Integration of Social Sustainability in Software Requirements Using Requirements Pattern: A Case of Equality

Thesis submitted for the degree of
Doctor of Philosophy
at the University of Leicester

by

Maryam Ali Said Al Hinai
Department of Informatics
University of Leicester

June 2019

Abstract

Software is taking an increasingly prominent role in all of the areas relevant to social sustainability; e.g. by creating and enabling communities, facilitating connectedness between groups of people, enforcing standards, rules, and laws over individuals' and business' interactions. It enables the sharing of information and facilitate cooperation.

Presently, there is no solution, framework, or method that supports software and requirements engineers in accounting for social sustainability requirements during software development. Most requirement engineers don't even understand what social sustainability is or what it may have to do with software.

Thus, the *main aim of this thesis is to deliver a method* that de-mystifies the notion of social sustainability, and *enables requirements engineers to integrate (otherwise implicit) social sustainability requirements into a software systems specification.*

This thesis tackles this issue by eliciting the structure of the social sustainability concern, as reported within in the current scientific literature (systematic literature review), and delivering a generic, repeatable approach through which each of the identified sub-concerns could be operationalised into reusable requirements. The proposed approach is instantiated for the equality concern, for which a value-based pattern is derived (qualitative analysis), which is operationalised through templates. An elicitation method is also derived using the pattern and usage guideline is proposed.

Several targeted studies are designed to evaluate the developed pattern and method with both requirements engineering experts and general software users. The studies demonstrate the utility and (re-)usability of the method, pattern along with its template and requirements.

This thesis reveals that the generic social sustainability requirements can be used (but contextualised) across all countries and cultures. It also declares that each social sustainability concern can be represented by a value pattern. For equality, this pattern is related to equal support of: stakeholders variability, goal achievement and access to services that will facilitate goal achievement.

Acknowledgements

First, I would like to thank my supervisors, Dr. Ruzanna Chitchyan and Professor Reiko Heckel, for the continuous support and guidance offered throughout my PhD journey. I would also like to express my gratitude to the staff and students in the department of Informatics at the University of Leicester. Moreover, I would like to thank my family for supporting and providing advice all the way from Oman. Also, I would like to thank my friends in Leicester for their support and encouragement. In addition, I am grateful to my sponsor, the Ministry of Manpower and the Higher College of Technology, Muscat, Oman. I would also like to thank Dr. Maria Viskaduraki from the Bioinformatics and Biostatistics Analysis Support Hub (B/BASH) at the University of Leicester for her help in the area of statistical analysis.

Publications

- Maryam Al Hinai and Ruzanna Chitchyan. Equality Requirements for Software Systems: A Survey. In *Sixth International Conference on ICT for Sustainability 2019*. Lappeenranta, Finland. June, 2019 (forthcoming).
- Maryam Al Hinai and Ruzanna Chitchyan. Evaluating Equality Requirements for Software Systems. In *Fifth International Conference on ICT for Sustainability 2018*. Toronto, Canada, May, 2018.
- Maryam Al Hinai and Ruzanna Chitchyan. Engineering requirements for social sustainability. In *Fourth International Conference on ICT for Sustainability 2016*. Amsterdam, The Netherlands, August, 2016, Atlantis Press.
- Maryam Al Hinai and Ruzanna Chitchyan. Building social sustainability into software: Case of equality. In *Fifth IEEE International Workshop on Requirements Patterns, RePa 2015*, Ottawa, ON, Canada, August 25, 2015.
- Maryam Al Hinai and Ruzanna Chitchyan. Social Sustainability Indicators for Software: Initial Review. In *Proceedings of the 3rd International Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy14)*, Karlskrona, Sweden, August 2014.
- Maryam Al Hinai. Quantification of social sustainability in software. In *Proceedings of the Doctoral Symposium at the 2014 IEEE 22nd International Requirements Engineering Conference (RE)*, Karlskrona, Sweden, August 2014.

Contents

List of Figures

List of Tables

Abbreviations

HCI	**H**uman **C**omputer **I**nteraction
ICT	**I**nformation and **C**ommunication **T**echnology
LCA	**L**ife **C**ycle **A**ssessment
RE	**R**requirements **E**ngineering
SII	**S**ocial **I**mpact **I**ndicator
SNA	**S**ocial **N**etwork **A**nalysis
SRPs	**S**ustainability **R**equirements **P**atterns
VATs	**V**ulnerability **A**ssessment **T**echniques
VSD	**V**alue **S**sensitive **D**esign
sCE	**s**ituated **C**ognitive **E**ngineering
SE	**S**oftware **E**ngineering
SLR	**S**ystematic **L**iterature **R**eview
WVS	**W**orld **V**alue **S**urvey
VME's	**V**alues, **M**otivations and **E**motions

Dedicated to my beloved parents, Ali Said Al Hinai and Zuweina Mohammed Al Sulaimani.

Chapter 1

Introduction

This chapter outlines the research subject of the thesis. Section 1.1 discusses the context in which the thesis fits. In Section 1.2, the research problem is discussed and the research objectives are described in Section 1.3. Then, Section 1.4 describes the contributions of this thesis. The research method and the outline of the remaining chapters of the thesis are presented in Section 1.5.

1.1 Research Context

As software increasingly becomes the central mediator in more and more spheres of our lives - from business, communication, innovation, healthcare to education, and even art - we must consider the impact it has on the well-being of humans and society, i.e., on social sustainability.

Social sustainability is defined as “a positive and long-term condition within communities and a process within communities that can achieve and maintain that condition” [2, 3]. In a social sustainability literature review, Landorf concluded that "basic needs and equity are consistently evident as fundamental themes of social sustainability. Both concepts are necessary for the physical and psychological survival of individuals" [1]. In addition, social sustainability is related to access to services and opportunities that promotes "longer life expectancies, less crime, stronger civic engagement and more robust economic vitality" [1].

There is strong evidence that software does foster and maintain this "positive condition" by, to name a few avenues, fuelling economic growth, easing education, enabling contact between like-minded individuals and geographically distant family members and friends. However, there is also substantial evidence of the negative effects associated with software, such as cyber bullying [4], theft of intellectual property [5] and financial assets, the spread of online child abuse [6] and loss of privacy [7] to name a few. There are also indirect losses resulting from cybercrimes, such as lost business opportunities due to banks inability to communicate with customers via email [8]. Thus, in order to support social sustainability in the long run, we must ensure that software is engineered in such a way that its negative effects are countered, or, at least minimised. This, however, is an open challenge.

This thesis aims to demonstrate how social sustainability can be engineered into software systems.

1.2 Research Problem

Because the functionality, constraints and properties of a software system are set through requirements engineering (RE), RE is also the key stage for engineering sustainability into software [9]. Nevertheless, requirements engineers have not yet incorporated social sustainability requirements into software systems engineering [10].

One of the reasons for this is that requirements engineers do not have clear guidelines for what constitutes a "positive impact on communities", and how it can be identified, modelled, or measured [10] (except when the functionality of the software is aimed explicitly at supporting the disadvantaged users, e.g., speech synthesising software to support those with severe speech impediments [11]).

The problem of software's impact on its user communities has been faced primarily within the organisational context, for a long time [12–16] and has, to some degree, been addressed via such techniques as value-based design [17–19], participatory design [20–22], user experience evaluation [23–25], and even iterative agile development [26]. Essentially, all of these techniques allow for explicit [18] or implicit [20, 26] incorporation of user values into the intended software systems. In this thesis we maintain that the positive contribution of software systems to "user communities" (i.e., its positive contribution to organisational and social structure) manifests when the given software system supports and promotes the social values of the given user community.

However, all of the previously noted techniques [17–26] set out to discover 'from scratch' the values and their respective requirements that would improve software acceptance in a community or organisation, or would increase user satisfaction with it. While we agree that each organisation and sub-community will have its own diverse cultural and traditional values with respective requirements, we also note that there are certain central values (such as equality, security and freedom) that are recognised as fundamental to modern societal well-being. These remain relatively stable, although they may have a distinctive flavour in various types of communities (e.g., gender equality in Western society vs. Eastern one). Such values are often universally recognised (e.g., the Human Rights Convention [27, 28]) and even standardised (e.g., International Standard on Social Accountability [29]). In this thesis we propose that such fundamental values should become the starting point for engineering social sustainability requirements for software.

Previous work on RE [30] has demonstrated that incomplete and hidden requirements are one of the main causes of failure in software projects, which is often exasperated by a lack of experience and/or domain knowledge within the RE team [31]. To help with both of these issues, the software engineering community has long observed the utility of extracting empirically proven "good practice" and expert knowledge into reusable patterns [32–37]. Thus, this thesis aims to develop a method to support the integration of social sustainability requirements into the software system requirements specification such that it supports the following characteristics:

1. The method will support operationalisation of the notion of social sustainability into software requirements;
2. It will facilitate reuse of requirements without necessitating full-scale user engagement for "from scratch" elicitation of the social sustainability concerns for each new project;
3. It will be suitable for use by both novice and experienced requirements engineers;
4. It will be usable with different RE practices and processes, without imposing a specific RE process (e.g., from waterfall to agile).

1.3 Research Objectives

The goal of this research is to develop a method for integrating social sustainability concerns into software development in a way that helps software engineers to derive the software's social sustainability requirements. It takes into consideration the commonly shared societal values that frame social sustainability. The motivation is that achieving social sustainability should be an ongoing process that starts from the early stages of the software; i.e. requirements engineering.

The work proposed in this thesis guides software practitioners in their approach to social sustainability. The objective of this work is to develop a method that enables requirements engineers to integrate (otherwise implicit) social sustainability requirements into software systems specification. In order to support the ongoing RE practices within the various software engineering organisations, this method should be easy to integrate with an arbitrary RE practices, facilitate reusability and be amenable to use by both novice and experienced practitioners. In addition, the proposed method needs to be applicable to various social sustainability aspects.

1.4 Thesis Contributions

The key contributions of this thesis are in:

1. Presenting a generic requirements identification methodology that is based on core societal values and, thus, is clearly directed towards a social sustainability objective. This methodology is abstract and can be instantiated to support social sustainability requirements in various domains.
2. Demonstrating an instantiation of this methodology to support equality-related requirements. In doing so, this thesis has delivered:
    (a) a pattern for equality concern representation, as well as for its requirements identification and elicitation;

(b) a requirements elicitation template that supports use of the equality pattern;

(c) an elicitation method using the equality value pattern.

(d) a detailed guideline on the usage of the elicitation method.

