015, Paulin 2014).

Software engineers created LD tools, but these were never used for the purpose of RE. LD techniques have the potential to gather requirements based on a large crowd of end-users of software applications.

The field of *online requirements gathering* (**ORG**) intends that end-users formulate requirements collaboratively. A tool that is made based on this philosophy is Requirements Bazaar (Renzel & Klamma 2014). It is an ORG tool that focuses on open source projects. The tool implements a voting system for proposals. However, it does not support delegation of voting power, and therefore misses the main LD property. Besides, it focuses on a negotiation process between end-users and developers. This study is rooted on the idea that feedback from end-users using LD techniques can be a valuable source of inspiration for stakeholders, such as developers.

Requirements Bazaar or the LD tools create online communities. Typically, only a small fraction of users participate in such tools. Designers of such tools should use social psychology insights to leverage participation in online communities, and therefore increase the online community's usefulness (Ling et al. 2005). The public recognition by LD techniques could motivate users to participate. That would further increase the potential value of a LD for RE.

1.2 Aim of study

The outcomes of the research will give an indication of the added value of LD techniques for the RE processes of existing software applications. New insights can be used by companies willing to improve their RE processes. Findings could also be used for a web application which makes requirements gathering possible based on a large crowd of end-users for several software applications.

1.3 Research questions

The research question is the following: *How can LD techniques contribute to the RE processes of existing software applications?* This question is split into four sub questions as described below:

1. What are the effects of using LD techniques on the requirements understanding of software developers/designers and users?
2. What are potentials and areas of concern when applying LD techniques for gathering end-user requirements of an existing web application?
3. Do end-users feel more motivated to participate in the RE processes by LD techniques?
4. How can findings be used to contribute to the RE processes of the CC?

1.4 Contextual entity diagram

Main entities used in this research, their acronyms, and their relations are depicted in Figure 1.1.

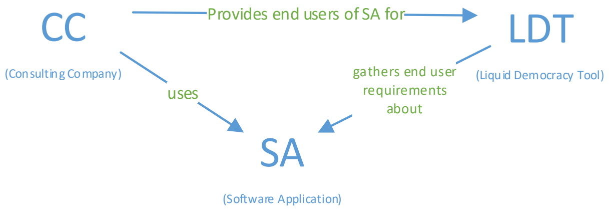

Figure 1.1: Main entities, their acronyms and their relations

1.5 Structure of this thesis report

In chapter 2 (Background), the foundations of this research are described in more detail. In chapter 3 (Setup & Method), the approach of this research is described. In chapter 4 (Results), all relevant data of this research is summarized. In chapter 5 (Discussion), the most important remarks on limits, shortcomings or unused opportunities of this research that could have influenced the results and the conclusions are described. Chapter 6 (Conclusions & further research), provides answers to the research questions and directions for further research.

Chapter 2

Background

In the following, the field of RE and LD are described in more detail. In this section, the foundations for this research are presented.

2.1 Requirements engineering

Successful software should be developed according to its purpose. RE is defined as "the process of discovering that purpose" (Nuseibeh & Easterbrook 2000). A more specific definition is provided by Zave (1997):

Requirements engineering is the branch of software engineering concerned with the real-world goals for, functions of, and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families.

This definition has been called "one of the clearest" definitions for RE because it includes goals, precise specifications and their evolution (Nuseibeh & Easterbrook 2000). Software solutions always need to be in line with their goals. RE provides theories and techniques that aim to let the solutions fit to the needs of stakeholders.

The term *engineering* is characterized by creating cost-effective solutions to practical problems, by applying scientific knowledge to build artifacts in the service of mankind (Shaw 1990).

Zave's (1997) definition contains the notion of a goal. A goal is an objective that the system under consideration should achieve. "Goals may be formulated at different levels of abstraction" (Van Lamsweerde 2001). A high-level abstraction would be "obtain higher customer satisfaction" whereas a low-level goal would be e.g. "passport photo gathered from a user".

A requirement can be seen as a statement that indicates "what the customers need from the system, described in terms of its effect in the environment" (Gunter et al. 2000). This notion mentions only the role of the customer, whereas requirements can also represent the needs of other stakeholders like developers or system operators.

Needs of stakeholders could overlap with the definition of a goal that was given above. A goal "under responsibility of a single agent in the (system to be built) becomes a requirement, whereas a goal under responsibility of a single agent in the environment of the (system to be built) becomes an assumption" (Van Lamsweerde 2001). Requirements can be realized by the system to be built, whereas assumptions should be satisfied by "organizational norms and regulations, physical laws, etc." (Van Lamsweerde 2001)

The following definition of requirements was used in this research: A *requirement* is a stakeholder's need from the system, described in terms of its effect in the environment. It is a goal which falls under responsibility of the system to be built.

A *stakeholder* is defined by someone who potentially stands to gain or lose in the process of developing the new or improved information system (Krogstie et al. 1995). This includes managers, developers, clients, suppliers, system operators, and end-users. The term "*key stakeholder*" is used for stakeholders who decide for a (large) group of end-users on software requirements.

In the field of software engineering there seems to be a need for distinguishing functional concerns ('what does the system deliver given a specific input?') and other 'non-functional' concerns (Glinz 2007).

Non-functional requirements are defined in this study by "a system requirement (1) that pertains to a performance concern, (2) that pertains to a quality concern other than the quality of meeting the functional requirements or (3) that constrains the solution space beyond what is necessary for meeting the given functional, performance, and specific quality requirements" (Glinz 2007). Among performance concerns would be e.g. timing, speed, volume and throughput, whereas a quality concern other than the quality of meeting the functional requirements would be associated with the commonly accepted "-illities" such as reliability, usability, security, availability, maintainability, etc. Physical, legal or cultural constraints would also fall under this definition of non-functional requirements.

In this research, the term *requirements practitioner* is frequently used. This is a professional who is concerned with the elicitation, analysis, documentation, and change management of requirements for software applications.

2.1.1 Techniques for RE

There are many different RE activities that can be considered "techniques". A *technique* is defined as a prescribed activity (Nuseibeh & Easterbrook 2000). These techniques can be arranged in several activity clusters (Nuseibeh & Easterbrook 2000, Paetsch et al. 2003). In this research, RE practices are linked to one or more of the following clusters: elicitation, analysis, documentation, and change management. When comparing the clusters ”Analysis and Negotiation” and ”Validation” by Paetsch et al. (2003) and ”Modelling and Analysing” and ”Agreeing” by Nuseibeh & Easterbrook (2000), it was observed that the activities linked to these clusters strongly overlap. For this reason, they were merged into the cluster ”analysis” in this research.

Elicitation

Eliciting requirements means capturing requirements based on the problems in the application domain (Nuseibeh & Easterbrook 2000). ”Understanding the application domain, business needs, system constraints, stakeholders and the problem itself is essential to gain an understanding of the system to be developed” (Paetsch et al. 2003). Techniques of requirements elicitation are traditional techniques (interviews, questionnaires, use cases/scenarios and analysis of existing documentation), group elicitation techniques (brainstorming, focus groups and RAD/JAD workshops), prototyping, modeling (KAOS, CREWS), cognitive techniques (protocol analysis, laddering, card sorting, and repertory grids), and contextual techniques (participant observation) (Nuseibeh & Easterbrook 2000).

Analysis

The analysis activities are concerned with the quality of requirements (e.g. necessity, consistency, completeness, and feasibility) and prioritization of requirements (Paetsch et al. 2003). Techniques such as pair-wise comparison, Analytic Hierarchy Process and JAD sessions can be used for this (Paetsch et al. 2003). Modeling (data-flow models, semantic data models and object-oriented approaches) also serve the analysis of requirements (Nuseibeh & Easterbrook 2000).

Requirements validation techniques are activities that ”certify that the requirements are an acceptable description of the system to be implemented” (Paetsch et al. 2003). Techniques for requirements validation are requirements reviews and requirements testing (Paetsch et al. 2003).

Requirements analysis contains tools for automated checking, such as SCR (Software Cost Reduction), to ensure that requirements are consistent and complete (Nuseibeh & Easterbrook 2000). Such tools require formal models of requirements. Techniques that test a ”correspondence with the real world problem” like prototyping, specification animation and the usage of scenarios are also arranged in this cluster (Nuseibeh & Easterbrook 2000).

Risk analysis and resolution are also activities which are linked to requirements analysis in this research.

Documentation

The activities in the cluster of documentation are concerned with recording requirements in one or several types of notations. ”The way in which requirements are documented plays an important role in ensuring that they can be read, analysed, (re-)written, and validated” (Nuseibeh & Easterbrook 2000). Documentation activities are therefore important for the other activities concerning RE. Most RE documentation research is about the notation of requirements (Nuseibeh & Easterbrook 2000). A challenge for documentation is that products are readable and traceable by many people (Nuseibeh & Easterbrook 2000).

Change Management

Requirements evolve over time. It is important to keep track of these changes. This is part of change management. It includes all activities ”concerned with the change & version control, requirements tracing, and requirements status tracking” (Paetsch et al. 2003). ”Managing inconsistency in requirements specifications as they evolve is a major challenge” (Nuseibeh & Easterbrook 2000). Traceability links ”help to scope the possible impact of change” (Nuseibeh & Easterbrook 2000). The ViewPoints framework makes ”automated support for propagation of change” possible (Nuseibeh & Easterbrook 2000).

2.2 Liquid democracy

In politics, decisions are made for large groups of people. Therefore, politics can be considered a very complex and challenging area of decision making. Democratic voting systems aim to empower the individual in political decision processes (Johann & Maalej 2015). However, the techniques used for this seem promising also for gathering requirements from end-users within software engineering (Johann & Maalej 2015). This study is rooted on the idea that feedback from end-users using specific e-democracy techniques can be a valuable source of inspiration for stakeholders, such as developers.

The field of *e-democracy* aims to communicate individuals’ opinions relating to a public issue to politicians (Thomas & Streib 2005). E-democracy ”refers to the use of information technologies and communication technologies and strategies in political and governance processes” (Manoharan & Holzer 2011).

E-participation denotes the citizens’ involvement in the decision making process through information technology (Manoharan & Holzer 2011). Voting systems are instruments for e-participation.

In the following, definitions of three democratic voting systems as used in this study are presented. Among them LD is described, which is within the focus of this study.

2.2.1 Democratic voting systems

The definitions are visualized by graphics with specific elements: Individuals (human-like figures) voting on proposals (circles labeled with 'A', 'B', 'C', and 'D') and/or on other individuals. Votes are visualized with arrows. The color saturation visualizes the individual's (potential) influence which may be higher through delegation (representative democracy) or copied votes (LD).

A *direct democracy* (Figure 2.1) is a voting system in which every individual directly vote for every proposal (Johann & Maalej 2015). Direct democracy empowers the individual to a large degree, but is not feasible. People do not have the time and knowledge to vote for every proposal (Green-Armytage 2015).

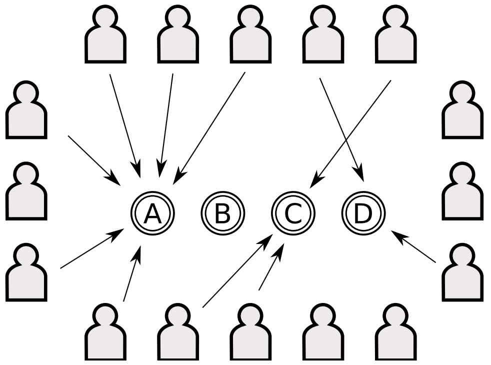

Figure 2.1: Direct democracy

A *representative democracy* (Figure 2.2) is a voting system in which every individual elects representatives for a given period of time and empower them to decide about proposals (Johann & Maalej 2015).

The representative democracy solves the time and knowledge problem of the direct democracy. However, it restricts the individual's influence. Hence, it is considered "democratic to a lesser degree" (Green-Armytage 2015).

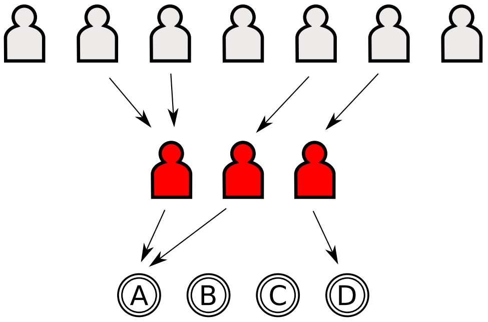

Figure 2.2: Representative democracy

A *liquid democracy* is a voting system in which an individual is able to vote directly on proposals, but simultaneously, he is able to copy votes of other individuals which are called *proxies* (Green-Armytage 2015). Proxies can copy votes from other individuals as well which results in ”delegate cascade” (Green-Armytage 2015).

In theory, individuals retain influence on decisions, while they are able to copy votes from others if they do not have the time and knowledge to make informed votes (Green-Armytage 2015). In the context of RE, this can also be relevant for communicating opinions of end-users to key stakeholders, such as developers(Johann & Maalej 2015).

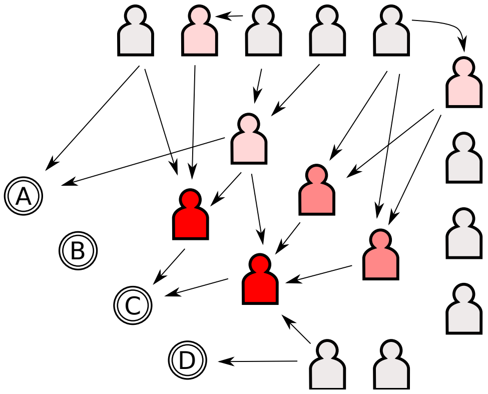

Figure 2.3: Liquid democracy

The LD concept is sometimes also called ”proxy voting system”, ”direct/proxy voting system” (Green-Armytage 2015), or ”delegated voting”[1].