(e) a set of evaluation studies that reflect on the utility and usefulness of the proposed pattern and method.

1.5 Research Method and Outline

The methodology for social sustainability requirements identification proposed by this thesis is depicted in Figure 1.1.

1. It starts by applying qualitative data analysis of the social sustainability literature for a particular social sustainability aspect (e.g. equality, health, security, and others, as discussed in Chapter 4) (marked as social value literature in Figure 1.1) to understand the underlying meaning of a social value and relate it to software. This helps to understand the components or sub-values that contribute to the social value. This can then be presented in a value pattern. The value pattern is then associated with a template(s) that allows instantiation of the pattern in several software cases (marked as social value pattern and template in Figure 1.1). By the end of this step, a value pattern and template(s) is produced. This step operationalises the social sustainability concepts into software requirements. The current thesis demonstrates this process for the equality aspect of social sustainability, as discussed in Chapter 4. The value pattern and templates for social sustainability requirements is exemplified by an instantiation for the equality requirements domain. They are domain-independent patterns.

2. Then, an elicitation method is developed based on the equality pattern as well as usage guidelines in Chapter 5.

3. Next, the value pattern and templates are validated against several software specifications documents (marked as requirements documentations in Figure 1.1) to test its applicability and to identify common social sustainability requirements contributing to the value under investigation. Thus, the value patterns are related to a number of specific requirements, reusable across various application domains. We demonstrate this process for the equality value pattern in Chapter 6.

4. Then, the resulting value pattern and templates as well as the requirements (marked as social requirements in Figure 1.1) are evaluated. Evaluation involves software users as well as pattern and templates users. This allows the generalisability of the specific social sustainability aspects to be checked across software user and developer communities. For instance, in this thesis we demonstrate that the derived equality requirements are meaningful across user communities world-wide (as discussed in Chapter 7), to the requirements engineering professionals (as discussed in Chapter 8).

The thesis is structured into 9 chapters as follows (also see Figure 1.2):

Chapter 2 presents a literature review of social sustainability. The chapter begins with a literature review with the focus on sustainability models and frameworks in ICT. The challenge addressed in this chapter is the lack of generic, operationalisable and reusable view of social sustainability. To address this challenge, a systematic literature review was conducted. The key contribution of this chapter is in the 12 social sustainability indicators i.e. employment, health, equality, education, security, social cohesion, services and facilities, resilience, human rights, social acceptance. cultural and political factors.

Chapter 3 presents a literature review of value research as the resulted indicators in the previous chapter correspond to human values. The review confirmed that there is lack of generic, operationalisable and reusable view of social sustainability.

Chapter 4 introduces the proposed social sustainability requirements identification method and the resulting pattern and template. This contribution is a result of a qualitative analysis that was conducted to address the lack of a social sustainability requirements identification method. This is demonstrated as part A of the proposed methodology in Figure 1.1. The chapter details the value pattern construction method conducted and the resulting pattern and templates.

Chapter 5 presents an equality requirements identification method and guideline to assist in using the method. In addition, a relationship model depicting dependencies between equality and software quality attributes is discussed as well as a model for stakeholders role in equality. This chapter is an element in part A of the proposed methodology in Figure 1.1.

Chapter 6 presents the implementation of the equality requirements elicitation method (using the value pattern) on seven case examples and discusses the lessons that have been learned. The chapter addressed the challenge of the utility, reusability and applicability of the identification method with existing RE practices and domains. Additionally, the chapter presents experts evaluation of the resulting requirements using a small scale survey. The results reveals the usefulness of the elicitation method in identifying social sustainability (i.e equality) requirements. Furthermore, the chapter presents the application of the value pattern and templates in Agile software engineering[1]. This chapter presents part B of the proposed methodology in Figure 1.1

[1]This study is a independent study by Monica Bahl [234].

Chapter 7 details the evaluation of the requirements resulting from the value pattern. The challenge in this chapter was to prove the generalisability of the specific social sustainability aspects (i.e. equality) across software's user communities. The challenge is addressed by a user survey. 155 responses were analysed using frequency and inferential statistical analysis. The results indicate that there is an agreement (albeit not unanimous) among the respondents on equality supporting requirements. This chapter presents part C of the proposed methodology in Figure 1.1.

Chapter 8 demonstrates the requirements experts' evaluation of the social sustainability value pattern by using it in verbal protocol activity. There were 13 participants with academic and industrial backgrounds. The verbal protocols were analysed using qualitative analysis. The equality pattern and template were found useful and assist in identifying equality requirements. In this chapter, the challenge of the usability of the identification method regardless of practitioners expertise was addressed. This chapter presents part C of the proposed methodology in Figure 1.1.

Chapter 9 reviews the work presented in this thesis, discusses the conclusions drawn from the research as well as the relevant future work considerations and suggestions.

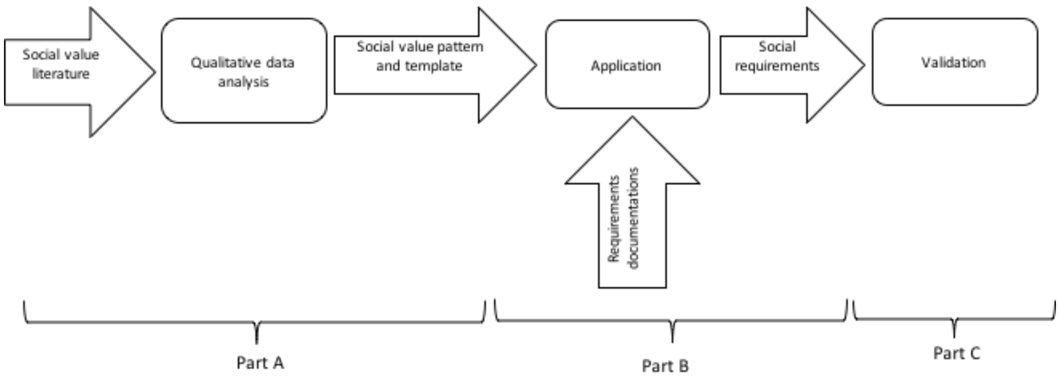


Figure 1.1: Social sustainability integration methodology

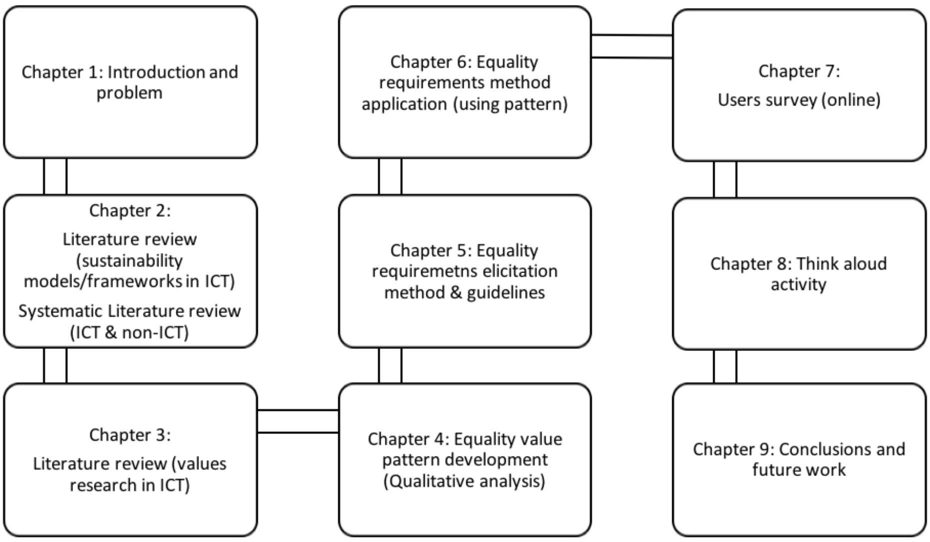


Figure 1.2: Thesis contents

Chapter 2

Literature Review on Social Sustainability

As the present work aims to provide support for engineering social sustainability requirements into software, we first turn to the current work on social sustainability within Requirements Engineering (RE). This reviwe consists of two parts: a review of the way that social sustainability is reported upon the RE/SE literature, followed by a detailed systematic study of the way that social sustainability concerns are structured and measured.

Section 2.1 presents a survey on the available sustainability models and frameworks in ICT against a set of criteria to evaluate their ability to establish a general, holistic and reusable way of integrating social sustainability concerns into software requirements. This is done using traditional literature review were the author searched through literature published in most related areas and venues [38] (such as conferences, journals, and workshops on requirements, systems, and software engineering).

In addition, using the snowballing technique, citations of the initially identified relevant papers were examined to identify further references to relevant publications [39].

As discussed in Section 2.1 below, we note that all studies so far have failed to establish a general, holistic and reusable framework for treatment of social sustainability concerns, instead each has taken a particular, incomplete view on it, and often delivered solutions targeted to a single community or problem solution. This motivated our work of integrating social sustainability issues into software requirements using patterns and templates.

The results provoked the need to undertake a systematic literature review to define a clear structure for social sustainability concerns. This study is presented in Section 2.2.

2.1 Social Sustainability

Social sustainability was defined as:

- Maintaining social capital (investments and services). Maintaining the social capital reduces "the cost of working together and facilitates cooperation" [40]. Goodland remarked on the importance of having shared rules, laws and information to promote social sustainability [40].
- "A positive and long-term condition within communities and a process within communities that can achieve and maintain that condition" [2, 3]. Willis, Mckenzie and Harris in [2, 3] highlighted several features of social sustainability. The features include equity in access to essential services such as health,

diversity and disparate culture integration within communities, citizens' political participation, fulfilling community needs by community action and/or political mechanisms (e.g. laws), "mechanisms for a community to identify collectively its strengths and needs", equity between generations (future vs current generations), social sustainability awareness systems to ensure awareness transmission between generations and "a sense of community responsibility for maintaining that system of transmission" [2, 3].

To support social sustainability, any proposed technique or methodology needs to meet several characteristics. The following are characteristics of the solution that we consider to be relevant for a methodology on social sustainability requirements engineering. These characteristics are: support for social sustainability requirements operationalisation and re-usability, present a generic view of a social sustainability aspect, usability irrespective of the practitioners experience and with different RE practices.

1. C1: The approach assists operationalisation of the concept of social sustainability and its aspects into software requirements. Requirements are functional and non-functional requirements. A functional requirement is a statement that decribes a system's functions, features or actions [36, 41]. A non-functional requirement is a statement that describes a constraint or a property that may affect software acceptance such as performance and security [36, 41]. In many ocations, non-functional requirements are viewed in the lights of the ISO/IEC 25010 standard [42]. The standard addresses software quality models in use as well as in product quality [42]. Quality in use model identified characteristics of effectiveness, efficiency, satisfaction, freedom from risk and context coverage [42]. The product quality characteristics includes functional suitability, performance efficiency, compatibility, usability, reliability, security, maintainability

and portability. Each characteristic represents non-functional requirements. Sommerville classified non-functional requirements into product requirements (such as efficiency, dependability and security), organizational requirements (such as environmental, development and operational) and external requirements (such as regulatory and ethical) [41].

The **criteria for the operationalisation** of the social sustainability concern is motivated by the fact that that incomplete operationalisation (i.e., incomplete representation of the relevant notions that make up social sustainability within the requirements for a given system) will lead to missing requirements which, in turn, will either result in an unacceptable solution, or necessitate additional changes and unnecessary re-work, leading to extra monetary and time costs.

It was reported that incomplete and/or hidden requirements are problems in RE that are relevant to project failure [30]. The problem can also affect customer satisfaction and product acceptance [30].

Here we use three levels of evaluation: full, partially and no (when social is not viewed as contributing to technical requirements). For sustainability definitions, please refer to Appendix A.

2. C2: The approach facilitates re-usability by all/or some of its parts. The **reusability requirement** is motivated by time and effort saving through reuse (simply by not repeating the task which has been completed once), as well as by the increased quality, reliability and understandability of the reusable components, because each time they are reused, they are likely to be checked and as where errors are discovered, corrected.

   Liu, Li and Peng reported the results of a survey conducted in China involving RE practitioners [31]. The survey reported that the third-most common requirements elicitation technique used is by referring to similar existing systems

[31]. Also, time restrictions imposed on the RE practitioner can be a cause of project failure [31].

According to Zave [43], RE represents an aspect of software engineering that focusses on 'real-word goals' to identify the functions and constraints of software systems. It also focusses on the relationship between the functions and constraints associated with software behaviour, and "their evolution over time and across software families" [43]. Due to the evolution of software, RE activities need to be reusable when developing similar systems [43].

Nuseibeh and Easterbrook have pointed that Zave's definition of RE marked the "changing world and the need to reuse partial specifications, as engineers often do in other branches of engineering" [44].

This provides an insight into the importance of having reusable requirements and requirements elicitation techniques that can be tailored to fit different software systems.

Three levels of evaluation are used here; full, partial and no. Full support can be achieved by producing reusable components (i.e. standards/common practice guidelines, patterns, templates). Partial support refers to using reusable components. No re-usability is when the approach does not produce or use standards/common practice guidelines or patterns or templates. Full support of re-usability implies that less effort is needed for operationalisation. It will also imply that only the part of operationalisation is repeated per project and not the process of discovering what a sustainability aspect might mean.

3. C3: The approach provide a generic and comprehensive view of a social sustainability aspect. This is facilitated by *representing a view of a single social aspect (learned from the sustainability literature).* In turn, this allows unified instantiation of the operationalisation part of the approach per software. The **generic view** supports the instantiation and elicitation of social sustainability

requirements across different software project. This criteria is borrowed from the pattern development process (see Appendix A) which requires building a full and intensive view of the issue under investigation and the solution to it. In addition, studies reported one of the challenges in sustainable development is resulting from the inconsistent sustainability perceptions held by designers [45]. Thus, the study recommended introducing sustainability concepts to build a "common understanding and definition of sustainability" prior to the commence of sustainable design process [45]. According to Shapira, Ketchie, and Nehe, sustainability concepts will also highlight requirements of sustainability without restricting creativity and freedom [45]. *This provides an insight to the importance of the* ***generic view*** *criteria in sustainability frameworks and models.* This criteria has two levels of evaluation. Yes, is used when a definition/model is provided to describe and present the issue and it can be used during operationalisation of social sustainability into software requirements. Else, no is used as the other evaluation level.

4. C4: Used by software engineers and practitioners regardless of their level of expertise or way of approximating social sustainability [46]. **Usability of the solutions irrespective of the experience** of the requirements engineer ensures that even novice engineers will be able to adequately address social sustainability concerns. As noted in related work [30, 31, 47], a lack of experience has been shown to lead to poor quality work products, where the work process is found to be difficult for the novices. It was reported that lack of experience and the weak qualifications of the RE team are among the common causes of RE problems [30].

   Here, two levels of evaluation are used, yes (for different skills levels support) and no (when some specific skills are required/can restrict usage).

5. C5: Used with different RE practices [46].

This is motivated by the current landscape of RE within software engineering, where some organisations use formal and heavyweight RE processes (such as waterfall) and others integrate RE into the development process, with hardly any distinguishable "RE stage" (such as agile processes). As social sustainability is a key issue for software success, all of these diverse practices should be able to tackle it through the proposed generic methodology, irrespective of their own engineering practices differences.

The importance of integrating new requirements models/methods with the current RE practices was discussed as an important factor of the acceptance and usage of a newly proposed methods/models [47–50]. In [49], the restricted time and budget were discussed as factors hindering organisations from adopting new methods as they prefer to use what is already know to them. Morris, Masera and Wilikens reported that one of the reasons behind new requirements engineering methods adoption problems is the integration of the new method within the business practice [48]. RE practitioners mentioned that adopting new methods could bring issues of the need of new ways of thinking, working and communicating [48]. This in turn affects the willingness to embrace the new methods [48].

Here, two levels of evaluation are used, yes (for usability with different RE practices) and no (when some restrictions are imposed).

Section 2.1.1 presents the analysis of sustainability models using the five mentioned characteristics.

2.1.1 Sustainability Models and Frameworks in ICT

Dick, Naumann and Kuhn [51] proposed a model for "Green and Sustainable Software Engineering". This model comprised a process/lifecycle model and guidelines/checklists. The process and lifecycle points to the impacts of software product usage in each of its life cycle phases on sustainability. This is done by looking at the activities in each phase and how they affect sustainability. For example, reflection and assessment meetings between the members of the development team are part of the development phase. Those activities are viewed in terms of the related environmental impacts. As a consequence, the team decides to avoid business trips for the team meetings to increase resources efficiency.

The guidelines and checklist part "provide[s] tips and helpful hints on how to develop, use, provide, and maintain software products in a sustainable way. Therefore, the guidelines and checklists are aligned to the activities and product scenarios of the lifecycle model". An example of how the model contributed to social sustainability was explained by considering end users and recommending the use of the "Generating User-specific Interactive Documents (GUIDO)" system. GUIDO is a document database in which all documents or forms (governmental) are kept in one repository [52]. The documents are displayed to a user based on their preferences as well as their capabilities. This is achieved by having user profiles that are used to define the suitable display for a user [52]. Doing so is expected to support impaired people who are allowed to participate in the process of society related decision making process without requiring assistance [51]. This model emerged later to the GREENSOFT Model in [53].

Naumann and colleagues defined a sustainable software as a software that is built to meet sustainability objectives [53]. Moreover, sustainable software is a software with curtailed negative usage effects; i.e. social and environmental "(first order effects)" (See Appendix A for order effects). Additionally, a sustainable software is a software that reflects sustainable development in its functionalities "or at least has no negative impacts on the society or environment (second-order and systemic effects)" [53]. Naumann and colleagues [53, 54] propose a GREENSOFT Model that catalogues sustainable software and sustainable software engineering. The model took advantage of the life cycle assessment (LCA) and built the model with this insight [54]. The four components of the reference model include the life cycle of software development, sustainable criteria and metrics, procedure models and recommendations and tools.

The life cycle component looks into the software product phases from development to disposal and recycling [54]. For each phase, ecological, social, human, and economic compatibility are assessed [54]. The results of the assessments are used to optimise the product or compare it to other competing products [54]. For example, at the end of life (deactivation) phase, an organisation's economic sustainability can be affected by the ability to access data that is stored in an old format that should be converted easily [54].

The sustainable criteria and metrics component comprises measures of software quality and sustainability criteria and metrics classified as directlty and indirectly related [54]. For example, 'usability' is a criteria for the usage phase and the 'project's footprint' (as "natural resources and environmental impact used during software development" [55]) is related to the development phase [54]. The procedure models component looks into four software procedures; develope, purchase, administer, and use [54]. In a sustainable software purchasing procedure, for example, selecting bidders can be done based on their commitment to social and environmental responsibilities [54]. Another example would be selecting the software product to be acquired based on energy consumption [54]. The administrate procedure is related to software configuration, installation and maintenance [54]. To ensure a sustainable administrate procedure, an organisation's IT team needs to continually evaluate energy consumption and resource consumption. The use procedure is related to software users where they reflect on the effects of their usage on sustainability. This will allow them to look for possible mitigation measures and guidelines.

The recommendations and tools component supports software stakeholders in applying green or sustainable techniques. An examples of a recommendations for developers is the energy-efficient software guidlines [56]. An example of a tool available to users is the green tracker [57].

Additionally, a quality model for sustainable software was presented in [53]. The model suggests common criteria for sustainable software (component of the reference model). The common criteria are based on ISO/IEC 25000:2005. Social aspects were categorised as criteria directly related to sustainability. The social criteria relates to accessibility, usability and organisation conditions. Naumann et al. [53] said that:

> *Accessibility* indicates whether the software product can be used by as many people as possible without any restrictions. *Usability* covers the understandability, learnability and operability of a software product, i.e. if the software usage is user-friendly and easily learned. The aspect *Organization Sustainability* covers the social situation in the software company including working conditions, for example.

Although the model suggested accessibility, usability and organisation sustainability, the GREENSOFT model is a general model that provides general references to possible concepts such as accessibility. It does not help with specifically identifying software requirements and features that supports social sustainability. More evaluation points are illustrated in Table 2.1.

Table 2.1: Evaluation Characteristics in Sustainability (Naumann et al)

Literature	Characteristics	Category	Why?
Naumann et al. [53], [54]	C1: Operationalised requirements	Partially	The model translated social sustainability into accessibility and usability as general concepts. Yet, what functions or features are to be implemented in software system is not given.
	C2: Re-usable	Partial	Encourage using available recommendations such as energy-efficient software guidelines
	C3: Generalisable concepts	No	No definition provided that can be used in operationalisation
	C4: Expertise	Yes	The model is supposed to be used by different stakeholders with different skills levels. As social sustainability is approximated to general requirements concepts, this suggests that software practitioners will be able to tailor those concepts to their level of expertise.
	C5: Used with different RE practices	Yes	As it is a conceptual model that provided abstract meaning of social sustainability (accessibility and usability), then we say it will be usable with any RE practice.