Some political parties aim to change the representative democracy approach usually applied in current democracies by using e-democracy like the ”Pirate Party” existing in Germany, Austria, Italy, Switzerland and Brazil

[1] See the report of the degree project by Karin Ottesen ”Flexible representation by use of delegated voting - a case study of practical use”. `http://demoex.net/files/Ottesensuppsats.pdf`

(Green-Armytage 2015) or the recently established Dutch political party "GeenPeil"[2].

The Pirate Party mentioned above called their approach "liquid democracy" (Green-Armytage 2015) which was adopted by researchers (Johann & Maalej 2015). The term "liquid democracy" is chosen in this research, because it makes clear that it is a new voting system (a term like "proxy voting" or "delegated voting" could easily be understood as another term for representative democracy), but also because LD can be seen as a voting system in between two other voting systems of the democracy concept. This is comparable with liquid that is between two other states of aggregation, namely gas and solid matter. LD can be considered as a new democratic voting system that mixes properties of two known democratic voting systems.

Software engineers created liquid democracy tools, but these were never used for the purpose of RE. LD techniques allow the requirements gathering being based on a large crowd of users. The field of Large Scale Social Requirements Engineering aims communities to formulate requirements collaboratively. A tool that is made based on this philosophy is Requirements Bazaar (Renzel & Klamma 2014). It is a ORG tool that focuses on open source projects. The tool has implemented a voting system for proposals. However, it does not support the main LD property, and therefore misses the potential associated with the LD concept.

2.2.2 General LD properties

Voting systems are characterized by a set of properties. In this study, a *voting system property* is defined as a mechanism or policy for a voting system. Furthermore, a *LD property* is considered a voting system property associated with the LD concept.

In the following, LD properties suggested by existing literature are presented. The set of presented LD properties is not considered complete. No widely accepted definition of the LD concept could be found. Such a definition would be necessary to decide more confidently which voting system properties can be considered a LD property and which not.

Simultaneous direct and indirect voting means that individuals can vote directly for proposals *and* they can copy votes of others at the same time. Assume that user u_A voted directly for proposals p_1, p_2 and p_3. Additionally, he specified that he wants to copy all votes from person u_B. u_B voted for proposals p_2 and p_4. Consequently, u_A will adopt u_B's vote for p_4 and u_A's

[2] The political party was established on 5 December 2016. According to the mission statement on the organization's website, the party wants to "bring democracy back to the essentials: representation of the people with the will of majority". They want to conduct little referendums on which the political party's representatives base their decisions in the Dutch parliament. See `https://geenpeil.nl/over-geenpeil/` for more information.

vote for p_2 will be composed of his direct vote, but also of his copied vote. Simultaneous direct and indirect voting is considered the *main LD property* in this research.

Other properties are the possibility to adjust the votes anytime (Johann & Maalej 2015, Paulin 2014) which is called "*voting anytime*" and the possibility of a *collaborate discussion* between the participants (Johann & Maalej 2015). *Topic-based delegations* (Behrens et al. 2014), in which vote copies are restricted to specific domains could also be considered a separate LD property.

On the internet, you can also find a property that is very distinctive to the main LD property as presented above: copying votes by *vote recommendations*. Instead of copying votes directly from proxies, vote recommendations are reviewed first. The users are able to take the average of vote recommendations (in case the liquid democracy tool (**LDT**) supports multiple values for expressing acceptance or rejection) or to adjust the votes according to their interests[3]. This property may decrease scalability and therefore is not further discussed in this research report.

E-democracy tools that implement(ed) LD techniques are Adhocracy[4], Liquidizer [5], LiquidFeedback [6] and Votorola[7] (Johann & Maalej 2015).

2.3 Description of the used LDT

Each LDT uses a specific set of voting system properties (including properties not mentioned in this report). The properties used by the tool that was used during this research are described in this section.

The source code was written in a programming language that the researcher was not familiar with. Therefore, the properties of the used LDT are explained by situations. Each situation lists relevant input and output data. This way, it is impossible to provide an explanation that covers all properties used in the LDT. However, the provided examples were considered sufficient for a rough understanding of the tool's behavior.

At first, general information about the original LDT on which the used LDT was based on is provided. Then, the user interface is described. Furthermore, the properties of the used LDT are discussed in more detail. For this, terms and notations are defined. Then, the basic properties that were validated by all situations are listed. Additionally, specific situations are presented. Finally, all additional properties that were derived from the presented situations are summarized.

[3] See `http://campaigns.wikia.com/wiki/Liquid_Democracy`

[4] See `https://liqd.net/en/`

[5] See `http://liquidizer.github.io`

[6] See `http://liquidfeedback.org`

[7] See `http://reluk.ca/project/votorola/home.html`

2.3.1 About liquidizer

The LDT used in this study is mainly based on the LDT called ”liquidizer” made by Stefan Dirnstorfer (2012 version[8]).

Beside the main property of LD (delegate voting power to other users *and* vote on proposals simultaneously), the LDT provided the functionality to create new proposals, adjust the votes anytime, see the popularity of other users (based on their votes) and post a comment for each proposal.

In the following, the liquidizer is described in more detail. This information was retrieved from the ”wiki” of the German Pirate Party[9].

The liquidizer is an interactive voting game. It is a web application optimized for conciliation and finding compromises in a large group. The goal of liquidizer is not to accept or reject proposals, but to get a quick overview of opinions and conflicts. Political agreement would be a social process which could not be replaced by software according to liquidizer's philosophy. Liquidizer aims to be easy to use. Revocation of votes is possible anytime. This means that topics remain open. Decisions must be performed outside of the tool. Liquidizer gives every user a certain voting weight which is divided on all proposals a user votes on.

The mathematical concepts behind liquidizer are explained in a paper by Stefan Dirnstorfer[10].

2.3.2 Interface

In this section, a short description of interface of the used LDT is provided.

list of users: On this page, all users that expressed at least one vote are displayed. The user logged in can directly click on buttons for copying votes that are shown right next to each user name.

list of proposals: On this page, all proposals submitted by the researcher or other users are displayed. The user can sort the proposals to attributes like voting score, recent change, or time of submission

about proposal: On this page, information about a specific proposal is displayed: Users who voted for or against this proposal (including the vote value and the information whether the vote was copied by delegation), the time of creation of the proposal, and the author. Additionally, graphics are provided that display the development of the total voting score, the delegations associated with this proposal, and

[8] According to the GitHub repository of liquidizer, the last update was uploaded on 13 May 2012.

[9] See `http://wiki.piratenpartei.de/Liquidizer`.

[10] See `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=1679002`

a histogram showing how many individuals influence the selected proposal with their voting weight. Finally, the user is able to remove a proposal on this page if he created it.

about user: On this page, the user can see information about a specific user (or about himself). He can see which votes the user has given for which proposals. Additionally, he can see followers and role models of this user. Furthermore, the popularity (the similarity of voting behaviour compared to the voting behaviour of all other LDT users) is displayed. Additionally, the similarity of voting behaviour between the specific user and the user logged in can be seen. Finally, buttons are provided to enable the LDT user to copy votes of the selected user (in case that another user than the user logged in is selected).

help page: This page shows general information about the research and the tool.

my profile: This page enables the user to change the email address, the text displayed on the user's profile page, the password, or to delete all personal data (all votes, all comments, or the whole account which results also in the removal of the user's votes and comments).

after login: This page is the first page presented after the user logged into the system. On this page, only a link to the 'help page', the 'my profile' page, and a link redirecting to the list of proposals is presented.

overview: On this page, the following information is displayed: votes, followers, and role models of the LDT user who is logged in. Additionally, new submitted proposals and recently registered users are displayed. This page is displayed when a user clicks on the logo in the header of the web application. Therefore, it can be considered the homepage of the LDT.

In theory, anyone can register an unlimited number of accounts in the LDT.

Users can click on voting buttons. There are buttons for copying votes from a specific user with the values 0 (default), +1, +2, or +3. Additionally, the user can specify a preference for a proposal with the values -3, -2, -1, 0 (default), +1, +2, or +3. Unless the user does not create a new room (which is required once during the setup of the system), or click on such a voting button, his user account is not displayed to other users in the 'list of users' screen.

2.3.3 Terms and notations

In this section, a *situation* is defined as a combination of proxy preferences, proposal scores, voting weights and total voting weights.

A *proposal score* is the user's expression of acceptance respectively rejection for a specific proposal. It is defined by

$$S_{up} \in \{-3, -2, -1, X, 1, 2, 3\}$$

which means the proposal score of user u for proposal p. The value X means that the proposal score is not specified (the LDT does not distinct between proposals for which no proposal score is specified and items for which the user presses on the "zero" button. The symbol 'X' is chosen instead of 0 to improve readability and understanding of the situations that are presented in the following).

The vector S_u summarizes all proposal scores of a user u and is defined by

$$S_u = \begin{bmatrix} S_{up_1} & S_{up_2} & S_{up_3} & S_{up_4} & S_{up_5} & S_{up_6} \end{bmatrix}$$

Where p_1, p_2, p_3, p_4, p_5, and p_6 stand for the six proposals used in this section.

The *voting weight* is the user's contribution to the total voting weight of a proposal. It is calculated based on the proposal scores of users and proxies. It is defined by W_{up} ($W_{up} \in \mathbb{R} \mid W_{up} \geq -1$ and $W_{up} \leq 1$) which means the voting weight assigned by user u to proposal p.

The vector W_i summarizes all voting weights of a user i and is defined by

$$W_i = \begin{bmatrix} W_{ip_1} & W_{ip_2} & W_{ip_3} & W_{ip_4} & W_{ip_5} & W_{ip_6} \end{bmatrix}$$

Where p_1, p_2, p_3, p_4, p_5, and p_6 stand for the six proposals used in this section.

A *proxy preference* specifies how much a user's voting weight is influenced by all votes from another user (called a proxy, see section 2.2.1). It is defined by $P_{ij} \in \{0, 1, 2, 3\}$ which expresses the strength of influence of all votes from user j for the voting weights of user i. Proposals which corresponding voting weights are influenced by a proxy are called "copied votes". Those that are not influenced by proxies are called "direct votes". All voting weights that are influenced by a copied vote are **underlined and printed in a bold lettertype** in subsection 2.3.5.

The *total voting weight* T_p is the sum of voting weights of all users for proposal p and is defined by

$$T_p = \sum_{i=1}^{n} W_{u_i p} \qquad (T_p \in \mathbb{R})$$

Where u_x ($x \in \mathbb{N}$) stands for a user and n ($n \in \mathbb{N}$) stands for the total number of users.

The vector T summarizes all total voting weights for each proposal and is defined by

$$T = \begin{bmatrix} T_{p_i} & T_{p_{i+1}} & \dots & T_{p_n} \end{bmatrix}$$

Where p_x ($x \in \mathbb{N}$) stands for a proposal and n ($n \in \mathbb{N}$) stands for the total number of proposals.

2.3.4 Basic properties

Basic properties that were confirmed by each situation simulated in the tool are described below.

1. The sum of squares of all individual voting weights is always equal to 1 ($\sum_{i=1}^{n}(W_{up_i})^2 = 1$; where p_x ($x \in \mathbb{N}$) stands for a proposal, n ($n \in \mathbb{N}$) is the total number of proposals existing in the system, and u is any user).

2. Each user may vote on an unlimited number of proposals.

3. Each user may change his direct votes anytime.

4. Each user may change proxy preferences anytime.

5. A *daily vote decay* can be defined for each discussion room. If the user does not log in within 24 hours, the individual voting weights are decreased by a certain value (default is 0.01). This decrease accumulates for each day the user does not log in. As soon as a user logs in, the decrease by the daily vote decay will be set to zero again for that user.

2.3.5 Situations

The situations were composed of two users and six proposals. The properties are based on situations that were simulated in the LDT with a clean database. The daily vote decay was set to zero. Situation #5 was removed after the data for this section had been collected. The identifiers were not changed.

$$P_{AB} = 0 \qquad P_{BA} = 0$$

$$S_A = \begin{bmatrix} 3 & X & X & X & X & X \end{bmatrix} \qquad W_A = \begin{bmatrix} 1.00 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$S_B = \begin{bmatrix} X & 2 & X & X & X & X \end{bmatrix} \qquad W_B = \begin{bmatrix} 0 & 1.00 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 1.00 & 1.00 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2.4: Situation #1: Scores for single proposals

Change: None.

Observation: User A votes with a proposal score of 3 on p_1. User B votes with a proposal score of 2 on p_2. This results in a full voting weight (1.00) for p_1 and p_2.

$$P_{AB} = 0 \qquad P_{BA} = 0$$

$$S_A = \begin{bmatrix} 3 & 2 & 1 & X & -1 & X \end{bmatrix} \qquad W_A = \begin{bmatrix} 0.78 & 0.52 & 0.26 & -0.26 & 0 & 0 \end{bmatrix}$$

$$S_B = \begin{bmatrix} 2 & 2 & X & X & X & X \end{bmatrix} \qquad W_B = \begin{bmatrix} 0.71 & 0.71 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T = \begin{bmatrix} 1.48 & 1.22 & 0.26 & -0.26 & 0 & 0 \end{bmatrix}$$

Figure 2.5: Situation #2: Votes on multiple proposals

Change: Each user votes now on multiple proposals.

Observation: The voting weight is distributed on the different proposals. The higher the proposal score, the higher the voting weight.

$$P_{AB} = 3 \qquad P_{BA} = 0$$

$$S_A = \begin{bmatrix} 3 & 2 & 1 & -1 & X & X \end{bmatrix} \qquad W_A = \begin{bmatrix} 0.61 & 0.41 & 0.21 & -0.20 & 0 & \underline{\mathbf{0.61}} \end{bmatrix}$$

$$S_B = \begin{bmatrix} X & X & X & X & X & 3 \end{bmatrix} \qquad W_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.61 & 0.41 & 0.21 & -0.20 & 0 & 1.61 \end{bmatrix}$$

Figure 2.6: Situation #3: Single copied vote with high proxy preference

Change: User A defines user B as one of his proxies with the maximal proxy preference of 3.