Penzenstadler, Mehrabi and Richardson [58] contemplated sustainability effects on requirements engineering processes using a requirements engineering approach for sustainability RE4S [59]. The approach integrates sustainability concerns into the available requirements engineering approaches and methods [59]. As part of this approach, the stakeholder model, goal model, a system vision and a usage model are prepared. Although Penzenstadler, Mehrabi and Richardson stated that social sustainability was indirectly tackled, they mentioned that "[a] decrease in the need to drive to the doctor's office will reduce traffic on the roads, allowing other drivers to get to destinations more quickly (2nd order effect)" [58]. In addition, Penzenstadler, Mehrabi and Richardson identified the technical sustainability of the system as its compatibility with different platforms and devices. However, providing a list of non-functional requirements (in the technical report [60]) Penzenstadler, Mehrabi and Richardson propose that sustainability requirements are non-functional requirements. This raises a question on regarding whether we could find functions or features relating to sustainability. More analysis details are available in Table 2.2.

Table 2.2: Evaluation Characteristics in Sustainability (Penzenstadler, Mehrabi and Richardson

Literature	Characteristics	Category	Why?
Penzenstadler, Mehrabi and Richardson [58]	C1: Operationalised requirements	Partially	Only non-functional requirements. Also, technical sustainability is not related to social sustainability.
	C2: Re-usable	No	No patterns, templates or standards.
	C3: Generic concepts	No	No definition was provided.
	C4: Expertise	No	Engineers should be familiar with stakeholder model, goal model, system vision and a usage models
	C5: Used with different RE practices	Yes	No restrictions imposed by the model

Lago et al. [61] proposed a framework for sustainability software-quality requirements. The framework focused on environmental sustainability and its relationship to the other sustainability dimensions. In the the framework:

> Social sustainability focuses on ensuring current and future generations have the same or greater access to social resources by pursuing generational equity. For software-intensive systems, it encompasses the direct support of social communities in any domain, as well as activities or processes that indirectly create benefits for social communities. [61]

The framework was applied to two case studies, one of which was a paper-mill control system. Two social quality requirements were found to be related: employment and education. For each requirement, evaluation criteria were also incorporated. For employment, the number of highly specialised employees, the total number employees, total number of indirectly engaged employees, level of engagement in production and level of engagement in sustainability were identified as evaluation criteria. Specialised competencies, education programmes, the calculated education gap and level of engagement with education institutes were the evaluation criteria associated with education. The other case study was a car-sharing platform and two social quality requirements were identified: public acceptance of service and car sharing community acceptance. Evaluation criteria for the first quality were the number of users, number of cars, average usage/users and average usage/car. For car sharing community acceptance, customer satisfaction and customer surveys were found to be related. The framework enabled relationships to be found between qualities in the same domain or even in different domains such as car sharing community acceptance (social) and profits from users (economic) [61]. The framework provides a basis for sustainability metrics [61]. However, those metrics are related to the project and not identifying the software's functions/features that contribute to sustainability. More analysis can be found in Table 2.3.

TABLE 2.3: Evaluation Characteristics in Sustainability (Lago et al.)

Literature	Characteristics	Category	Why?
Lago et al. [61]	C1: Operationalised requirements	No	Technical requirements (software) were viewed as a separate dimension. This is similar to Penzenstadler and Femmer [62, 63]. Thus, social requirements are not specific to be implemented in the software/more of the environment surrounding the software system.
	C2: Re-usable	Full	The model is extension of ISO/IEC 42030 Architecture Evaluation [1]
	C3: Generic concepts	No	No general definition relating social sustainability to software requirements.
	C4: Expertise	Yes	No specific RE expertise is imposed.
	C5: Used with different RE practices	Yes	It is an assessment framework.

Roher and Richardson suggested the use of sustainability requirements patterns (SRPs) to guide software engineers when writing environmental sustainability requirements [64]. Their work was based on Withall's software requirement patterns [65]. According to Roher and Richardson, "[s]ustainability requirements may be used to specify system behavior (e.g. requirements that will reduce a systems energy consumption) as well as to influence the users behavio[u]r (e.g. the system incentivi[s]es sustainable actions)" [64]. The proposed SRPs are presented in a template that includes summary, applicability, content, archetype, examples, discussion and related patterns sections. An example of the identifed environmental SRPs is called "Incentivi[s]ing Minimal Resource Consumption". A resulting requirement from this pattern is, for example:"[t]he system shall display the average and minimum amounts for each type of resource consumed by previous users" [64]. This requirement is derived from a generalised requirement: "[t]he system shall communicate to the user as much information about resource consumption as possible", which is used as a template to derive specific requirements for the system under study.

Roher and Richardson's suggested patterns are based on requirements documentations of the same software application; i.e. a hotel system in which environmental sustainability was a concern. However, this means that the environmental sustainability issues were based on the view of requirements engineers. This view might be limited to the requirements engineers' understanding of environmental sustainability and their ability to identify the applicability of the environmental concerns in the software domain. In addition, the suggested patterns are limited to hotel management applications. This could limit the generalisability of the addressed sustainability concepts. Further analysis is available in Table 2.4. The SRPs were then used in a recommender system for eliciting software sustainability requirements [66].

Mahaux, Heymans and Saval [67] were interested in "how to discover requirements that help minimi[s]e the negative environmental impacts of (the activities supported by) the software under construction". Mahaux et al. also aimed to explore how the currently existing tools supports the discovery of possible minimisation of the negative environmental impacts. The focus in this paper was on environmental sustainability and how it is affected by the software. First, Mahaux et al. started

Table 2.4: Evaluation Characteristics in Sustainability (Roher and Richardson)

Literature	Characteristics	Category	Why?
Roher and Richardson [64]	C1: Operationalised requirements	Partial	Templates facilitated operationalsing environmental sustainability into software requirements. The given examples are functional requirements.
	C2: Re-usable	Full	Produced patterns
	C3: Generic concepts	No	The patterns were build on existing requirements documentations of hotel system only. No sustainability literature was used. It might have missed the underlying meanings of sustainability from the sustainability side.
	C4: Expertise	Yes	No restrictions imposed.
	C65: Used with different RE practices	Yes	No restrictions imposed.

with stakeholders identification and noted the need for environmental specialist to be part of the discovering process. Then they moved to soft scoping and context discovery to get a rich picture on the software. This was followed with harder scoping which included the use of case analysis and misuse cases analysis to highlight environmental risks and mitigation. This resulted in a set of requirements. To prioritise the resulting concerns, they used a generic environmental sustainability goal taxonomy (to prioritise concerns and refinements) that is independent of the case study and can be reused. The goal taxonomy is based on the idea of Cabot et. al in [68] (see A.6). Authors identified "sustainability-related system functionalities" [67]. As they progressed in the study, Mahaux, Heymans and Saval felt that there was no need to have a specific technique for sustainability requirements and instead tailoring the existing ones would be sufficient [67]. Mahaux et al. also added that: "it would be necessary for requirements engineers to acquire a minimum of expertise in environmental analysis, and probably that some toolkit could help the requirements engineer take sustainability-related decisions" [67]. The question is what if we don't have sustainability officers (social, environmental or economic) as part of the requirements engineering teams? What if the available requirements engineers have limited knowledge of sustainability? Will this process be easy for them? More details are available in Table 2.5.

Table 2.5: Evaluation Characteristics in Sustainability (Mahaux, Heymans and Saval)

Literature	Characteristics	Category	Why?
Mahaux, Heymans and Saval [67]	C1: Operationalised requirements	Partial	Environmental sustainability concerns were translated to software functionalities.
	C2: Re-usable	Partial	Goal taxonomy used for prioritising requirements can facilitate re-usability
	C3: Generic concepts	No	No definition was provided.
	C4: Expertise	No	Prioritization through goal modelling be restriction. Also using use cases and misuse-cases.
	C5: Used with different RE practices	No	Prioritization through goal modelling might be restriction. Also using use cases and misuse-cases.

Albertao et al. [55] discussed sustainability performance metrics for software development projects. The metrics covered the three sustainability domains; economic, environment and social. A set of process-, usage- and development-related attributes (e.g. usability, accessibility, efficiency and others) are to be analysed. They believe that each of these attributes contribute to economic, environment and social improvements and, in turn, to sustainability. An example of metrics related to accessibility is internationalisation and localisation support. Analysis of this work based on the characteristics in 2.1 is illustrated in Table 2.6.

Table 2.6: Evaluation Characteristics in Sustainability (Albertao et al.)

Literature	Characteristics	Category	Why?
Albertao et al. [55]	C1: Operationalised requirements	Partially	The metrics and quality properties provides insights on non-functional requirements.
	C2: Re-usable	Full	Extended the list of quality attributes by adding how attribute improvement contribute to sustainability.
	C3: Generic concepts	No	No definition to be used in operationalisation.
	C4: Expertise	NA	it is mainly assessment after software release
	C5: Used with different RE practices	Yes	Assessment does not require specific RE practises

Penzenstadler and Femmer presented a generic sustainability model that can be employed for the analysis and assessment of environmental sustainability [62, 63]. The model incorporated sustainability concerns with goal modelling and human values. The model has three sub-levels: the meta model (M0 level), the reference model (M1 level) and the instances (M2 level).

In the meta model, the **goal** has aspect(s) represented as **dimension(s)**. A dimension is represented by **value(s)**. **Activity** elements contribute to values and **indicators**. Indicators are either quantitative or qualitative metrics that approximate values. **Regulation**, optional element, supports values and affects indicators.

The reference model (M1 level) has three levels. The top level incorporates the goal and its associated dimensions, whilst the middle level includes the values, indicators and regulations. The lower level consists of activities. Sustainability is the goal and its dimensions are represented as the dimensions in the reference model.

Penzenstadler and Femmer illustrated the social dimension of the model through community, tolerance, trust, fairness, peace and culture values which are further decomposed into sub-values. For example, the fairness value is represented by justice and equality sub-values. The equality value is supported by the activity of providing for equal chances. In turn, this activity is supported by another activity; i.e. conducting appraisal interviews.

Instantiation of the model (M2 level) is carried out during the requirements engineering process. At this stage, the model is tailored to fit a specific software project.

Conejero et al. [69] used the approach of Penzenstadler and Femmer to create sustainability model for an organisation aiming to utilise unmanned aerial vehicles (UAVs) to achieve sustainability goals; i.e. environmental, social, economic and technical. Conejero and colleagues found it to be a useful approach.