Observation: The voting weight of the single copied vote is equal to the voting weight of a direct vote with proposal score 3 ($S_{Ap_1} = S_{Bp_6} \wedge W_{Ap_1} = W_{Ap_6}$).

$$
\begin{aligned}
&P_{AB} = 2 & &P_{BA} = 0 \\
S_A &= \begin{bmatrix} 3 & 2 & 1 & -1 & X & X \end{bmatrix} & W_A &= \begin{bmatrix} 0.69 & 0.46 & 0.23 & -0.23 & 0 & \underline{\mathbf{0.46}} \end{bmatrix} \\
S_B &= \begin{bmatrix} X & X & X & X & X & 3 \end{bmatrix} & W_B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \\
& & T &= \begin{bmatrix} 0.69 & 0.46 & 0.23 & -0.23 & 0 & 1.46 \end{bmatrix}
\end{aligned}
$$

Figure 2.7: Situation #4: Single copied vote with low proxy preference

Change: Similar to the previous situation, but now with a lower proxy preference ($P_{AB} = 2$).

Observation: The voting weight of the copied vote is equal to the voting weight of a direct vote with proposal score 2 ($S_{Ap_2} = S_{Bp_6} \wedge W_{Ap_1} = W_{Ap_6}$).

$$
\begin{aligned}
&P_{AB} = 3 & &P_{BA} = 0 \\
S_A &= \begin{bmatrix} 3 & 2 & 1 & -1 & X & X \end{bmatrix} & W_A &= \begin{bmatrix} 0.61 & 0.41 & 0.21 & -0.20 & \underline{\mathbf{0.34}} & \underline{\mathbf{0.51}} \end{bmatrix} \\
S_B &= \begin{bmatrix} X & X & X & X & 2 & 3 \end{bmatrix} & W_B &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0.56 & 0.83 \end{bmatrix} \\
& & T &= \begin{bmatrix} 0.61 & 0.41 & 0.21 & -0.20 & 0.9 & 1.34 \end{bmatrix}
\end{aligned}
$$

Figure 2.8: Situation #6: Multiple copied votes

Change: Similar to situation #3, but now B also votes on p_5.

Observation: The voting weight of multiple copied votes are not equal to the voting weight of a direct vote with the same proposal score ($S_{Ap_1} = S_{Bp_6} \wedge W_{Ap_1} \neq W_{Ap_6}$).

$$P_{AB} = 3 \qquad P_{BA} = 0$$

$$S_A = \begin{bmatrix} 3 & 2 & 1 & -1 & X & 3 \end{bmatrix} \qquad W_A = \begin{bmatrix} 0.78 & 0.52 & 0.26 & -0.26 & 0 & \underline{\mathbf{0}} \end{bmatrix}$$

$$S_B = \begin{bmatrix} X & X & X & X & X & -3 \end{bmatrix} \qquad W_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1.00 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.78 & 0.52 & 0.26 & -0.26 & 0 & -1.00 \end{bmatrix}$$

Figure 2.9: Situation #7: Differing direct and copied vote (1/3)

Change: Similar to situation #3, but now A votes on p_1 while he still copies all votes from B.

Observation: The conflicting proposal is not listed on the user page of the LDT. However, the user is able to see on other webpages produced by the LDT that a direct vote conflicts with a copied vote.
The conflicting proposal results in a voting weight which is equivalent to not voting at all ($S_{Ap_5} = X \wedge S_{Ap_5} \neq S_{Ap_6} \wedge W_{Ap_5} = W_{Ap_6}$).

$$P_{AB} = 3 \qquad P_{BA} = 0$$

$$S_A = \begin{bmatrix} 3 & 2 & 1 & -1 & X & 2 \end{bmatrix} \qquad W_A = \begin{bmatrix} 0.75 & 0.50 & 0.25 & -0.25 & 0 & \underline{\mathbf{-0.25}} \end{bmatrix}$$

$$S_B = \begin{bmatrix} X & X & X & X & X & -3 \end{bmatrix} \qquad W_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1.00 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.75 & 0.50 & 0.25 & -0.25 & 0 & -1.25 \end{bmatrix}$$

Figure 2.10: Situation #8: Differing direct and copied vote (2/3)

Change: Similar to the previous situation, but now S_{Ap_6} is 2.

Observation: The difference between absolute proposal scores of the conflicting votes is not equal to zero as in the previous situation ($|S_{Ap_6}| - |S_{Bp_6}| \neq 0$). Although A submits a positive vote ($S_{Ap_6} = 2$), A's voting weight is negative. Proxy B specified a much lower proposal score ($S_{Bp_6} = -3$) than A ($S_{Ap_6} = 2$).

$$P_{AB} = 3 \qquad P_{BA} = 0$$

$$S_A = \begin{bmatrix} 3 & 2 & 1 & -1 & X & 2 \end{bmatrix} \quad W_A = \begin{bmatrix} 0.48 & 0.32 & 0.16 & -0.16 & 0 & \underline{\mathbf{0.79}} \end{bmatrix}$$

$$S_B = \begin{bmatrix} X & X & X & X & X & 3 \end{bmatrix} \quad W_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1.00 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.48 & 0.32 & 0.16 & -0.16 & 0 & 1.79 \end{bmatrix}$$

Figure 2.11: Situation #9: Differing direct and copied vote (3/3)

Change: Similar to the previous situation, but now S_{Bp_6} is 3.

Observation: A's voting weight on p_6 is higher than the voting weight on p_1, although he specified the highest possible proposal score for p_1. In this situation, the voting weight on p_6 seems to be amplified by B's (even more) positive vote ($W_{Ap_6} > W_{Ap_1}$, while $S_{Ap_6} < S_{Ap_1}$). A combination of a direct vote and a copied vote on the same proposal may result in a higher voting weight which would not be possible by a direct or copied vote on its own.

2.3.6 Additional properties

Below, a table of properties (Table 2.1) is depicted which summarizes crucial properties that were validated by specific situations presented in subsection 2.3.5.

ID	**Property**	Situation
1	If a user submits only one vote, the vote results in a full voting weight independent of the proposal score.	#1
2	If a user votes on multiple proposals with different proposal scores, this results in different voting weight according to the proposal score.	#2
3	A copied vote does not always result in a voting weight equal to the voting weight of a direct vote with the same proposal score as the copied vote from the proxy.	#6
4	A combination of a direct vote and a copied vote results in an voting weight that takes the user's *and* the proxy's proposal score into account. This may result in a higher (see situation #9), neutral (equal to zero/not voting; see situation #7) or even a negative voting weight (see situation #8).	#7, #8, #9

Table 2.1: Additional properties of the used LDT

Chapter 3

Setup & Method

In this section, general information about the research is given. This is followed by more detailed information on the activities during this research such as the first round interviews with the CC, using the LDT for RE, the second round interviews with the CC, and the data analysis.

3.1 General approach

The following research question was stated: How can LD techniques contribute to the RE processes of existing software applications? This question was split into four sub questions as follows:

1. What are the effects of using LD techniques on the requirements understanding of software developers/designers and users?
2. What are potentials and areas of concern when applying LD techniques for gathering end-user requirements of an existing web application?
3. Do end-users feel more motivated to participate in the RE processes by LD techniques?
4. How can findings be used to contribute to the RE processes of the CC?

Data was gathered and linked to theoretical variables. These are presented below. Additionally, an overview of all main data gathering activities is provided.

3.1.1 Theoretical variables

For answering above research questions data was gathered which was mapped to the following theoretical variables:

- **Requirements understanding**: The set of requirements and their prioritization known to the requirements practitioners of a software application.

- **Motivational factors**: The LD techniques that were used in the selected LDT of this research. It was investigated whether these techniques could be considered motivational incentives for end-users to participate in ORG.
- **RE approaches of CC**: The techniques used by the CC and the involvement of end-users during RE activities. This was necessary information for an advice of using LD techniques for RE processes that suits to the CC.
- **Potentials and areas of concern**: Findings concerning LD for RE, or ORG in general. These findings are based on interviews with requirements practitioners of the SA and the CC, and on the case study of using the LDT. Findings that are associated with added value by using LD techniques for RE were considered '*potentials*'. Findings that reveal risks for such an added value were considered '*areas of concern*'.

The conceptual model of this research is depicted in Figure 3.1. It shows that all findings linked to the theoretical variables affect the conclusion about the contribution of LD to RE processes.

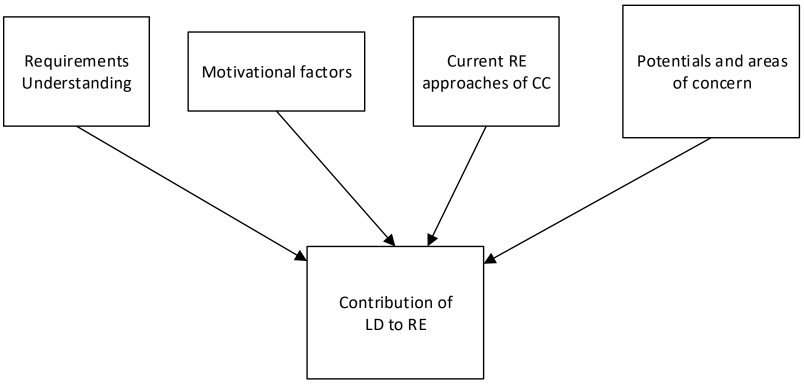
Requirements Understanding
Motivational factors
Current RE approaches of CC
Potentials and areas of concern
Contribution of LD to RE

Figure 3.1: Conceptual model

3.1.2 Research activities

General RE practices of the CC were determined by conducting interviews with five requirements practitioners of three different business units of the CC (the CC is divided in 15 different business units in total).

A LDT was set up in order to gather end-user requirements for a SA that the CC uses as internal software application. It was determined whether the results added value to the requirements understanding of the requirements practitioners of the SA and the end-users who participated in the LDT.

Also, it was investigated whether LD techniques can be seen as motivational incentives for end-users to participate in requirements activities. Two interviews were conducted with the requirements practitioners of the SA. One interview before and one interview after using the LDT.

Based on the findings of the LDT case study, a second round of interviews with requirements practitioners of the CC was conducted to investigate potentials and areas of concern of using a LDT for RE purposes of the CC.

All interviews were conducted in the Dutch language. Also, the LDT interface was translated to the Dutch language.

The process diagram of the research that shows the order of the main data gathering activities is depicted below (Figure 3.2).

Finally, findings of the LDT case study and the interviews with requirements practitioners were analyzed.

"*End-users*" are individuals directly interacting with a software solution. The term "*LDT user*" is used for every person who participated in the LDT during the case study.

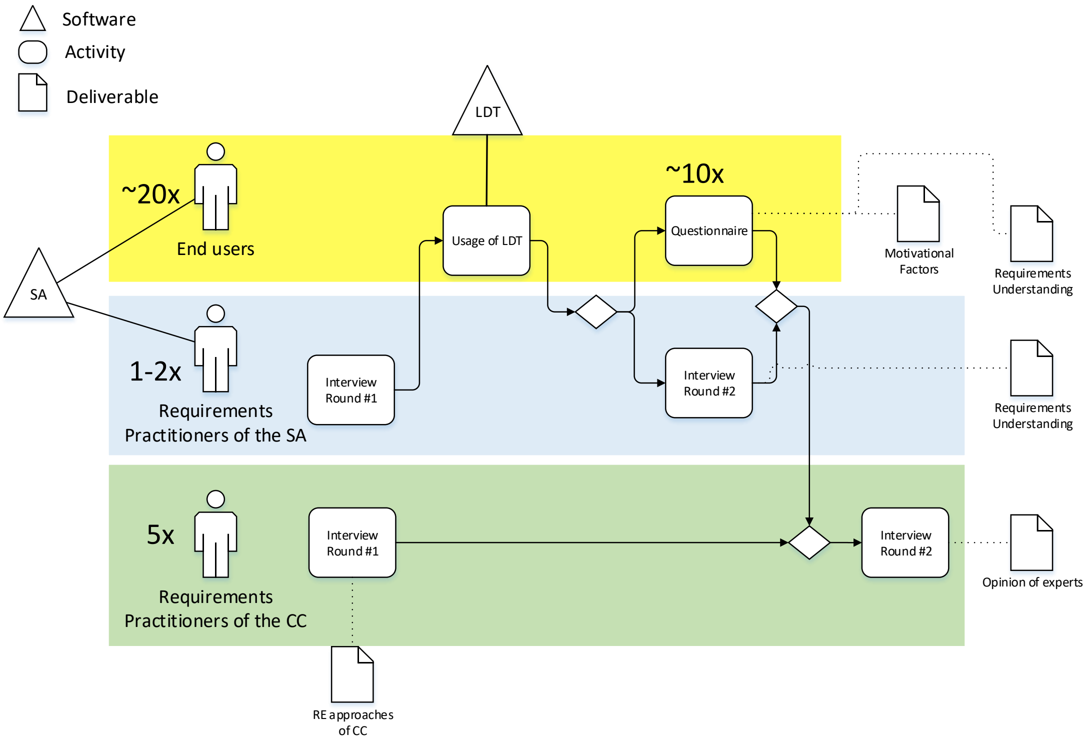

Figure 3.2: Process diagram of the main data gathering activities during this research

3.2 First round of interviews with CC

Requirements practitioners of the CC were interviewed for the first time to gather information about the current RE practices of the CC. The first round interviews focused on the current RE approaches the interviewee's business unit conducts, and which RE approaches the interviewee in particular conducts. The first round of interviews was conducted before using the LDT to ensure that there was enough time to prepare the second round interviews (see section 3.4). The questions were sent by mail to the interviewees with the request to read and think about the questions themselves before the interview started. The interviews were recorded by an audio recording device. Transcriptions of the audio recordings were made. These transcriptions were sent to the interviewees by mail with the request to check and edit them if necessary.

Below, the questions asked to all of the requirements practitioners are listed.

1. What is your definition of ”requirements for software applications”?
2. How does your business unit generally handle requirements?
3. How do you elicit requirements?
4. How do you evaluate requirements?
5. How do you document requirements?
6. What do you do for quality assurance of requirements?
7. What do you do to handle changes of requirements?
8. How high do you estimate the end-user satisfaction of projects you have been working on (in a scale from 1 (very unsatisfied) to 10 (very satisfied)?
9. Are there RE techniques your colleagues of your business unit use but you don't? In case of yes, which ones?
10. What are situations in which you involved end-users for gathering requirements?
11. What are situations in which you missed the input of end-users?
12. Which problems do you encounter when end-users say something about requirements?

3.3 Using a LDT for RE

One of the LDTs described in section 2.2.2 was selected. A software application (*"SA"*) was selected for which the LDT gathers requirements about.

The LDT has been installed on a server, so it could be easily accessed by the CC employees who use the SA.

Around 120 employees of a certain business unit (business unit C) were invited to participate in the LDT by mail. On the fifth day, 10 employees of other business units were invited for participation face-to-face by the researcher. The LDT was available for nine days. In the morning of the eighth day, all participants who gave input in the LDT until that time were requested via a mail to fill in an online questionnaire about using the LDT. On the seventh day, six additional employees of arbitrary business units were invited face-to-face to participate in the LDT and to fill in the questionnaire afterwards immediately. These face-to-face invitations were considered necessary to increase the number of respondents.

The questionnaire was an online questionnaire which focused on the end-user's changes of requirements understanding after using the LDT, the end-user's motivation to participate in the requirements gathering process and on issues that end-users might encounter during their usage of the LDT.

In the following, more specific information about the selection of the LDT, adjustments on the LDT for this case study, the questionnaire, the selection of the SA, and the interviews with the requirements practitioners of the SA is provided.

3.3.1 The used LDT

The following criteria were determined for the selection of a LDT:

- The tool implemented the main LD property (delegate voting power to other users *and* vote on proposals simultaneously)(see section 2.2).
- The source code of the LDT was publicly available and it's license allowed adjustments.
- The LDT could be easily installed and run on a standard server environment.
- The source code was documented.
- The interface was perceived as simple to use.

The only LDT of the four LDTs mentioned in section 2.2.2 that matched to these criteria was the web application "liquidizer" (see subsection 2.3.1). Therefore, this application was chosen as the code basis for the LDT applied in this research.

3.3.2 Adjustments on LDT

The source code of "liquidizer" was almost completely reused and slightly changed. The LDT was tried to be made as accessible as possible with the available resources of this research.

Early versions of the LDT were tested by some employees of the CC. New insights of these tests resulted in further adjustments of the tool.

Below, the final adjustments to the original "liquidizer" web application are listed.

- Changes to the look-and-feel of the interface to make the LDT more accessible to the users: The name of the web application was changed to "RanX"[1]. All colors were adjusted to a new color scheme. The original logo was replaced by a logo which included the new name and which was designed according to the new color scheme.
- The possibility to create new rooms or switch to another room was removed because the LDT should only gather information about one topic (namely the SA). For that, one default room was sufficient.
- The interface was completely translated to the Dutch language to make the LDT more accessible to the users. The option to switch to other languages was removed.
- The homepage was filled with basic information about the tool, the purpose of the tool (which should clarify that the tool is supposed to gather wishes about the SA of end-users) and a link to the help page for users that are interested in more detailed information about the tool and the research.
- The content of the help page was edited.
- Buttons with dead links were removed.
- The input of an email address in the corporate domain of the CC was made mandatory on the web page to register a new account. That way personal data such as age or sex could easily be looked up in the intranet of the CC and the request of the questionnaire could be send to the specified email address.

The term 'requirement' was avoided in the interface of the LDT, because it could suggest that end-users could expect that their (votes on) proposals would really be implemented. Instead, all the pages included the clause

[1] The name "RanX" was chosen to suggest that end-users work collaboratively on a "ranking" that gives an indication for what is important for the end-users. The X was supplemented to suggest that the result is unknown and needs to be determined (comparable with the variable name 'x' that is often used in Algebra).

"wishes for the [SA]" (while the LDT displayed the real name of the software application instead of [SA]).

In some tests, before the LDT was applied for data gathering purposes it had been observed that people hesitated to formulate new wishes. The initial set of requirements as described in subsection 3.3.5 had been submitted in the LDT by the researcher before the first end-users of the SA registered their account in the LDT. 17 of the 26 requirements gathered from the SA were translated in user stories of the format "as a <role> I want <functionality> in order to <goal>". This initial set of requirements was provided to facilitate quick voting and to inspire users for formulating new ones by themselves.

3.3.3 Questionnaire

A questionnaire was designed to measure requirements understanding, motivational factors, potentials and areas of concern.

The questionnaire contained ten items in total. The first three questions were concerned with requirements understanding. The five questions that followed were concerned with motivational factors. The last two questions were open and gathered information about potentials and areas of concern.

For the closed-ended questions, a five-points likert-scale ranging from disagree to agree was used. The questionnaire should provide a rough indication for the motivational capabilities of LD techniques in ORG. Five categories were considered sufficient for such a rough indication. An alternative would have been e.g. ten categories. However, respondents would need more time to fill in the questionnaire and the difference between 80% of agreement or 90% of agreement was considered nonrelevant.

The complete online questionnaire provided to the users of the LDT is shown below (see Table 3.1).

Position in questionnaire	**Item**	**Value range**
1	I am aware of more wishes for [SA] after using RanX.	1-5
2	I think that particular wishes for [SA] are more important after using RanX.	1-5
3	I think that particular wishes for [SA] are less important after using RanX.	1-5
4	I participate in RanX because someone asked me to do so.	1-5
5	I participate in RanX because I would like to win a [CC] labeled plastic water bottle	1-5
6	I participate in RanX because others can vote on wishes I specified	1-5
7	I participate in RanX because colleagues can delegate their voting power to me	1-5
8	I participate in RanX because I can adjust my votes anytime	1-5
9	Was there anything unclear during the usage of RanX?	text
10	Do you have questions and/or comments about RanX, this questionnaire or the research in general?	text

Table 3.1: Complete online questionnaire for users of the LDT

3.3.4 Selection of the SA

The selection of the SA was based on the following criteria:

- The application is frequently used by more than 400 users (used at least 4 times a year). These bottom limits were considered a necessary precondition for gathering data that could confirm or refuse the LD potential of scalability.
- The developer of the SA has interest in the wishes of end-users for the

SA. Otherwise the results of the LDT would be worthless from the start.

- It is possible to interview at least one requirements practitioner of the SA. This was important for measuring the change in requirements understanding through the LDT.
- End-users of the SA can be easily invited face-to-face to participate in the LDT in case not enough end-users would participate after they had been invited by mail.

Several applications (especially software applications that were used internally in the CC, because of the last criterion in the list) were taken into consideration. Finally, a software application was chosen which gathers continuous feedback about the work performance of employees digitally from colleagues, managers, or clients. It is an application which is not only used by employees of the CC, but also by employees of several other companies and by freelancers. However, all employees of the CC were supposed to use the tool as evidence for performance reviews between employees and their managers. That means that the SA was used internally in the CC by around 2400 people[2]. One of the founders of the SA stated that end-users are considered "most important". The SA was developed by a team of ten members[3]. Two of them were frequently concerned with the requirements of the SA and were also available for research interviews. They were entitled "*requirements practitioners of the SA*".

Other characteristics of the SA are the following: The SA is available as web application but also as app on mobile devices, closed groups of users (e.g. for internal use by companies like the CC) are administered by operators, and users interact with the system in the roles of feedback requester or feedback giver.

3.3.5 First interview with requirements practitioners of the SA

One of the two requirements practitioners of the SA had been interviewed before the LDT was applied. The respondent was asked to prepare for the interview by reading the questions sent by mail. He was asked about common practices within his development team and the team's current requirements understanding for the SA.

Current requirements of the SA were gathered during the first interview. The interviewee provided a list with current requirements by email that supplemented the requirements gathered during the interview. From these sources of information an initial set of requirements was created.

[2] According to the annual report of the CC in 2015.

[3] According to the website of the SA.

3.3.6 Second interview with requirements practitioners of the SA

After using the LDT, a second interview with requirements practitioners of the SA was conducted. In this interview, the feedback given by LDT users was discussed with the requirements practitioners. Questions were associated with requirements understanding, potentials, and areas of concern.

3.4 Second round of interviews with CC

Requirements practitioners of the CC were interviewed for a second time to investigate potentials and areas of concern of using a LDT for RE purposes of the CC. In the second round of interviews, the results of using the LDT were recapitulated and the requirements practitioners were asked about the intention to use a LDT in the company/business unit. These interviews were conducted after the LDT had been used to provide the interviewees with the findings of that part of this research.

The findings of using the LDT (including the comments of end-users and the requirements practitioners of the SA) were delivered to the requirements practitioners of the CC together with the questions for the second interview via mail.

Interviewees were asked to read and think about that information and the questions before the interview started. The interviews were recorded by an audio recording device. Also, for these interviews transcriptions of the audio recordings were made. These transcriptions were sent to the participants by mail with the request to check them and edit them if necessary.

The following questions were asked in each interview:

1. Could you use a tool such as RanX in your work? If yes, how? If no, why not?

2. Would a tool such as RanX maybe add value for other projects within your business unit that you have heard of?

Additionally, two or three questions were asked which were based on a interviewee's statement made in the first round interview, and which were related to the topic of ORG.

3.5 Data analysis

In this research, data originated mainly from the following sources: the interviews with requirements practitioners of the CC (two rounds, in total ten interviews) and of the SA (two rounds, in total two interviews), the database of the LDT which was populated by the input of the participating end-users, and the questionnaires filled in by end-users.

Some people gave information via mails or during unplanned talks that was considered relevant according to the scope of this research. This information was included in this report and marked as such in the results section.

For analysing the potentials and areas of concern, a list with findings was created. Quotations from respondents were associated with these findings. All findings were clustered for readability.

Further information that is relevant for the data analysis of this research is provided in this section.

3.5.1 Transcriptions

All spoken language during the interviews was transcribed. Every word was literally transcribed except in the following cases:

- utterances had a bad block format which were difficult to read
- utterances could not be understood acoustically
- utterances did not contribute to the understanding of the interviewee's statements
- utterances were considered nonrelevant according to the scope of the research (see subsection 3.1.1). These were summarized in rectangular brackets ([and])

Below, further symbols used in the transcriptions are explained.

> Two rectangular brackets [[and]]: Short utterances by respondent or researcher. It could also be a substitution of terms that are secret for the public.
> Normal parentheses (and): Replacements based on the researcher's assumptions about what the speaker meant by the particular sentence to improve readability and quick understanding.
>
> Dots within parentheses (...): utterances which could be ignored without losing the ability to understand what the speaker meant to improve readability.
>
> Dots without parentheses ...: an utterance that was interrupted by another utterance (either by another speaker or by the speaker himself).

3.5.2 Coding

The transcriptions of the interviews were analyzed and marked with codes that direct to the theoretical variables of this research (see subsection 3.1.1). Main codes with their explanations are listed below.

potential: Statements that are concerned with the added value of a LDT for RE, a reason why a LDT would add value, a reason why a requirements practitioner would use it or an use case for a LDT in RE.

areasOfConcern: Important areas of concern when applying a LDT for RE, a reason why a LDT would not add value, a reason why a requirements practitioner would not use it or characteristics of situations in which ORG with end-users should not be applied.

intentionToUse: Statements of requirements practitioners of the CC to the question whether he would use a LDT in his projects.

Chapter 4

Results

This chapter shows all relevant data gathered during this study. At first, information about the respondents and their context is provided. The sections that follow are structured according to the theoretical variables (see subsection 3.1.1): Requirements understanding, RE approaches within the CC, and potentials and areas of concern.

4.1 Respondents and their context

In this section, information about the respondents is provided. First, the requirements practitioners of the CC are described, then the requirements practitioners of the SA, and finally, the end-users who participated in the LDT.

4.1.1 Requirements practitioners of CC

Some basic information about the requirements practitioners of the CC are depicted in the table below (Table 4.1).

Respondent	Sex	Age	Official function	Business Unit
R1	M	41	Senior Business Analyst	Business Unit A
R2	M	23	Senior Software Engineer	Business Unit B
R3	M	61	Lead Business Analyst	Business Unit A
R4	M	46	Management Consultant	Business Unit B
R6	M	54	Senior Business Intelligence Specialist	Business Unit C

Table 4.1: Data about requirements practitioners of the CC

The five interviewed requirements practitioners worked in three different business units. In the following sections basic information about the three business units and the requirements practitioners is provided.

Business units

The names of the business units have been replaced by letters to avoid disclosing confident information.

Business unit A was dedicated to implement new IT technology in large enterprises. Important aspects were business processes and the integration of new systems with existing systems.

Business unit B was concerned with designing and building applications for mobile devices.

Business unit C dealt with projects in the area of big data and business analytics. The professionals store, structure, analyze, distribute and visualize information so it can be used for better decision making in businesses.

Contextual information about the requirements practitioners

In the past, R1 elicited requirements. At the time of the first interview he said that he instructs others in the field of requirements management for five years. R1 is currently responsible for modeling business processes. He got requirements as input for his modeling activities.

R1 defined a requirement as ”wishes and demands” by all stakeholders. He defined a stakeholder as someone who is affected by the system to be built. However, he did not see software developers as stakeholders for requirements.

R2 is lead developer for an app on iOS. He is also scrum master within the development team.

R2 defined requirements as wishes and needs of stakeholders. End-users would be important stakeholders, but also software vendors of external services which his project needs to interact with.

R3 formally works as Lead Business Analyst in a bank. He said that his function would be ”Senior Managing Consultant”/”Business Consultant”. He underlined that this description is vague and that his job is not limited to specific tasks. In general, he works in very large projects and is concerned with tasks in business analysis, testing, and requirements management. He also creates contracts and helps in defining which work needs to be done. He formulates high-level requirements, but also specifies how interfaces should be implemented. Currently, he works on a new payment system for web shops. In the past, R3 worked on processes of delivering bank cards to clients, on contactless payment, fraud detection, and online payment systems.