Penzenstadler and Femmer's model directly links sustainability with values. However, instantiation of the operationalisation of social values into software requirements/features is not clear. Further analysis is illustrated in Table 2.7.

Table 2.7: Evaluation Characteristics in Sustainability (Penzenstadler and Femmer)

Literature	Characteristics	Category	Why?
Penzenstadler and Femmer [62, 63]	C1: Operationalised requirements	No	The generic model provides general activities. But it does not say how to operationalise it. For example, the generic model of social sustainability listed the activity of "provid[ing] for equal chances". This is due to the view on dimensions of sustainability where technical requirements are not related to social sustainability.
	C2: Re-usable	Full	The generic model is reused with instantiated per project.
	C3: Generic concepts	No	No definition to guide operationalisation
	C4: Expertise	Yes	No restrictions imposed.
	C5: Used with different RE practices	Yes	No restrictions imposed.

Barn et al.'s model also linked sustainability to values. They inspected how software engineering practice can take the role of "agent of change" for societal sustainability through the manifestation of value sensitive concerns" [70]. Barn et al. proposed a conceptual model that integrates values into the software engineering process. The model includes co-design workshops that bring stakeholders together to create the design features and specifications. Co-design workshops are part of the co-design space of the conceptual model. The model is based on the work in [71] and was applied on a research project named "Mobile Apps for Youth Offending Teams MAYOT". Upon application in the project, the value identified was privacy and autonomy that led to "design and specification of features/functions" such as the exclusion zone feature which "is a function that is available on the MAYOT application that allows a case worker to define a geographic region from which a young person is prohibited". The exclusion zone feature represents the prompts in the conceptual model. The model tied values to social sustainability and expressed a way of involving values in software engineering practices. For analysis, please see Table 2.8.

TABLE 2.8: Evaluation Characteristics in Sustainability (Barn et al.)

Literature	Characteristics	Category	Why?
Barn et al. [70]	C1: Operationalised requirements	Partial	Only features
	C2: Re-usability	Yes	Conceptual model to handle values
	C3: Generic concepts	No	No definition to guide instantiation of operationalisation
	C4: Expertise	No	Need for experts to run and analyse Co-design activities results.
	C5: Used with different RE practices	No	Co-design activities are specified as part of the model.

Table 2.9 presents the overall analysis of the studies discussed above. It shows that there is a lack of generic, operationalisable, reusable view of social sustainability. In order to define software's social sustainability requirements, we need to have clear representation of what social sustainability is. Thus, we demonstrate a structured literature review next.

2.2 Systematic Literature Review

The objective of this study[2] is to understand what social sustainability is, how it is measured today, and what has been published with regards to the evaluation of software's social sustainability effects. By looking at what is considered to be an indicator of social sustainability and what is considered to be relevant for measuring and reporting on, we aim to uncover the structure of social sustainability concern, as reported in the reviewed scientific literature.

[2]This study was presented at the 3rd International Workshop on Requirements Engineering for Sustainable Systems in 2014 [72]

Table 2.9: Overall Analysis

Literature	Operationalistion	Re-usability	Generic view	Expertise	RE practices
Naumann et al. [53]	Partial	Partial	No	Yes	Yes
Penzenstadler, Mehrabi and Richardson [58]	Partial	No	No	No	Yes
Lago et al. [61]	No	Full	No	Yes	Yes
Roher and Richardson [64]	Partial	Full	No	Yes	Yes
Mahaux, Heymans and Saval [67]	Partial	Partial	No	No	No
Albertao et al. [55]	Partial	Full	No	NA	Yes
Penzenstadler and Femmer [62, 63]	No	Full	No	Yes	Yes
Barn et al. [70]	Partial	Yes	No	No	No

2.2.1 Literature Review Design

To investigate these issues, we formulated the following set of research questions:

RQ 1. What metrics are used for measuring social sustainability and how are they constructed?

This question aims to explore how social sustainability has been evaluated and which specific metrics are used to measure social sustainability. With this question, we aim to explore the broader literature relating to social sustainability issues, regardless of the area of application and irrespective of whether or not it is related to software development. This question also aims to review how social sustainability metrics are built and what their bases are.

RQ 2. What are social sustainability indicators?

This question aims to study the finer-grained constituents on which social sustainability metrics are built and the ways that these constituents are quantified. Furthermore, we will identify what are the common aspects of social sustainability are in each area. We will identify common dimensions/constituents used in various domains and how they are customised to adapt to a specific context or domain.

RQ 3. What is the role of software in social sustainability?
The intention here is to establish what the relationship is and use of software applications within the social sustainability domain. This question will be used to look at a set of issues including: What social sustainability areas of life and activities does software support and how? Why (if any) challenges related to social sustainability could be expected to be addressed via software?

RQ 4. What are the indicators of software's social sustainability?

The objective here is to study how software's social sustainability is assessed. We are interested in establishing the indicators related specifically to software applications. We are also looking at how similar or different these indicators are to indicators in other domains (e.g., agriculture, etc.). The reason for looking at indicators and metrics is to revel what is considered to be relevant and how it is operationalised into measurable (and so more detailed and specific) characteristics.

As sources for the structured literature review we used a number of digital libraries, namely ACM, IEEE, Scopus, Springer Link, Web of Science and the Applied Social Sciences Index & Abstracts (ASSIA). These libraries were chosen based on their subject coverage of both computer science and social sciences. ACM and IEEE cover computer science and engineering. Social sciences and engineering are covered by the Scopus and Springer link libraries. Web of science and ASSIA cover social sciences to obtain content on (computer-science domain independent) social sustainability.

To select the articles from the digital libraries, we used a combined search string extracted from the above discussed research questions to ensure that we obtain relevant results [73]. Although the combined search string (which we arrived at after an initial piloting of several search strings) was customised to each digital library, it always covered the topics of "Social Sustainability" AND (metrics OR indicators OR software). The results of the search and the initial screening for this study are shown in Table 2.10. Papers were excluded if there was no access to abstracts, they were not available in English or they were found to not be relevant to the research questions (i.e., did not address the topic of social sustainability or had no relation to indicators/metrics for social sustainability). One hundred fifty five (155) of accepted papers have then been studied. The screening process is depicted in Figure 2.1.

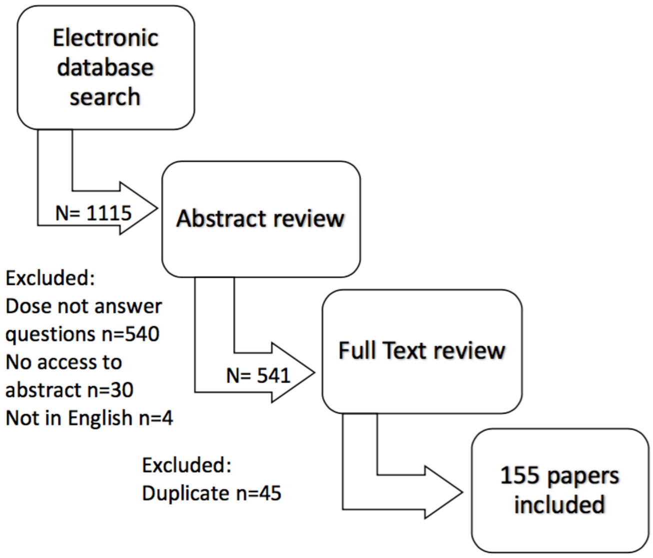


Figure 2.1: Screening Process

The following data were extracted from each studied article:

- General administration details (i.e.,: title, author(s), source, year)
- Social sustainability indicator

Table 2.10: Screening Results

Digital Library	No of results returned	No. of accepted	No. of duplicate removed	No. of papers included	No. of papers not included
IEEE Xplorer	63	45	0	28	17
Scopus	137	115	9	62	44
ASSIA	1	1	0	1	0
Web of Science	79	68	29	9	30
ACM	3	2	0	2	0
Springer Link	832	310	7	53	250
Total	1115	541	45	155	341

- Social sustainability metric
- How social sustainability is supported
- Type of study (e.g. case study, rigorous analysis, prototype)
- Context of study/domain

2.2.2 Literature Review Results

We have observed that the general set of indicators, metrics and domains has now been well stabilised. In other words, review of additional articles does not tend to significantly change/add to the current set of results. Thus, the review has reached data saturation and stopped at a total of 155 studies (See Appendix B for the full list). The current findings that address the stated research questions are presented below:

2.2.2.1 Construction of Metrics Used for Measuring Social Sustainability (RQ1)

Assessment Frameworks for Social Sustainability

The most commonly used framework for assessing social sustainability is the life cycle assessment (LCA). This is a "cradle-to-grave" method of evaluating the inputs, outputs and environmental impacts of a product during all phases of its life cycle [74]. An example includes land consumption and environmental emissions in the case of municipal waste management [75]. The LCA has been adapted to include such social concerns as labour force, communities' living standards, cultural heritage, freedom, health and safety, equity and poverty prevention [75–84]. A Social Impact Indicator (SII) has previously been applied in [85]. SII is based on LCA and is used to calculate social effects such as human resources and stakeholders' participation [85]. In [84], the LCA is merged with the Economic Input and Output analysis method (EIO) to form an economic input/output- based life cycle assessment (EIO-LCA). The EIO-LCA is used to quantify the direct and indirect sustainability impacts of U.S construction industries (e.g. indirect work injuries) [84].

Vulnerability assessment techniques (VATs) have been used [86] to assess the social impacts resulting from urban redevelopment projects. This was achieved by identifying the most vulnerable people and then assessing the social negative impacts affecting them [86]. This approach provides insights for policy makers into areas to consider reducing the negative social effect of the project [86]. Doloi presented a framework for social performance assessment of infrastructure projects based on Social Network Analysis (SNA) [87]. SNA was utilised to identify groups of stakeholders affected by the project (actors), their degree of influence (relationships between actors) and their specific social needs [87]. Then the groups' satisfaction of needs was measured and the project's social performance was derived [87].

It was proposed that Maslows Hierarchy of Needs should be combined with LCA to develop a social sustainability measure for organisational decisions [88]. Organisations can use a specified need to derive a social indicator from it [88]. For example, taking into account health as a basic need, an organisation considers improvments in the quality of food and health insurance policies as social indicators [88]. Companies aiming at morebetter social sustainability shouldl focus on meeting their employees's higher order needs (e.g. equity) while others will focus on satisfying the lower order needs such as food [88].