R3 defined requirements as written needs for a solution by stakeholders. He underlined that software is a possible part, but not always necessary part of a solution.

R4 formally works as Management Consultant in the company, but he defined his function as "innovator". He focuses on design, mobile and Internet of Things. He prefers to talk about people and their behavior instead of talking about technology. He is committed to functional requirements and often is not concerned with non-functional requirements.

R4 defines requirements as "wishes and demands that are influential". They are point of references for what is valuable for end-users. Sometimes they are more technical than functional and sometimes they are non-functional in terms of e.g. performance. He also mentioned that requirements can be related to the environment of the system to be built, or to business processes.

R6 formally has the function of Senior Business Intelligence Specialist. Currently, he works on a project concerning the setup of a new data warehouse. At that project he works as Lead Business Analyst and Lead Functional Designer.

R6 defined requirements as needs "of the client for a specific system or a specific functionality". The needs of other stakeholders are included in the needs of the client according to R6.

4.1.2 Requirements practitioners of SA

Two requirements practitioners of the SA were interviewed: R5 and R7. The first interview was conducted only with R5 via a Skype meeting. The second interview was conducted with R5 and R7 face-to-face at the SA's place of residence.

R5 is 26 years old and works as operation manager. He is responsible for the support of clients and end-users. Besides, he is concerned with the marketing activities of the SA.

R7 is 24 years old and works as UX designer.

4.1.3 LDT users

In total, 25 end-users of the SA participated in the LDT. They all were employees within the CC.

In the table below (Table 4.2), the age, sex, the number of submitted wishes, the number of wishes the user voted on, the number of submitted comments, and the duration of voting behaviour of each user is depicted. The average values are depicted in the last row of the table.

The data about sex and age were obtained by searching for the corporate email address of the respondent within the intranet of the CC.

Respondent	**Age**	**Sex**	**#submitted wishes**	**#wishes the user voted on**	**Minutes between first and last voting activity**	**#comments**
R8	46	m	1	13	9,90	0
R9	45	m	1	16	3,75	0
R10	53	m		8	4,52	0
R11	23	m	1	10	2,97	0
R12	43	m		18	2,05	0
R13	27	m	1	19	10,78	0
R14	23	m		1	2,08	0
R15	33	m	1	16	5,43	0
R16	48	m	1	22	5,20	0
R17	58	m	1	19	6,57	0
R18	48	m		8	2,77	0
R19	35	m		21	17,63	0
R20	51	m	1	18	35,20	0
R21	28	m	1	22	6,57	0
R22	23	m	2	5	2,77	0
R23	24	m		12	4,33	0
R24	unknown	m		6	5,03	0
R25	26	m		11	6,78	0
R26	42	m		1	1,13	0
R27	27	m		22	6,07	0
R28	27	f		10	30,43	0
R29	unknown	m		3	3,10	0
R30	26	m		21	6,32	0
R31	unknown	f		27	6,4	0
R32	31	m		6	4,42	0
	36			13,4	7,82 00:07:49	

Table 4.2: Data about LDT users and their LDT usage behaviour

4.2 Requirements understanding

In Table 4.3, the following data about the submitted requirements in the LDT is presented: An identifier (column 1), who submitted the requirement (that can be either the researcher or a LDT user; column 2), whether the priority of the requirement was considered very important by the SA (column 3), whether the requirements were considered known (yes) or unknown (no) for the SA before using the LDT (column 4), the total voting weight of the requirement (column 5), the number of positive votes (column 6), and the number of negative votes (column 7).

The list is sorted by the total voting weight in descending order.

The identifier corresponds to the identifier in the database of the LDT. It also corresponds with the order of submission. The first three items submitted to the LDT contained errors and had been removed before the first LDT users registered. Hence, the first requirement's identifier was four.

During the setup of the LDT, the daily vote decay (see subsection 2.3.4) accidentally was not changed to zero and was set automatically to the default value of 0.01. Every LDT user logged only once into the system. Therefore, the total voting scores were influenced by the daily vote decay. For the data analysis, the total voting scores were used as shown in the LDT 13 days after the first voting activity by a LDT user. This could have altered the position of requirements 4 and 8. Without daily vote decay, they would possibly have even higher total voting scores. The effect of the daily vote decay on the other requirements was considered nonrelevant.

Table 4.4 and Table 4.5 show almost all requirements submitted to the LDT. One submit was considered non-serious and was left out (item #21). The requirements were formulated in Dutch. The tables contain the translated items.

1	**2**	**3**	**4**	**5**	**6**	**7**
ID	**By**	**Prio**	**K?**	**Tvs**	**# +**	**# -**
25	LDT user		No	5.1	15	0
04	Researcher	X		4.16	19	4
08	Researcher	X		3.06	14	3
27	LDT user		Yes	2.6	11	1
05	Researcher	X		2.53	15	0
20	Researcher			2.35	12	2
15	Researcher			2.28	14	5
24	LDT user	X	Yes	1.58	08	5
26	LDT user		No	1.29	06	3
18	Researcher			1.26	10	5
30	LDT user		Yes	1.21	04	0
22	LDT user		No	0.95	08	4
13	Researcher			0.77	09	4
23	LDT user		Yes	0.53	06	1
31	LDT user		No	0.39	02	0
28	LDT user		No	-0.09	03	3
10	Researcher			-0.41	04	7
29	LDT user		No	-0.41	0	3
17	Researcher			-1.17	02	9
09	Researcher			-1.35	05	10
14	Researcher			-1.37	05	11
07	Researcher			-1.41	04	10
16	Researcher			-2.05	01	10
11	Researcher			-2.29	02	11
12	Researcher			-2.45	03	13
19	Researcher	X		-3.14	01	13
06	Researcher	X		-3.58	01	13

Table 4.3: Data about submitted requirements in the LDT

ID	Requirement
4	As feedback requester, I want to define questions on my own in order to have feedback that is more relevant for me.
5	As a feedback giver, I want to determine goals for a specific specification in a questionnaire and receive feedback about this.
6	As external person, I want to look up the feedback of a specific [SA]-user.
7	As a feedback requester, I want to generate a link for requesting feedback in order to show this link during presentations.
8	As feedback requester, I can make reports in order to gain insight into my progress during a longer period.
9	As a feedback giver, I see a photo of the person for whom I give feedback in the feedback request mail.
10	As a feedback requester, I want to reuse mail addresses of feedback givers that I specified earlier so that I don't need to remember them.
11	As feedback requester, I want to receive a mail weekly in order to gain insight into my progress.
12	As feedback requester, I want to receive a mail monthly in order to gain insight into my progress.
13	As a feedback requester, I want to be able to change my password when I am logged in.
14	As a feedback requester, I want to mail a copy of the feedback to the feedback giver.
15	As a feedback giver, I want to receive a mail with the given feedback automatically.
16	I want to have a [SA]-app for Blackberry.
17	I want to have a [SA]-app for Windows Phone.
18	As a feedback requester, I want to schedule feedback requests in advance in order to send requests on an appropriate moment.
19	As a user, I want to follow other users in order to look up their feedback anytime.
20	As a feedback requester, I want to specify that a specific question does not apply to avoid that I gather nonrelevant data.
22	As a user, I want to create groups of users (favorites) and create reports for comparing.
23	Being able to see what feedback I have given.
24	As a feedback requester, I want to have my feedback directly available [in some other internal software application of the CC] for myself.

Table 4.4: Submitted requirements in the LDT (1/2)

ID	Requirement
25	As a feedback requester, I want to have separate questionnaires for different groups of people in order to notice differences between technical, functional and managing people.
26	As a feedback requester, I want all functionality to be included in the web application, no nonsense such as a compliment that is only available in the app.
27	As a feedback giver, I want to look up my given feedback from the past.
28	As a feedback requester, I want to use mail addresses of feedback givers from e.g. outlook.
29	As external person (client), I want to specify on which moment in a project a feedback is given.
30	As a feedback requester, I don't want to receive "grades" based on four stars, that is to vulnerable to nuances. Let me receive scores between zero and ten, just like with the other questions.
31	As a feedback requester, I want the estimated time for filling in the questionnaire to be realistic, so that the feedback giver does not think to be finished within a few minutes (expectation management).

Table 4.5: Submitted requirements in the LDT (2/2)

Six of ten requirements were completely or partially not discussed within the development team of the SA before.

Three of the six requirements (#04, #08, and #05) that were considered very important by R5 during the first interview got very high scores (within the top 5 of the list).

However, two of six requirements (#06 and #19) that were considered very important by R5 (in the first interview) got very low scores by the end-users. This fact surprised the requirements practitioners of the SA. These requirements both were related to sharing (sensitive) personal data to other users of the SA.

Below, the results of the items in the questionnaire linked to requirements understanding are depicted (see Table 4.6). The third column shows the average rating.

Position in questionnaire	Item	Average rating
1	I am aware of more wishes for [SA] after using RanX.	3,8/5
2	I think that particular wishes for [SA] are more important after using RanX.	3,6/5
3	I think that particular wishes for [SA] are less important after using RanX.	2,7/5

Table 4.6: Results of questionnaire items concerning requirements understanding

4.3 Motivational factors

Below, the results of items in the questionnaire related to motivational factors are depicted (see Table 4.7).

Item #04 had the highest average rating. Participants agreed most to the statement that they participate, because they were asked by someone to participate.

Item #07 had the lowest average rating. The LD property of delegating voting power was largely rejected for being a motivational incentive for participation in ORG by the end-users.

Position in questionnaire	Item	Average rating
4	I participate in RanX because someone asked me to do so.	4,1/5
5	I participate in RanX because I would like to win a [CC] labeled plastic water bottle	2,2/5
6	I participate in RanX because others can vote on wishes I specified	2,6/5
7	I participate in RanX because colleagues can delegate their voting power to me	2,1/5
8	I participate in RanX because I can adjust my votes anytime	2,4/5

Table 4.7: Results of questionnaire items concerning motivational factors

4.3.1 Unplanned talk with R12

Some information about motivational factors was obtained by an unplanned talk with an employee within the CC. R12 said that he noticed the functionality of delegating voting power. However, he did not use it, because he did not want to spend time for that. It was not clear for him what the profits for him would be, when using that function. ”If I had earned money for it, I would probably have used it” he said.

4.4 RE approaches within the CC

In this section, RE approaches of the requirements practitioners of the CC are described. In the first part, it is depicted which RE techniques are common in the different business units. In the second part, it is described how end-users are involved in the RE processes.

The CC is a detachment firm. Most people of a business unit work on different projects. Therefore, the interviewed persons do not certainly and not completely know what their colleagues do in their projects.

4.4.1 RE techniques used in the business units

The RE techniques applied by requirements practitioners of the CC are described below.

Elicitation

In business unit A, requirements were elicited in collaboration with key stakeholders, such as decision makers in businesses, category managers, sales, website operators during interviews and meetings. R1 stated that he often makes a business function model before he defines requirements. Sometimes requirements practitioners of the business unit use questionnaires/surveys to elicit requirements.

In business unit B, beta groups of end-users are frequently approached, information on download platforms of applications is reviewed, and websites such as Reddit or Facebook would provide lots of feedback to the requirements practitioners. In the project of R2, ”featureleads” who take the responsibility to elaborate specific requirements are defined within the team. The projects often worked according to the Agile principles. According to the Scrum process model, the teams often use user stories, epics and backlogs to gather requirements. Personas and scenarios were also used to get more insights in the requirements. R4 stated that getting insight in the solution’s right to exist, and its rationale (reason for why a solution is built) would be more important than the requirements themselves.

In business unit C, a lot of information is extracted from existing documentation (former requirements analysis or documentation of systems to be replaced by the current project). Interviews/meetings with stakeholders like system operators or business people who need information for making decisions are organized. Often, the systems to be built need to create reports with specific information in a specific format. Examples of such reports are often guiding for eliciting requirements.

Analysis

In business unit A, mails are sent to key stakeholders for reviewing. Often priorities are suggested by the requirements practitioner which are generally accepted by the other stakeholders. Requirements practitioners make use of the MoSCoW scheme for prioritization or the SMART scheme to ensure that requirements meet certain quality criteria. The requirements practitioners do not use specific techniques for dealing with conflicting or duplicated requirements. In case of conflicts, issues are communicated to the key stakeholders and they get solved collaboratively with them. R3 writes test scripts/test cases based on the requirements to assure that requirements have been implemented. Results of these tests are documented in Microsoft Excel. Workshops are organized for creating requirements documents in some projects. Models like use cases or process diagrams are used frequently. According to R3, sharing narratives are starting points for discussions about requirements with stakeholders. It is difficult to verify that end-users are satisfied. This information is deducted from conversations with key stakeholders.

Requirements practitioners in business unit B check what people of the beta group think of new features. Requirements are discussed frequently within the development team. The technique of "planning poker" is used for prioritization. Demos of the application or "clickable prototypes" are important points of reference for discussions between developers and other stakeholders. Analytics are used to have quantitative evidence for what is important for end-users. Also, in business unit B the MoSCoW scheme is used for prioritization.

In business unit C, the development team discusses requirements to reach certain qualities like unambiguity. Requirements practitioners try to find rationals for requirements in the documentation of former requirements analysis or documentation of former systems. R6 stated that in the end, the business will always check whether the implementation adds value to the organization which is the ultimate feedback for evaluating requirements.

Documentation

In general, business units do not seem to use elaborate tools for documenting requirements. Requirements are often documented in Excel.

In business unit A, generally no formal requirements documents are used.

Requirements practitioners and key stakeholders prefer frequent communication according to the Agile principles. Business processes (that are linked to requirements) are formulated in Adonis or Aris. Two years ago the business unit worked with the requirements tool ”Caliber”. The usage was ceased later. Diagrams are often used to visualize ”elements of a requirement”.

In business unit B, tools like ”Jira” and ”Confluence” for documenting requirements are used beside the traditional Microsoft Office tools like Word, Powerpoint and Excel. Scenarios are an important type of documenting requirements for this business unit.

R6 did not mention any other tools than Excel for documenting requirements.

Change Management

As with documentation of requirements, changes of requirements are documented mainly in Excel in each business unit. Only business unit A mentioned that ”Caliber” was easy to use for requirements changes, but after policies stopped its usage, it is not in common use any more by the requirements practitioners.

4.4.2 User involvement

In this section, it is described how end-users of software are involved in the RE processes of the CC for each business unit.

Business Unit A

Requirements practitioners of the CC stated that user involvement is not needed. In projects with five million end-users, it is necessary to rely on the input of representatives. They said that this communication with representatives would be sufficient. They do not exclude that new requirements had been identified if end-users would have been requested for giving input.

R3 underlined that the input of end-users is not the most important criterion for making decisions about requirements to be implemented. They do not focus on gathering information, but delivering a product as fast and agile as possible to the market.

Generally, R3 does not miss the input by end-users for his projects.

R1 stated the following problems with user input: It is not always clear enough to deduce high quality requirements from it and end-users would not understand what is meant by ’requirements’. R3 said that he is involved in projects where end-users are not known yet. Communication with end-users would be difficult, because they are spread around the world.

Business Unit B

R2 mentioned that they process input by end-users (reviews on app stores, websites such as Reddit or Facebook, beta group), but that they try to surprise the end-user with solutions that are better then the end-users ever could imagine. He mentioned the quote that is commonly attributed to Henry Ford: "If I had asked people what they wanted they had said faster horses"[1].

R4 discussed prototypes with a group of end-users. The feedback was valuable to get more confidence that users are satisfied.

Problems with user input would be the following: End-users would not have the required system knowledge to distinct between what is possible and what not. Not all wishes by end-users could be fulfilled. Sometimes different user groups would have conflicting demands (e.g. simpler interface versus an interface that offers more functionality). Additionally, end-users would suggest features which they do not use after they had been implemented. Analytics could help confirming or refuting needs of end-users to solve this problem. In some cases principals promised to facilitate talks with end-users, but later it was found out that these users were actually not available.

Business Unit C

R6 stated that user input should be gathered and elaborated by representatives such as senior users. He does not feel responsible for this task. Generally, requirements are formulated by business people. R6 could not mention any situations in which he missed the input of end-users. R6 considered the communication with representatives of end-users sufficient.

4.5 Potentials and areas of concern

In this section, potentials and areas of concern encountered during the research were listed. The requirements practitioners of the CC would (currently) not use a LDT for RE purposes. In the remaining section, subsection 4.5.2, the main reasons for this were listed.

4.5.1 Main clusters

In this section, potentials and areas of concern are presented. At first, an overview of the clusters is provided. Additionally, all potentials and areas of concern encountered during this case study are explained. The explanations are structured according to the main clusters. The findings are motivated with quotes. These quotes were mainly kept in the original language which

[1] Webpages like `https://hbr.org/2011/08/henry-ford-never-said-the-fast` show that it cannot be confirmed that the American industrialist really said this.

was used during the whole case study (Dutch). Some quotes were translated into English.

Overview

The table below (Table 4.8) shows clusters of the findings that were associated to potentials or areas of concern for using LD in RE. Sometimes an area of concern is a potential at the same time and vice versa. Therefore, markers for this are included in the table.

The first column shows an ID of a cluster for communication reasons. The second column displays a label of each cluster so the explanation of the cluster can be easily found in the sections that follow. The third and fourth column specify whether the cluster can be associated to "potentials" (in the table abbreviated with '**P**') and/or "areas of concern" (in the table abbreviated with '**AoC**'). The fifth column specifies whether the cluster was linked to a finding that was considered very specific to LD and its application in RE. Columns 6, 7 and 8 specify which respondents supported the findings that are related to the cluster.

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
ID	**Label**	**P?**	**AoC?**	**LD specific?**	**CC**	**SA**	**Others**
1	Applicability	X	X		R4, R2, R3, R1, R6	R5, R7	
2	Quality of (set of) requirements	X	X	X	R1, R3, R4, R6, R2	R5, R7	X[a]
3	Time consumption	X	X	X	R2, R4, R1, R6	R5, R7	R28
4	Clear profile		X		R4, R6		
5	Common use		X		R1, R4		
6	Understanding of LD		X	X			R28, R29
7	Lack of commitment		X		R1		
8	Innovation		X		R3		
9	Trust		X	X	R6		

Table 4.8: Main clusters of potentials and areas of concern

[a] One of the project leaders and initiators of the SA

Applicability

The cluster "Applicability" includes all potentials and areas of concern related to different situations in which a LDT for RE should/could (not) be applied.

Potentials A LDT could be used in every area where feedback of end-users can be gathered and is considered valuable[2].

A LDT could be useful for marketing agencies which investigate what needs of users are of e.g. e-commerce-products[3].

A LDT could not only be used for requirements elicitation, but also for measuring the usage of implemented functionality so that it would be a part of an overall measurement framework[4].

The requirements practitioners suggested that other people than just end-users should submit input to a LDT. The input of administrators[5] and customer services[6] would know valuable information that could be used for new requirements.

A LDT could be used for refining deliverables like brainstorming[7] results or an unclear backlog[8].

[2] R3: "Op zich zo wat ik daarvan heb (begrepen) heb je een tooltje ontwikkeld om feedback te analyseren. Dat kan in allerlei situaties handig zijn, dus niet alleen bij echte requirementstrajecten, maar overal waar je feedback verwacht."

[3] R3: "ja ik zou me wel kunnen voorstellen dat (een tooltje als RanX interessant is voor als) je als onderzoeksbureau of wat dan ook enquêtes houdt (over) de consumenten die e-commerce-producten gebruiken."

[4] R4: " Dat gaat vooraal om (een) measurement-framework. Dus wat je zou willen is dat als je vooraf vraagt van "wat wil jij beste eindgebruiker?/Wat zou jij willen in deze oplossing?/Wat zou je waarderen aan deze oplossing?", als je het dan gerealiseerd hebt dat je ook gaat meten of dat dat dan ook in der daad (de) meest waardevolle (suggesties) zijn. En dat betekent dus dat daar vooral een measurement-framework omheen moet komen, waarbij je de feedbackloop sluit. Dus niet alleenmaar vragen aan het begin, maar ook echt gaan meten van "gaan die mensen dat ook echt doen?" En (ultimo) natuulijk ook die mensen teruggeven van "nou (...) bij nader inzien hebben we gezien dat dat toch niet het meest gebruikte feature was", net als dat je zegt bij [[SA]]."

[5] R1: "(. . .) dan kun je wel met een paar belangrijke stakeholders om de tafel gaan zitten, maar dan mis je misschien een hele boel voorbeelden (die de klantenservice kent) die handig kunnen zijn."

[6] R5: "Ik denk echter dat de meeste ideeën meer door de beheerders van de software worden aangedragen. Dat die dicht in contact staan met de eindgebruiker en dat vervolgens krijgen we regelmatig eigenlijk van de beheerders ideeën die ze dan naar ons toesturen."

[7] R3: "The picture I have of that tool is that you gather opinions from a specific group of respondents. What you could do with it would be e.g. using it as a brain storming result and refine it."

[8] R4: "(bij een tool als RanX zou je) kunnen denken: 'oh ja wacht we hebben hier wel een soort onduidelijke backlog' of 'we hebben hier eigenlijk bredere commitment nodig van de rest van de organisatie, maar niet het budget om die allemaal door workshops heen te trekken."

A LDT could be used for gathering new ideas[9,10,11]. It has the potential to gather requirements where the key stakeholders themselves did not think of[12]. A LDT allows to reach a broader audience for RE activities[13,14].

A LDT could be used for making decisions between two options[15].

A LDT could be used if it is too busy for stakeholders to plan a meeting (as alternative to long mail threads)[16].

Furthermore, if a LDT would be in common use, it could be used as memory aid for key stakeholders[17].

A LDT could be used to stimulate acceptance of new requirements[18]. For example in cases where wider commitment by the user group is needed, but no budget to let them participate in workshops[19].

Developers would learn how to communicate new features to the user group[20].

Areas of concern A LDT would not be suitable for projects with fixed requirements by key stakeholders (situation that regularly appears at CC when e.g.

[9] R3: ”Over het algemeen zijn er twee verschillende manieren (om een LDT te gebruiken): een als ideeën-spuier: ’waar zitten de knelpunten in je dagelijks proces, wat voor verbeteringen (vind je) belangrijk?’, dus helemaal in het begin. En de andere is iets verder: ’We zouden het zo kunnen doen, we zouden het zo kunnen doen. Wat lijkt jou de betere keuze?’ ”

[10] R4: ”As such it is interesting to use it (the LDT) for making an inventory (of ideas).”

[11] R6: ”(. . .) such a tool would of course be very suitable for senior users to understand what (needs of users are).”

[12] R6: ”But in general, end-users use the systems daily, they encounter lots of matters.”

[13] R6: ”This (using a LDT) means that you are able to gather information from a broader audience.”

[14] R4: ”(. . .), but of course, you cannot have face-to-face dialogues around the whole world. (. . .)(for reaching acceptance around the whole world) you could use such a tool.”

[15] See footnote 9.

[16] R1: ”(in sommige situaties) moet je eigenlijk een meeting inschieten, maarja dan is het druk, dus ik kan me voorstellen dat er situaties zijn dat als het al in gebruik is en ik hoef me daar niet druk over te maken dat ik zeg ’ok, laten we (een tool als RanX) dan gebruiken’.”

[17] R1: ”Stel dat iedereen ermee bekend is en iedereen heeft inloggegevens want ze gebruiken het al een paar jaar en het gaat prima, (. . .) ja dan kan ik het zelf ook wat meer sturen, dat het ook minder uitmaakt dat het niet SMART is geformuleerd, maar kijk als zij ergens aan denken en het anders kwijt zouden raken of zouden vergeten omdat de workshop drie weken later is voor dit project, ja dan kunnen ze (in de LDT) een opmerking maken en dan neem ik het wel degelijk mee en dan hoef ik het niet per se te zien als goed geformuleerde SMART geformuleerde requirement, nee dat is gewoon input.”

[18] R4: In plaats dat je het over de schutting gooit (en zegt:) ’hier is het’, ’hier hebben je collega’s voor gekozen’ zeg je dan ’kijk, hier ben jij mee betrokken geweest’/’hier ben jij onderdeel geweest van deze constructie’ en dat helpt met acceptatie soms.”

[19] See footnote 8 on the previous page.

[20] R5: ”Nee maar weet je ik denk dat het wel nuttig is voor ons om te kijken van ’hoe maken we duidelijk (. . .) ook aan de gebruiker van wat wij willen maken?’ ”

clients determined the requirements themselves)[21].

A LDT would not be able to obtain high level requirements[22].

The matters discussed in the LDT should be linked to the understanding/knowledge of the end-users’ real world[23]. This suggests that features should be imaginable by end-users. Therefore, a LDT would not be suitable for building new products. R1 confirmed this in his statement that a LDT would be useful in each project which has a customer service. In case of a new product, a customer service cannot provide information for requirements. He suggested that a LDT would not be useful because of that fact[24].

A LDT would not be suitable for software projects having just a small user group[25].

A LDT cannot deliver other kinds of requirements than end-user requirements e.g. organisational requirements[26].

Quality of input

Potentials Even if the input is not in line with guidelines such as the SMART scheme there would be still a possibility that the input is valuable[27].

Areas of concern Input by end-users should be in line with the SMART scheme[28].

Some wishes submitted in the LDT were considered not realistic by the requirements practitioners of the SA[29].

[21] R2: ”Er zijn gewoon een aantal projecten die wij doen die zijn voor een gefixed budget: ’dit moet er deze release komen’.”

[22] R4: ”Ik ben vaak bezig met hoog over requirements. Die doe ik ook best wel vaak persoonlijk. Dus in persoonlijke dialoog. Helpt een digitale tool met die persoonlijke dialoog? Ik denk minder.”

[23] R3: ”(het) moet de mensen wat zeggen en dan kunnen ze er een mening over hebben; dan heb je ook iets aan die feedback.”

[24] R1: ”(een LDT is nuttig) waar dus een klantenservice een rol bij heeft. Stel dat je een nieuw product gaat ontwikkelen dan volg je gewoon de standaard stakeholders activiteiten om tot requirements te komen dat uit te werken en dan de klantenservice die krijgen geen telefoontje daarvoor want het gaat om (een) nieuw te ontwikkelen product.”

[25] R6: ”Dat lijkt me voor hen wel nuttig ja, tenzij (...) het aantal gebruikers heel beperkt is dan zijn alle gebruikers in feiten senior user, dan heeft zo’n tooltje natuurlijk geen zin.”

[26] R3: ”vragen van ’wat voor rapportages heb je nodig?’ en ’wat voor betaalmethoden wil je in je webshop aanbieden?’ en ’hoe veel mag het kosten?’, ’wat moet de verhouding zijn tussen de vaste kosten en de transactiekosten?’ Allemaal zaken die kun je niet zomaar stellen aan consumenten.”

[27] See footnote 17 on the preceding page.

[28] R1: ”(als requirements niet SMART geformuleerd moeten worden) dan beginnen mensen te stemmen op iets wat onduidelijk is.”

[29] R5: ”Dus ik kan me (requirement 28, see Table 4.5 on page 43) wel voorstellen, maar het is niet echt haalbaar/niet echt realistisch om daar weken tijd in te stoppen”

Some wishes submitted in the LDT would not really add value, because they are already implemented according to the requirements practitioners of the SA[30].

There were multiple wishes submitted in the LDT expressing the same requirement[31].

Results of the LDT would be comparable with the results of a (common) questionnaire. Although the LDT supported a special voting system, the enriching step have been missed (because people did not use the main LD property)[32].