Metrics Construction Process

Based on the reviewed literature, we observe that the common way of constructing metrics or a methodology to assess social sustainability starts with the identification of general or domain-specific sustainability assessment guidelines that have already been published. For example, in [89], the researchers investigated the available higher education and campus sustainability assessments frameworks as a starting point for evaluating Malaysian campuses. Guidelines can be local or international. For instance, in [90] the researchers based their assessment on the International Hydropower Association (IHA) Sustainability Guidelines to evaluate the sustainability of a hydropower project in China. Moreover, established indexes/indicators of assessments (such as the Human development index and Wellbeing Index [91], Vanclay's definitional list of "social impacts" [83], Oregon Benchmarks [92] and European Commission indicators [93]) could be used as the basis to build assessment variables or to compare the assessment results against them[3].

Once the general guidelines have been chosen and complemented with domain-specific policies, the assessment methodology is then customised to fit a specific

[3]Although we cannot use the same methods directly, as these are constructed on bases of extensive country-wide surveys of such indicators as life expectancy at birth, mean years of schooling or gross national income per capita.

domain and case study. In order to do that, academics' and stakeholders' contributions are often involved. This is done through interviews, questionnaires or focus groups [77, 79, 86, 89, 91, 92, 94–104]. Stakeholders' participation also forms part of evaluating a project's sustainability [90]. In [102], for instance, the experts who took part in a customisation phase were selected based on their contribution to the research on future development of dairy farming. In [103, 104], stakeholders were involved in selecting or designing indicators for social themes because the available scientific information was limited or non-existent.

Once the indicators have been selected, metrics are constructed for them. For example, in [75] the social sustainability of municipal solid waste management system was evaluated by two indicators: damage to human health and income-based community well-being [75]. The damage to human health was calculated by summing the "factors for mortality (measured as years of life lost-YOLL), severe morbidity and morbidity (measured as years lived disabled-YLD)" [75]. The income-based well-being indicator was calculated using the potential employment opportunities for ith level (labour hrs/tonne), the rate of wages ($/hour) of ith level, the value of income generated from indirect activities ($/tonne) and the cost of living ($/person).

Another clearly emerging threat from the literature review is the current lack of trust towards the sustainability assessment metrics and methodologies. This, we believe, is caused by the relative immaturity of the field. Some publications propose to tackling this issue by "developing case study banks to translate experiences of using an indicator" [105]. This work also notes that such banks will help in "increasing criteria confidence and value usefulness to potential users . . . through case studies validation checks which can also assist with improving the indicators to meet a satisfactory degree of 'accuracy', and 'credibility' " [105]. This approach has, in fact, been used by a number of other researchers [85, 99, 103].

2.2.2.2 Social Sustainability Indicators (RQ2)

Social sustainability indicators should be relevant to the case under investigation. In [103], it was expressed that sustainability indicators need to satisfy certain criteria such as causality and sensitivity. These criteria are to ensure that the indicators are related to the monitored case and they respond to changes in the studied case [103].

Looking at the list of indicators and using the keywords and classifications that the authors provided, commonly used indicators were identified (regardless of the domain). The aggregated categories are:

1. Employment indicator: inspects a project's/product's impact on employment opportunities and job conditions. It comprises several sub-indicators related to employment statistics and job conditions [102]. Examples of measures used for this category are: (a) the number of employed women/ share of women in leading positions [96, 100, 106], (b) the number of full time/part time workers [107], (c) the utilisation of different working time arrangement [107], (d) compensation [108] and (e) job opportunities created [100].

2. Health indicator: covers the quality of health services provided to the people [100], health problems reported to authorities [108, 109], health risks [99] and health practices [108] in the community. Health indicators could be assessed by: (a) availability and access to drinking water [98], (b) child mortality rate [110], (c) percentage of workers with health benefits [81] and (d) voluntary health measures taken [100].

3. Equality (equality, equity, fairness) indicator: assesses the impact on the equal/fair treatment of people and opportunities. Examples include: (a) income/wealth distribution [90, 111], (b) social inclusion [1, 103], (c) diversity

of housing infrastructure [1], (d) provisions for basic needs of the disabled, elderly or children with proper access [95] and (e) fair competition [77].

4. Education indicator: evaluates impact on educational and knowledge opportunities and improvements. This can include: (a) the number of persons with higher education than secondary school/number of persons aged between 20-64 years [112], (b) employees educational level/literacy levels [1, 91, 99], (c) offered areas of employee training [100], (d) number of students per teacher [113] and (e) supporting educational Institutions [108] .

5. Security indicator: evaluates criminal conditions/status in a community. Examples include: (a) personal and property crime [1], (b) overall crime [92, 97], (c) vandalism [97, 114] and (d) juvenile arrests [92]

6. Social cohesion indicator: assess social ties and networks in the community. Some examples are: citizens' ability to walk to places in the local area such as shops and community facilities [104, 115], citizens empowerment by allowing initiations of community activities and voluntary work [1, 115] or decision-making [103, 109], networks [1, 103, 116, 117] and knowledge sharing [77, 99, 100, 118, 119], visible minorities, tolerance, identity [1, 103] and accountability and transparent decision-making processes [1].

7. Services and facilities indicator: focuses on the available services and facilities in communities such as schools [95, 115], health care services [95, 97, 115], sports facilities [95, 115], child care and housing [1].

8. Resilience indicator: analyses the community's ability to change after a disaster or conflict [1, 86, 95, 120].

9. Human rights indicator: inspects the impact on human rights against violations. Human rights can be related, for example, to discrimination, child labour or forced labour [77, 116].

10. Social acceptance of technology indicator: focuses on people's readiness to accept new technologies or projects. This can be observed through knowledge, perception and fear indicators [83]. Knowledge estimates what the level of public knowledge is about the technology, while perception will assesses what they think about it (positive or negative). Fear evaluates what issues/worries the community has about the technology [83].

11. Cultural indicator: looks into the preservation of cultural values and heritage. This can include respect of cultural heritage and local wisdom [77], respecting the customary rights of indigenous people [77], local heritage and listed buildings [116] and the protection of cultural heritage [90].

12. Political indicator: focuses on the level of respect for governmental and organisational laws as well as the trust in those laws [91].

To depict the relations between social sustainability metrics and indicators, we represent it using Figure 2.2.

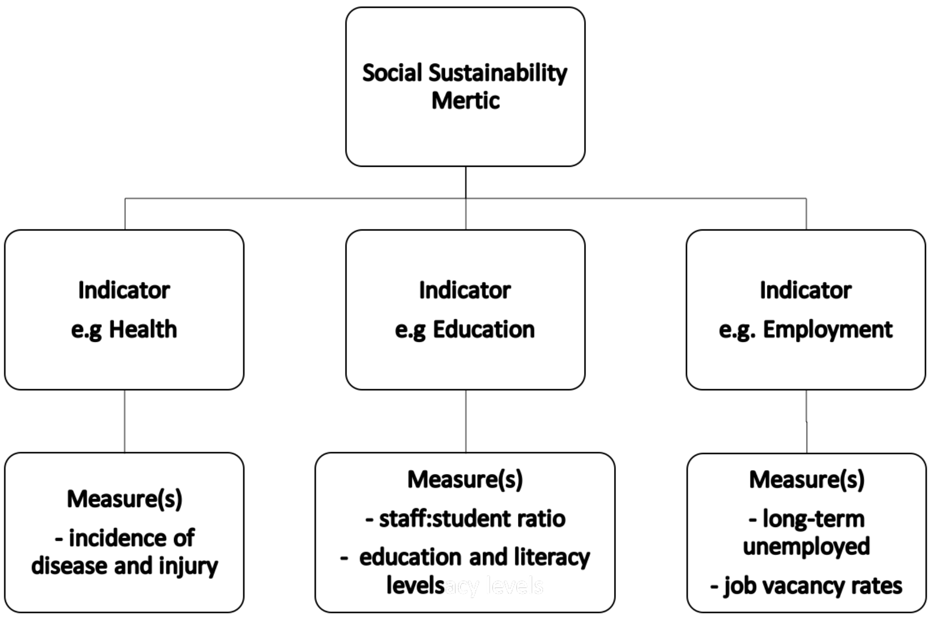


FIGURE 2.2: Social Sustainability Metrics and Indicators, examples are from [1]

Figure 2.3 depicts the twelve groups of indicators that reflect the structure of social sustainability. They integrate the reported relevant indicators from the reviewed literature.

FIGURE 2.3: Social Sustainability Indicators

Although we have summarised the social indicators into 12 cohesive categories, we must also note that the social sustainability indicators do not in fact always adhere to such a simple, flat hierarchy. In truth, they are often interchangeable and overlapping. We attribute this to the customisation during the metrics and methodology adaptation process (see Section 2.2.2.1) to suite the domain and the level of granularity relevant to a given case study. For example, employment can be used as an indicator by itself but can also be used as a sub-indicator for a community's equality.

Social indicators can also vary based on external and internal views of an organisation [85, 103, 108, 109, 121]. For instance, looking at a farm's social sustainability internally means relating social concerns to its employees and workers [103, 121]. External social sustainability involves assessing the community affected by the farm or the consumers of the farm's products. Meul et al. related animal welfare and health and landscape management to external sustainability [103].

By re-examining the 12 indicators in Figure 2.3, we recognise that indicators can be regrouped as two main categories; Community and Culture and Governance [72].

Community category (see Figure 2.4) is related to individuals and groups within a given society, their health, education, equality, etc [72]. This group of indicators can be categorised into human rights and community attribute indicators. Under human rights, equality, security and services and facilities (employment, health, education) indicators are grouped. The community attributes category includes social cohesion, social acceptance and resilience indicators. Culture and Governance indicators are concerned with cultural and political issues of a given society [72].

2.2.2.3 Role of Software in Social Sustainability (RQ3)

The articles related to the role of software in social sustainability suggest that software is often used to:

1. Promote social sustainability:
   For instance, in [122] a prototype of communication software is presented which is to be used as a communication enabler between virtual teams and a virtual organisation. The software is used to support social sustainability by enhancing the social networks.