According to many respondents, it would be very likely that end-user input within ORG would be ambiguous. It would not be clear what the meaning of end-user input (set of requirements and the voting results) would be if you do not have the possibility to clarify the input as it is possible in face-to-face meetings[33,34,35,36,37].

It would be very difficult to extract highly valuable new insights from plain text[38].

There should be enough evidence that the input by LDT users can be considered reliable[39].

[30] R5: "(requirement 29, see Table 4.5 on page 43) is een beetje lastige ... dus de feedbackgever moet aangeven wanneer de feedback gegeven is, maar dat doet die dus eigenlijk op dat moment zou ik het idee hebben"

[31] R5: "als ik dit zo lees dan heb ik het idee dat daar (requirement 22 and requirement 19, see Table 4.4 on page 42) een soort overlap in zit."

[32] R7: "Nu is het meer een wat uitgebreidere vorm van enquête, wel zit er natuurlijk wel echt met stemgedrag zit erin weet je, maar het is net (...) het extra stapje mis je dan."

[33] R4: "je kan doorvragen, de dialoog is veel dieper, er is überhaupt al een dialoog in plaats van 'wat vind je hiervan?' en dan van 1 tot 7 weet je allemaal van die arbitraire vragen met arbitraire schalen die je verder weg brengen van wat je probeert te bereiken."

[34] See footnote 28 on the previous page.

[35] R3: "Maar de interpretatie van wat we wel doorgeven gaat dan op sommige punten net een iets andere kant op dan wij bedoelen. Dat is heel complex. Als je met verschillende talen werkt in verschillende landen bijvoorbeeld."

[36] R5: "voor mij is het allemaal beetje logischer dat ik denk van 'ah ja dit willen we', (...) waarschijnlijk vertel ik dan te weinig en dan wordt er een vertaalslag overheen gedaan (...) en soms heb je het echt wel over andere dingen denk ik uiteindelijk."

[37] R5: "Weet je eigenlijk mis je hiervan waarom ze het afkeuren of waarom ze voor stemmen af en toe. Zo van 'als ze dit zeggen; dit is vanwege mijn privacy', dan is dat voor ons bijvoorbeeld heel duidelijk, maar misschien (zijn) het andere redenen."

[38] R4: "Dus het is dan ook wel heel moeilijk om te zien, zeker als je niet in die dialoog zit, om met platte tekst, daar ontzettend waardevolle nieuwe dingen uit te krijgen."

[39] R7: "Ja je weet niet echt of het echt speelt weet je; Het is natuurlijk een iemand die dat zegt en er zijn een paar mensen die daarop reageren, maar weet natuulijk niet wat het (gros) van de gebruikers daarvan (vinden)."

Enough end-users should participate in the LDT[40,41].

It should be verifiable that delegations of voting power were received honestly. R6 had doubts about the reliability of the mechanism for copying votes. He thought of extortion: A person could claim from someone else to copy votes from him and threaten this person in case he/she does not do so. For R6 it is important that the reliability of input from end users can be verified[42].

Value is sometimes not created by satisfying individual requirements, but by a combination of requirements. Probably you would not get such information out of a LDT[43].

According to R4, people have difficulties to prioritize. You really need to ensure that end-users make a selection of what they find important or not important[44].

R2 suggested that end-users lack crucial knowledge for suggesting satisfying solutions[45].

There are several groups of end-users. Probably some wishes would get a high score on user votes only because one specific group consists of more (active) users than other groups. One group could impose its interests on all other users that way. The functionality of voting via an online tool on wishes/requirements has been disqualified by R1 because a requirements practitioner would be "less in control" about the voting results. It was suggested that a requirements practitioner cannot get a good overview of how votes have been divided on the different stakeholder groups[46].

[40] After the results of the LDT had been discussed with R5 and R7, one of the project leaders and initiators of the SA wrote in an email that he does not really see value in the output of the LDT. He wrote that 25 respondents would be too less for reliable findings.

[41] R5: "Ik vind het wel jammer dat er maar 25 mensen gereageerd hebben."

[42] R6: "ik eis van jou dat jij mij jouw stem geeft want anders dan zal ik eh... (. . .) He dus al dat soort dingen en dat bedoel ik eigenlijk vooral met de betrouwbaarheid van die stemresultaten. In hoeverre gaat dat meespelen? Dat is niet zichtbaar denk ik."

[43] R4: "Een valkuil daarbij is natuurlijk wel dat je waarschijnlijk de functionaliteit te los gaat benoemen en dat waarde soms niet zit in losse functionaliteit, maar in combinaties van."

[44] R4: "Want als je gaat vragen 'wat wil je allemaal hebben?' dan krijg je ook een kind in een candystore die kan dan niet kiezen en kiest alles, dus die wijst naar alles (en zegt:) 'ik wil dit allemaal hebben'. En als je gaat zeggen: 'wat zou nou het eerste zijn waar je echt iedere dag waarde aan ontleend?' dan denk ik dat je daar wel een waardevolle inputbron uit kan hebben/zou kunnen krijgen."

[45] R2: "Daarbij is het wel heel belangrijk voor ons dat wij net even een stapje slimmer zijn dan gebruikers. (. . .) maar wij zijn hier in feiten wel met (. . .) natuurlijk allemaal diegenen met de systeemkennis en de platformkennis en wat wel en niet kan. (Dus we proberen) ook echt een stapje voor te zijn, vervolgens meer te bieden dan dat ze zelf denken te willen zeg maar."

[46] R1: "dan hebben we drie verschillende afdelingen, want dan van de marketingafdeling

Time consumption

Potentials A LDT can be more convenient than a questionnaire/survey, because such a tool would be less time-consuming[47].

Areas of concern The processing of LDT results would be too time consuming[48].

Stakeholders generally would take less time for giving input into a LDT compared to other RE techniques like face-to-face meetings[49].

Maintaining the recent requirements of the requirements practitioner in a separate tool like a LDT would be time consuming[50].

Setting up the LDT, explaining the LDT and solving e.g. authentication problems should not require too much time for the requirements practitioner[51].

Activities for obtaining vote delegations suggested in the LDT[52] sounded ”very time-consuming” to one of the LDT users[53].

kunnen we eindgebruikers hebben, van de klantenserviceafdeling (die) kunnen er misschien vier keer zoveel eindgebruikers hebben en die stemmen natuurlijk in grote getallen op requirements die voor hun belangrijk zijn en dan krijg je alleenmaar een winnaar omdat ze met meer mensen zijn. Dus dat is weer een reden om dat stemmen sowieso niet te gebruiken.”

[47] R6: ”Ja ik denk dat zo’n tooltje wel handiger kan werken dan een enquête. (...) ik heb zo het idee van nou je kunt zo’n tooltje openen en dan ben je een paar minuten bezig en dan ben je klaar. (...) En een enquête dat kan wel vaak veel uitgebreider zijn dan ben je soms een aantal uren mee bezig om dat allemaal in te vullen.”

[48] R2: ”Zou je bijvoorbeeld alleen multiple choice doen dan zou het ook publiekelijk kunnen, dan kun je daar vrij gemakkelijk statistieken van genereren als gebruikers echt zelf suggesties mogen gaan doen ja dan kan het met zo veel gebruikers echt uit de hand lopen denk ik.”

[49] R4: ”het is geen gesproken tekst neem ik aan, je typt iets, mensen nemen daar over het algemeen weinig tijd voor, ze zullen niet denk ik (gaan zeggen) ’ga eens echt lekker op de zondag middag twee uurtjes requirements toevoegen’, dus het is iets wat mensen snel doen.”

[50] R1: ”Ik ga (mijn lijst met requirements) (...) denk ik niet allemaal kopiëren naar RanX zodat iedereen kan gaan stemmen. Dat zou een hoop werk zijn.”

[51] R1: ”Ja maar het moet voor mij niet al te veel werk zijn om dat systeem te gaan uitleggen en uitrollen en links sturen en inlogproblemen te hebben met die mensen en dat ik dat daar tijd aan kwijtraak.”

[52] In the LDT, the following text was displayed: ”Convince other users to delegate their voting power to you by submitting new wishes, good comments and by adjusting your votes frequently”

[53] R28 wrote in a seperate email after using the LDT: ” ’Overtuig dan medegebruikers om hun stemvermogen aan jou te delegeren door wensen in te dienen, goeie opmerkingen te plaatsen en je stemmen regelmatig aan te passen!’ Dat klinkt als time-consuming.”

Clear profile

A LDT should have a clear profile to be useful for requirements practitioners[54].

The requirements practitioners should know for sure that a LDT adds value in certain cases[55].

Common use

A LDT should be known to a majority of people and most people should have an account for it[56].

Understanding of LD

Users of the LDT wrote that they do not understand how simultaneous direct and indirect voting works[57,58].

The interviews showed that some requirements practitioners of the CC did neither understand the concept to the full. A statement by e.g. R1 indicated that he confused the concepts SA and the LDT. He imagined that a customer service of some software application would use the software of the SA for gathering end-user requirements[59]. He seemed to confuse the SA with the LDT for almost the whole second interview.

Lack of commitment

On page 54, the phrase ”less in control” by R1 is mentioned. He linked ”less in control” not only to a unsatisfactory overview of how votes have been divided on different stakeholder groups, but also to absent participation in RE activities. Stakeholders would be less committed to engage in the requirements discussion via a tool than via face-to-face meetings. In case of meetings they need to reserve a time slot in their agenda because their manager says to do so and you immediately see that a stakeholder accepts a face-to-face meeting in his agenda. He suggested that a discussion via a LDT could not provide such a high level of commitment. However, as a requirements practitioner, you depend on high commitment[60].

[54] R4: ”En dat je dus goed moet nadenken over 'goh, wanneer zet ik welke tool in?' en 'wanneer is dat dan RanX?' en 'wanneer niet?' ”

[55] R6: ”Ja het zou kunnen. (...) Maar of het veel toevoegt dat vraag ik me af.”

[56] See footnote 17 on page 51.

[57] R28 wrote in a separate email after using the LDT: ”Snap niet hoe dit werkt.”

[58] R29 wrote in an answer to question 9 (see Table 3.1 on page 31) the following: ”Ik snap het delegeren van stemmen niet. Ik vroeg me af hoe groot mijn stemgewicht in totaal was.”

[59] R1:”Dus als zo'n klantenpanel nu met [[SA]] zou willen gaan werken, ja dan ga ik wel met hun praten (. . .)”.

[60] R1: ”(. . .)dus vergeleken met zelf in control zijn en een meeting inschieten, ook al is het maar een interview of een workshop of zo, dan zie je meteen (ze hebben) het geäc-

The need for remembering credentials would be a reason for end-users not to participate in a LDT[61].

Innovation

As an innovator, you do not want to reveal valuable information in the public domain which can be observed by competitors[62].

In real innovative projects, end-users cannot be accessed because they would be unknown[63].

Trust

The distance to people you trust by delegating voting power to them would be bigger than knowing the people face-to-face[64].

4.5.2 Intention to use LD for RE by CC

R1 has doubts whether he would ever use a LDT for RE. Main reasons he mentioned were lack of control[65], ambiguity[66], lack of commitment[67].

R2 would not use the LDT for gathering input from a crowd of end-users, because he thinks that processing the data would consume too much time[68].

R3 would not use a LDT in his current situation, because he does not recognize any necessity to gather input from end-users directly[69].

cepteerd of niet, ze moeten accepteren uiteindelijk want ze hebben een mandaat gekregen voor dit project van hun manager om mee te doen dus . . . ja. Terwijl als je hun een link stuurt van 'oh hier kun je feedback geven' dan kan dat blijven liggen. En als je dan Agile werkt waar alles heel snel moet gaan en dat blijft drie weken liggen, dan (. . .) heb je best wel een probleem of een gemiste kans."

[61] R1: "maar ook het hele inloggebeuren, want mensen hebben inmiddels zo veel passwoorden en dan daar inloggen, daar inloggen en dan komt dit erbij en dan kan het blijven liggen en dan hebben ze daar misschien weer vragen over of denken van "oh ja", weet je het is een extra reden om het misschien even te laten liggen."

[62] R3: "Je wilt ook je concurrenten niet wijzer maken."

[63] R3: "Wij zitten natuurlijk in de situatie dat we ergens naartoe willen met een product dat er nog niet is. En dat we ook de klanten die het gaan afnemen, ja dat zijn onze klanten nog niet he, die moeten we gaan werven."

[64] R6: "(. . .) Dan is dat tooltje (. . .) ook weer laagdrempeliger omdat naja je bent niet afhankelijk van locatie. Je bent alleen afhankelijk van mensen die je kent. En hoe je die kent op basis van aanbeveling (. . .) of op basis van promotie of op basis van dat je regelmatig met elkaar contact hebt gehad (...) (die afstand) kan dan ietsjes groter zijn dan vroeger.

[65] See footnote 46 on page 54.

[66] See footnote 28 on page 52.

[67] See footnote 60.

[68] See footnote 48 on page 55.

[69] R3: "(Als je) een grotere groep respondenten hebt die representatief zijn en die je rechtsstreeks wilt benaderen in plaats van via een vertegenwoordiger (. . .) daar zie ik het toegepast. Maar bijvoorbeeld (in) mijn huidige situatie zitten wij niet in zo'n situatie."

R4 would not use a LDT immediately, because in his opinion, a LDT would not help in gathering high level requirements[70]. Therefore, he prefers the personal dialogue with stakeholders. However, if the LDT would have a clear profile[71], he would see it as one of the valuable tools requirements practitioners could use. He could imagine experimenting with a LDT to support other RE techniques[72].

R6 said that it would be possible for him to use a LDT, but he has doubts whether it would add value. Therefore, it is important to know in which cases a LDT would add value. This underlines the necessity of a clear profile.

Chapter 5

Discussion

In this chapter, the most important remarks on assumptions, limits, shortcomings, and unused opportunities of this research that could have influenced the results and conclusions are described.

5.1 Important assumptions

It was assumed that users of the LDT did not manipulate other users and that nobody created multiple accounts in the LDT. No mechanism was implemented in the LDT to prevent that one person registers and uses several accounts. Users needed to specify an email address in the corporate domain of the CC. The authenticity of a user account was roughly checked by ensuring that the specified email addresses are linked to employees of the CC on the intranet of the organization. Theoretically, it would have been possible that one person registered several accounts with a colleague's email address and misused the increased voting power to alter the results. Based on the broadly distributed votes on proposals, the probability that a user misused the LDT this way is considered very low.

[70] See footnote 22 on page 52.

[71] See footnote 54 on page 56.

[72] R4: "Maar het is sowieso interessant om meerdere tools/meerdere vormverschijningen in te zetten om vergelijkbare dingen op te halen."

5.2 Understanding of parts of the research

R3 used another definition of the term 'end-user' than the researcher. He associated webshops which use R3's software solution with the term "end-users". In this research, end-users were defined as individuals directly interacting with a software solution.

R2 said that processing data of the LDT would be very time consuming. This indicates that he did not understand the potential of LD techniques communicated at the beginning of the interview. This underlines that the concept of LD needs to be further elaborated and that more attention is needed for clear explanations that are easily understandable by others.

5.3 Shortcomings of this research

The small group of people that filled in the questionnaire cannot be considered representative for certain conclusions about motivational factors. If the research had focused more on this part, more people could have been invited to participate in the LDT. However, findings were still considered indicative for the potential of LD techniques to motivate users to participate in ORG.

The amount of areas of concern encountered in this research is much higher than the amount of potentials. Probably the set of questions asked during the interviews was a reason for this. This is explained by the following example. The users of the LDT delivered some new requirements, but the requirements practitioners of the SA were not asked how they would actually deal with requirements that scored very low. Potentially, the LDT results would led to real adjustments in the set of requirements. Such "potentials" were not investigated in this research due to time constraints.

Seven of the 25 requirements within the initial set of requirements (see subsection 3.3.5) were not added because (1) only one source of information (the file communicated by mail) was actually used and (2) four requirements of the file communicated by mail were not understood at the time when the requirements have been submitted to the LDT.

Many closed-ended questions were asked during the second round of the interviews due to time constraints. This probably influenced the answers of the respondents. One example is the following question asked to R6 during the second round interview: "Do you think that a tool such as RanX would be useful to get more confidence in what representatives of end-users say?" The respondent affirmed the suggestion made by this closed-ended question. Such questions might avoid disagreement by the respondents.

This research did not investigate all possible LD properties (see subsection 2.2.2). A single LDT was used which implemented a specific set of LD properties (see subsection 2.3.1). Topic-based delegations could have been investigated by keeping the functionality for switching different "discussion rooms" originally provided by the LDT. For simplicity reasons (see subsection 3.3.2), this functionality was disabled. Vote recommendations were not supported by the LDT at all.

The only LD properties that were investigated for motivating factors were the properties "simultaneous delegating and direct voting" and "voting anytime" (see subsection 2.2.2). Item #07 in the questionnaire (see Table 3.1) should measure "simultaneous delegating and direct voting" which was considered the main LD property. Unfortunately, this question did not cover the property completely. The main LD property is not only concerned with delegating voting power, but is especially characterized by the combination of being able to vote directly on proposals *and* to delegate voting power simultaneously. This fact was realized after the questionnaire was filled out by respondents and there was too little time to repeat the questionnaire data gathering. The motivational factor of other LD properties ("collaborative discussion" between the participants and "topic-based delegations") were not measured by the questionnaire at all, because these properties were not available in the used LDT.

In the LDT, a collaborate discussion was very limited, because each user could only submit one comment for each proposal.

5.4 Other factors that may have influenced the results

The employees of the CC are for the most part highly educated and very familiar with IT system design. Input in the LDT would be probably different if people would have participated who received education in lower educational levels or who were less familiar with IT system design.

Requirements 6 and 19 were related to sharing (sensitive) personal data to other users of the SA (see section 4.2). LDT users may have interpreted this as sharing data continuously without confirmation or with an onetime confirmation, although this was not meant by R5 when he provided the initial set of requirements for the LDT. Probably, these requirements were submitted to the LDT too incautiously.

The explanations provided in the LDT were not always correct. It was stated that a specific button would be at the bottom of the page but that button was placed under a text section and not at the bottom of the whole page. Such explanations might confuse the LDT users, and therefore might result in lower participation.

Finally, the main questions of the first round interviews with the requirements practitioners of the CC could have been incorporated in the second round interviews. In that case only five instead of ten interviews with requirements practitioners of the CC would have been required for probably similar results. The time saved by these lower amount of interviews could have been used for other parts of the research, such as focusing more on inviting people for participation in the LDT.

Chapter 6

Conclusion & further research

The following research question was stated: How can LD techniques contribute to the RE processes of existing software applications? This question was split into four sub questions as follows: (1) What are the effects of using LD techniques on the requirements understanding of software developers/designers and users? (2) What are potentials and areas of concern when applying LD techniques for gathering end-user requirements of an existing web application? (3) Do end-users feel more motivated to participate in the RE processes by LD techniques? And (4) how can findings be used to contribute to the RE processes of the CC? The answers to the sub questions are subsequently provided. Finally, a conclusion is drawn to the main question.

6.1 Conclusions related to the subquestions

6.1.1 What are the effects of using LD techniques on the requirements understanding of software developers/designers and users?

The usage of a LDT slightly changed the requirements understanding of software developers and users. However, the number of respondents participating in the LDT was too low to consider the LDT results reliable.

The feature of copying votes had not been used by the respondents in the LDT. Therefore the changes in requirements understanding have not been reached by LD techniques, but by using an ORG tool in general.

6.1.2 What are potentials and areas of concern when applying LD techniques for gathering end-user requirements of an existing web application?

Many areas of concern and potentials have been identified. The most important are provided in this answer.

Generally, the respondents made more critical statements against the concept of LD for RE than optimistic ones. Far more areas of concern than potentials for LD in RE have been encountered during this study.

The general potential of inspiration for key stakeholders have been confirmed by most of the requirements practitioners. Besides, a LDT could be used to stimulate acceptance of new requirements. However, a LDT could also lead to the opposite, e.g. in case of conflicts between organizational and end-user requirements.

In some cases, the quality of input by end-users did not satisfy. More research could focus on how to improve the quality of input about requirements, especially by non-professionals who are not educated in IT system design or RE. Such research could also indicate whether user stories are suitable for the formulation of requirements in a LDT.

Activities for obtaining vote delegations suggested in the LDT were perceived "very time-consuming". Further research could investigate what people can do so that others copy their votes. In particular, within a digital environment in which participants do not see each other face-to-face getting delegations can be considered a difficult endeavor. Especially, due to the many areas of concern linked to the LD concept encountered in this study.

There should be enough evidence that the information provided in the LDT is reliable. Criteria that indicate when input by end-users can be considered reliable and how to reach such a reliable input could be researched in future. Certainly, the number of end-users participating in the LDT is crucial for this. Other research topics could focus more on how to increase that number of participants.

End-users probably did not understand the concept of LD or did not accept the applied LD properties to implement the LD concept. Topics for further research may be better explanations of the used LD properties and which concrete LD properties fit best to the acceptance of users.

Finally, an area of concern is the audience that is allowed to use a LDT for RE. This group of people should be clearly defined. Should all stakeholders participate in such a tool? Should key stakeholders or minorities of end-users obtain a higher voting weight? Should developers be allowed to comment to the input by end-users for acceptance or expectation management? How can a distinction be made between the different stakeholders (e.g. a key stakeholder can be also an end-user in some cases, or during the usage of the LDT theoretically any stakeholder could have given input)? Further research could focus on these questions.

6.1.3 Do end-users feel more motivated to participate in the RE processes by LD techniques?

All LDT end-users responded mainly because they were asked by the researcher or unit manager to do so (see item 4 in Table 4.7). No LD technique used in the LDT during this case study can be considered a motivational incentive for end-users to participate in ORG. Further research should investigate more the motivation for stakeholders to participate in ORG. Attention should not only be paid to end-users, but also to the requirements practitioners who need to facilitate the ORG and need to maintain the requirements in the LDT and other requirements repositories.

6.1.4 How can findings be used to contribute to the RE processes of the CC?

All requirements practitioners of the CC stated that they would not use a LDT immediately for their RE activities. Main reasons were the fact that a LDT (or an ORG tool in general) cannot be used in each case and that the quality of requirements would be lacking.

ORG added value to the SA which considers end-users ”most important”. In the CC, the end-user satisfaction does not always have the highest priority. The need for direct communication to end-users is therefore low in business units A and C. The end-user satisfaction was considered more important for business unit B than in the other business units. Within this business unit, LD techniques for RE could add value for the CC.

The CC could put more resources in experimenting with a LDT than was available for this research. This research used a LDT without heavily improving it for the context of RE. A further step could be that the CC develops a LDT that fits more to the field of RE. Such an implementation could address all or at least more potentials and areas of concern reported in this research, e.g. better explanations about the LD concept or a evidence-based policy about the audience that is allowed to submit input to the LDT.

The LDT should be in common use by many people so that everyone understands the LD concept. Besides, common use of such a system could reduce people’s resistance to use a LDT, because e.g. remembering the credentials would be easier. Common use could be achieved by building a portal that would gather requirements not only for one tool as in this case study, but for theoretically all existing software applications in which end-user satisfaction has a high priority. A company such as the CC could promote to include links to such a tool inside applications they implement and/or give advice for.

6.2 Conclusion related to the main question

This research confirms, albeit not decisively, that ORG contributes to RE processes. However, LD techniques probably did not influence the result. Maybe, LDT users behaved differently because they were influenced by one or more LD properties, but this was not confirmed in this study.

The applied LDT as one further example of ORG added value in comparison to other techniques the developers of the SA frequently use.

Many potentials and areas of concern have been identified. Two of them were not mentioned in preceding literature (Johann & Maalej 2015): The need for common use and the possibility that stakeholders do not understand the concept of LD. Other areas like security were never mentioned by the respondents who were interviewed, but this should definitely be addressed in further research about this topic.

LD techniques could be able to contribute to ORG, and therefore also to the RE processes in general. However, this could not be confirmed in this research. It should be ensured that at least the areas of concern mentioned in this study are covered in a LDT that is used productively. For this, more research is needed as suggested in the answers to the sub questions.

Research is not the only activity that can cover all the areas of concern. The areas of concern of ”common use” and ”understanding of LD” may only be reached if many people get used to applications implementing the LD concept.

Currently, the LD concept is not elaborated on a mature level. This research is one example of applying the LD concept. Many areas of concern are not limited to the RE field and also relevant for other fields of decision making. Research projects dealing with the concept like this one help to elaborate on the concept.

Bibliography

Bano, M. & Zowghi, D. (2015), ‘A systematic review on the relationship between user involvement and system success’, *Information and Software Technology* **58**, 148–169.

Behrens, J., Kistner, A., Nitsche, A. & Swierczek, B. (2014), *The principles of LiquidFeedback*, Interaktive Demokratie e. V.

Glinz, M. (2007), On non-functional requirements, *in* '15th International Requirements Engineering Conference (RE 2007)', IEEE, pp. 21–26.

Green-Armytage, J. (2015), 'Direct voting and proxy voting', *Constitutional Political Economy* **26**(2), 190–220.

Gunter, C. A., Gunter, E. L., Jackson, M. & Zave, P. (2000), 'A reference model for requirements and specifications', *IEEE Softw.* **17**(3), 37–43.
URL: *http://dx.doi.org/10.1109/52.896248*

Johann, T. & Maalej, W. (2015), Democratic mass participation of users in requirements engineering?, *in* '23rd International Requirements Engineering Conference (RE)', IEEE, pp. 256–261.

Krogstie, J., Lindland, O. I. & Sindre, G. (1995), Towards a deeper understanding of quality in requirements engineering, *in* 'International Conference on Advanced Information Systems Engineering', Springer, pp. 82–95.

Ling, K., Beenen, G., Ludford, P., Wang, X., Chang, K., Li, X., Cosley, D., Frankowski, D., Terveen, L., Rashid, A. M. et al. (2005), 'Using social psychology to motivate contributions to online communities', *Journal of Computer-Mediated Communication* **10**(4).

Maalej, W. & Pagano, D. (2011), On the socialness of software, *in* 'Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)', IEEE, pp. 864–871.

Manoharan, A. & Holzer, M. (2011), *E-Governance and Civic Engagement: Factors and Determinants of E-Democracy*, Premier reference source, Information Science Reference (an imprint of IGI Global).

Nuseibeh, B. & Easterbrook, S. (2000), Requirements engineering: A roadmap, *in* 'Proceedings of the Conference on The Future of Software Engineering', ICSE '00, ACM, New York, NY, USA, pp. 35–46.
URL: *http://doi.acm.org/10.1145/336512.336523*

Paetsch, F., Eberlein, A. & Maurer, F. (2003), Requirements engineering and agile software development., *in* 'WETICE', Vol. 3, p. 308.

Paulin, A. (2014), Through liquid democracy to sustainable non-bureaucratic government, *in* 'Proc. Int. Conf. for E-Democracy and Open Government', pp. 205–217.

Renzel, D. & Klamma, R. (2014), 'Requirements bazaar: Open-source large scale social requirements engineering in the long tail', *IEEE Computer Society Special Technical Community on Social Networking E-Letter* **2**(3).

Shaw, M. (1990), 'Prospects for an engineering discipline of software', *IEEE software* **7**(6), 15–24.

Thomas, J. C. & Streib, G. (2005), 'E-democracy, e-commerce, and e-research: Examining the electronic ties between citizens and governments', *Administration & Society* **37**(3), 259–280.

Van Lamsweerde, A. (2001), Goal-oriented requirements engineering: A guided tour, *in* 'Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on', IEEE, pp. 249–262.

Zave, P. (1997), 'Classification of research efforts in requirements engineering', *ACM Computing Surveys (CSUR)* **29**(4), 315–321.

Index