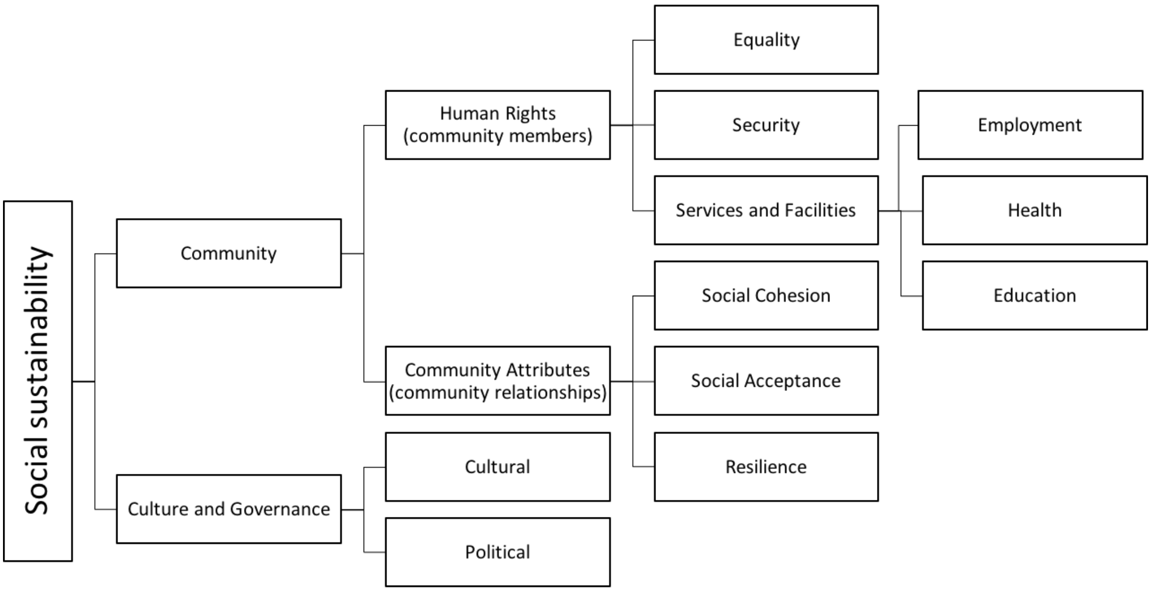


FIGURE 2.4: Social Sustainability Indicators-Revised

2. Design for social sustainability:

   For instance, in [100] software is utilised to provide guidance and reminders to researchers and managers when modelling a biotechnological product. The provided knowledge is about social sustainability issues to be taken into account when designing the product. This will help to support the decision-making process.

3. Educate about social sustainability:

   For instance, an educational game is used in a study to educate students about sustainability and social responsibility [117].

4. Assess social sustainability:

   For instance, Assefa and Frostel outlined a tool for assessing the ecological and economic sustainability of energy technologies [83]. They discuss the social indicators to be included in the tool.

We observe that our search on software and "social sustainability" resulted in fewer articles than expected. This is particularly surprising as we are well aware of a large body of research conducted in the area of human-computer interaction that focuses on various topics of social sustainability (such as stress, usability, loneliness etc. [123]). This issue indicates that although a concentrated effort has been underway for some time in the HCI community to address particular human-computer interaction issues, that work has not yet been consolidated under the umbrella of "social sustainability".

2.2.2.4 Indicators of Software's Social Sustainability (RQ4)

As for any other product, the social sustainability of software can be considered in its production, use, maintenance, and disposal stages. Below are the findings from the current study literature review on this topic:

1. Social sustainability at the production process is considered in [124], where it is suggested to use the "country of origin of a material and the manner in which it was produced (for example through child labour)" as a social sustainability indicator.

2. For software use, response time and scalability were used to evaluate software prototype that supports social networks and knowledge sharing between virtual teams [122]. This work also mentions that evaluating the prototype's performance includes evaluating "the degree of network congestion, the load on servers, the number of 3D objects to manage, and the complexity of the submitted query" [122]. They added that the database will support data availability in different contexts and data stability.

   Another work [119] provides "a theoretical basis for a multi-actor system as a simulation tool for social sustainability". Here software agents and humans simulate a social sustainability model [119]. For this the software agents must be "equipped with functions of perception, mobility, learning, communication, and coordination" [119]. The agents were proposed to simulate human individuals' and groups' behaviour related to knowledge generation, knowledge communication and knowledge use [119]. Knowledge, perception, learning, communication and coordination functions are all social sustainability indicators (discussed in Section 2.2.2.2)

At present we have not yet identified any work concerning the social sustainability of software maintenance and disposal. Similar to the comment in the preceding sub-section, we have observed that there are considerably fewer social sustainability indicators discussed for the software domain, compared to other domains (such as agriculture and supply chain management). This can be attributed to two factors:

i) On the one hand, the software effects on social sustainability are likely to have been studied for individual social sustainability characteristics (such as access to learning or other electronic resources, connectedness etc.), without aggregating these characteristics under the overall umbrella of social sustainability.

ii) On the other hand, the social effects of software products, once in use, are often indirect, take a long time to surface, and are difficult to discern. These effects are the so-called third-order impacts of ICT which "are long term indirect effects on the environment that result from ICT usage, like changing life styles that promote faster economic growth and, at worst, outweigh the formerly achieved savings (rebound effects)" [54].

2.2.3 Systematic Literature Review Remarks

This systematic literature review (commenced in Section 2.2) presents the social sustainability indicators and metrics and their relation to software. So far we have distilled a general social sustainability assessment framework and aggregated social sustainability indicators into 12 cohesive groups. A surprise finding of the review is that, in the 155 reviewed papers - taken from 6 digital libraries, software has been given hardly any consideration in relation to the concept of social sustainability. However, we are aware of significant work (most particularly in the HCI community) that has addressed a number of social sustainability features (such as usability, loneliness, etc.). This review has not been able to identify such relevant work because it is not explicitly related to the concept of social sustainability explicitly.

2.3 Threats to Validity

2.3.1 Construct Validity

Social sustainability is a new concept and term in Software and Requirements Engineering. Thus, past of this work is contributing to the better understanding of it. Our study purposefully did not restrict the paper search to requirements or software engineering domains, as the notion of social sustainability must be infused by concerns emerging from social and related sciences. However, this could lead to the treat of the emergent constructs not entirely relevant to the software domain. Furthermore, the granularity of the areas emerging from our study is somewhat uneven, for instance the cultural and political areas of the social sustainability are at rather high level, as the set of selected papers did not address these topics in detail.

The coding process could also have been impacted by the researcher's interpretations [125]. To enable critique, we submitted the initial version of the review for comments to a peer-reviewed workshop, which provided peer critique and validation feedback. Additionally, we conducted a cross validation of a sample of the coded work by a second reviewer (supervisor), with differences discussed and aligned before the full study was completed. However, calculation of Kappa test of inter-rater reliability [126] is not possible.

2.3.2 Internal Validity

Since the systematic literature review was conducted by a single coder, thereat to the internal validity is raised due to the possibility of researcher bias. To mitigate the internal validity of the study, we broadly followed Kitchenham's guidelines for systematic literature reviews [127], ensuring that a clear protocol was designed and applied, that exclusion and inclusion criteria for the paper selection were clearly stated and recorded. To provide a degree of validation, the protocol was initially piloted with a small number of papers, which resulted in a set of new categories defined, as well as inclusion/exclusion criteria update. In addition, a second reviewer periodically randomly sampled and cross-validated the coded work, ensuring that the process was consistently applied.

Nevertheless, because of the large number of papers the whole set was not second-checked, errors could have resulted due to the researcher's fatigue and absence of full cross-validation. Furthermore, as calculation of inter-rater reliability is not possible, internal validity cannot be clearly measured.

2.3.3 External Validity

External Validity could be threatened as the searches were limited to a set of databases, and no snowballing was conducted. However, the databases we used are commonly considered the main sources, particularly for the software/requirements engineering communities. Thus, the conclusions drawn here may not be generalisable to other areas. Additionally, as we terminated the review process as we reached saturation in the coding process, a number of potentially relevant papers were not reviewed. Thus, though our sample was saturated, it is possible that some very relevant concerns remain unobserved due to this. Moreover, the very recent work on social sustainability will also be missed from this study, as the search and analysis was carried out in 2016.

2.4 Summary

This chapter has presented two literature reviews on social sustainability. The first review presented the evaluation of sustainability frameworks, models and assessment in software. The second part of the literature presented a social sustainability systematic literature review. We observe that social sustainability indicators (RQ2 in Section 2.2.2.2) are closely aligned to social values. Thus, in the next chapter, we will discuss the value research.

Chapter 3

Literature Review on Values

First, this chapter presents the analyses of value research. In Section 3.1, a general overview of value research is demonstrated. This is followed by a specific literature review on values in ICT domain in Section 3.2. Similar to the review in section 2.1 of Chapter 2, this review is a traditional literature review of value research papers that contributes to the ICT field. The relevant studies were evaluated against the five characteristics discussed earlier in Chapter 2, Section 2.1. The results of the analysis (discussed below) confirmed the need of a general, holistic and reusable approach of treatment of social sustainability concerns (as values) in requirements engineering.

Then, we present how social sustainability indicators are mapping to values in Section 3.3. Next, equality value (social concept) is discussed in Section 3.4.

3.1 Value Research

Locke defined values as "what people want or consider beneficial to their welfare" [128]. In an attempt to form a values definition, Cheng and Fleischmann

[129] summarised seven previous studies [130–136] to come up with a comprehensive definition that incorporates previous work. Additionally, and with the aim of establishing a meta-inventory, Cheng and Fleischmann inspected twelve value inventories [130, 131, 136–142], (Crace & Brown, 1995; Scott, 1965; Bernthal, 1962; cited in [129]). As a consequence of this effort, value is defined as the " guiding principles of what people consider important in life". The result of comparing twelve value inventories composed a meta-inventory of human values with 48 value concepts/categories. Out of the 48 concepts, 16 of the concepts existed in at least 5 value inventories. The categories are freedom, helpfulness, accomplishment, honesty, self-respect, intelligence, broad-mindedness, creativity, equality, responsibility, social order, wealth, competence, justice, security and spirituality [129]. Due to the resemblance of concepts, equality, freedom from bias and fair treatment/fair competition were grouped under equality [129].

Schwartz and Bilsky [131, 143, 144] acknowledged five features that mark values: 1) belief, 2) desirable end states or modes of conduct, 3) "transcends specific situations", 4) selection or evaluation (behaviour, people and events) guidelines, and 5) ordered values reflecting priorities. Those features were incorporated in the values definition given by Schwartz [131]:

> I define values as desirable transsituational goals, varying in importance, that serve as guiding principles in the life of a person or other social entity. Implicit in this definition of values as goals is that[:] (1) they serve the interests of some social entity, (2) they can motivate action-giving it direction and emotional intensity, (3) they function as standards for judging and justifying action, and (4) they are acquired both through sociali[s]ation to dominant group values and through the unique learning experiences of individuals.

Ten values were identified: power, achievement, hedonism, stimulation, self-direction, universalism, benevolence, tradition, conformity and security [131]. These were joined to form four main value types: openness to change, self-transcendence, self-enhancement and conservation. For instance, values of universalism and benevolence are grouped under the self-transcendence value type [131].

The World Value Survey (WVS) is an international survey conducted since 1981 that studies possible impacts of values on social and political aspects [145]. The survey covers around 100 countries and is used in different areas such as in assessing happiness and wellbeing. In [146], Vinson and Ericson [146] studied Australians' happiness and life satisfaction (known as subjective wellbeing [147]). The data were extracted from the WVS. The study defined values as a predicting variable. The variable included social class, left-right (political preference), human rights (individual's perspective) and choices in life that reflects autonomy. In addition, the values variable included self-set goals, the meaning of life, and religious identity. Moreover, the trust variable was also a predictor variable in the study that included trust in people, trust in the family and the fairness of people. Almost all were found to be related to happiness and satisfaction (some only related, others with strong associations such as choice of life) except for the self-set goals. According to Vinson and Ericson, the study's results imply that "human service providers should engage with clients. Assistance should be rendered with an emphasis on maintaining and strengthening people's management of their lives" [146].

The Public Interest Research Centre [148] stated that values such as equality, tradition, wealth and creativity represent an abstract model of guiding forces that frame individuals' thoughts and actions. Understanding values will lead to better solutions of today's social problems such as poverty [148]. Understanding values allows the discovery of connections between different concerns such as community welfare and sustainability [148]. Sustaining and improving the values of a population is viewed as the way to achieve social sustainability [149].

Since values are universally recognised and used and because they also have influence on software design and usage, it is essential to incorporate them into software requirements.

Monetary values are beyond the scope of this thesis. They are mainly addressed in the value-based software engineering (VBSE). VBSE combines software engineering principles with stakeholders value concerns [150]. The VBSE is concerned with financial values gained by the software such as sales, costs, return-on-investment, and market share [150]. Those values are recommended to be used as evaluation criteria for decision-making purposes during the RE process.

Next, in Section 3.2, we analyse the frameworks for values. For the analysis, the same characteristics discussed earlier (Chapter 2, Section 2.1) are used.

3.2 Values and ICT

Friedman and colleagues define value sensitive design (VSD) as "a theoretically grounded approach to the design of technology that accounts for human values in a principled and comprehensive manner throughout the design process." [17, 136]. Human computer interaction domain (HCI) has the lion's share of the application of this design approach [151]. "Value Sensitive Design builds on an iterative methodology that integrates conceptual, empirical, and technical investigations" [17, 136].

- Conceptual investigation involves clarifying values, the values to be considered in the project, how to support the values within the design, how to deal with values' trade-offs as well as conceptualising how direct and indirect stakeholders could be affected by the design [17, 136].

- Empirical investigation includes studying the usefulness of the proposed design in a "human context" using qualitative or quantitative methods such as "observations, interviews, surveys, experimental manipulations, [the] collection of relevant documents, and measurements of user behavior and human physiology" [17, 136].
- Technical investigations can be performed in two different forms. One is to "focus on how existing technological properties and underlying mechanisms support or hinder human values". The other is to "involve the proactive design of systems to support values identified in the conceptual investigation." [17, 136].

Friedman and colleagues discussed the application of the VSD in three projects. The first project was 'Cookies and Informed Consent in Web Browsers'. During the conceptual investigation, values such as disclosure, comprehension, voluntariness, competence, and agreement were identified as being relevant. Next, a retrospective analysis was operated to find the various ways in which the cookie and web browser technology had changed using Netscape Navigator and Internet Explorer [17, 136]. Their conclusion was that there were improvements over time but some problems remained. As a consequence, the Mozilla browser was redesigned with three new mechanisms; i.e. peripheral awareness of cookies, just-in-time information about individual cookies and cookies in general, and just-in-time management of cookies [17, 136]. Periodicl assessments of the designed technology were conducted as part of an empirical investigation [17].

The second project discussed was 'Office Window of the Future' [17] later called 'Room with a View: Using Plasma Displays in Interior Offices' [136] which concerned replacing the direct view of nature with technology using a plasma window. The conceptual investigation in this lead to three related values; i.e. physical health, emotional well-being and creativity [136]. In addition, the investigation identified indirect stakeholders; i.e. individuals in public spaces who should be considered and their desired values such as privacy [17]. For the empirical investigation, a survey of 750 participants and 30 interviews on privacy were conducted targeting the indirect stakeholders in addition to the empirically collected data from users [17]. Additional empirical investigations were undertaken to assess the usefulness of the plasma window on peoples' welfare using "physiological and performance data, as well as data regarding the users' conscious perceptions" [17]. Those measures were:

> physiological data (heart rate), . . . performance data (on cognitive and creativity tasks) . . . video data that captured each subjects eye gaze on a second-by- second level, and time synchronized with the physiological equipment, so that analysis can determine whether physiological benefits accrued immediately following an eye gaze onto the plasma screen, and . . . social-cognitive data (based on a 50-minute interview with each subject at the conclusion of the experimental condition wherein they garnered each subjects reasoned perspective on the experience)[136]

The third project explained is 'UrbanSim; the Integrated Land Use, Transportation, and Environmental Simulation' [152]. During the conceptual investigation, it was discovered that there are various values desired by the diverse stakeholders and it is necessary to decide which to include. Fairness, specifically freedom from bias, accountability and democracy were the selected values to be embedded in the simulator system. "Most of the technical choices in the design of the UrbanSim software are in response to the need to generate indicators and other evaluation measures that respond to different strongly-held stakeholder values"[17, 136]. For example, having walking as a mode of transportation in the design supports fairness and democracy. Additional technical choice is to write simulation results in SQL database which will make the production of new indicators easier. Furthermore, the project development is decided to be more of an agile development process. Moreover, a prototype of the interface was developed based on the idea of helping "stakeholders characteri[s]e their underlying values, and agree upon the indicators to be computed by the simulation to help them evaluate the outcomes in light of those values."

Friedman et al. [136] provided a list of values that are most relevant to systems design as "a heuristic for suggesting values that should be considered in the investigation". The list comprises values of human welfare, ownership and property, privacy, freedom from bias, universal usability, trust, autonomy, informed consent, accountability, courtesy, identity, almness and environmental sustainability [136]. Further analysis is demonstrated in Table 3.1.

TABLE 3.1: Evaluation Criteria in Value Research (Friedman et al.)

Literature	Criteria	Category	Why?
Friedman et al. [17, 136]	C1: Operationalised requirements	Full	This results into design criteria and requirements.
	C2: Re-usability	Full	List of values produced
	C3: Generic view	No	No generic view is provided that can be used while operationalising the concepts.
	C4: Expertise	Yes	No restrictions on how to carry out the investigations.
	C5: Used with different RE practices	Yes	No restrictions on how carry out the investigations.

Thew and Sutcliffe proposed the values, motivations and emotions (VMEs) elicitation method [46]. VME's are viewed as reasons behind software acceptance and use [46]. The method comes in two versions: one to support experts and the other to support novice requirements analysts [46, 47]. The method is combined with VME's taxonomies that resulted from literature survey and is used during the analysis.

The values taxonomy lists 8 value categories, of which 6 are considered to be general concepts of trust, collaboration, morals/ethics, creativity/inovation, aesthetics and security [46]. Personal characteristics value is also part of the value taxonomy. The taxonomy also includes beliefs and attitudes as a value that changes rapidly compared the the other values which are considered to be more stable value [46]. For each value, related terms are provided, potential sources that analysts could investigate to find the desired value, and implications on requirements engineering practices. For example, collaboration value can be related to terms of cooperation, friendship, sympathy and altruism. Collaboration value can be explored through stakeholders' relationships with others. This value implies that there is a need for improved team cooperation during the project. The VME's taxonomies are used to build a website to support analysts. The website provides VME's that can be explored each in a page. Each page provides the value, possible interview questions that help to explore the value, scenarios which represent examples to the analysts on how a value can affect software development projects, and possible implications on project development or design.

The method starts with initial analysis of the project to anticipate possible 'hunches' (relevant VME's) based on analysts' previous projects and knowledge [47]. The related VME's are then further investigated in interviews or other investigations. Taxonomy categories are used to annotate the interviews' textual records. Then VME's are recorded and the taxonomies are modified after each investigation cycle. Then the VMEs' taxonomies and the annotated interviews, observations, meetings and other investigation records are reviewed to present lists of functional and non-functional requirements, recommendations for organisations, user procedures, functional allocation and work design [46].

Although taxonomies are used within the method, it still suggests re-investigating the values for each software to be developed. More analysis is depicted in Table 3.2.

Table 3.2: Evaluation Criteria in Values Research (Thew and Sutcliffe)

Literature	Criteria	Category	Why?
Thew and Sutcliffe [46]	C1: Operationalised requirements	NA	The method supports eliciting VME's from the requirements.
	C2: Re-usable	Full	Taxonomy and website tool
	C3: Generic view	NA	The method supports eliciting VME's from the requirements.
	C4: Expertise	Yes	Method presented with two versions to be used by novice and experts.
	C5: Used with different RE practices	Yes	No restrictions imposed. Validation found the method useful with different practices such as interviews, workshops and prototyping.

Ferrario et al. [153] proposed the concept of value-first software engineering which suggests a relationship between values and software engineering (SE) decisions. According to Ferrario et al., "values-First SE explicitly uses human-values as a reference framework for decisions making at key stages of software development" [153]. The concept is constructed on an empirical basis with a participatory nature (as other value research works) [153]. However, Ferrario et al. claim that complex issues such as sustainability are to be addressed as 'soft-goals' and not functional requirements [153]. This is similar to other studies [62, 63, 68]. The approach maps users' requirement to values. In the case study provided, user requirements and development team principles were mapped to "Self- enhancement and Conservation for the users; Self-transcendence and Openness to change for the research team" [153]. Those values were then used to identify system qualities:

> Intentionality, the system must afford control to its users including intentional interactions, not only passive automated sensing; Personalisation, the system needs to adapt to unique users' needs; Data Transparency, the system affords end-user data ownership, transparent data capture, storage, and curation; Openness, the system must access and can be accessed by other services; Modularity, system functionalities can be easily

added or removed; Reusability, the system can be easily re-purposed for other domains.

Ferrario et al. have notified that the values mapping can be done manually or automatically using thematic extraction [153]. However, these skills are mainly found in software engineering researchers. The novice practitioners might be limited in terms of these skills and also project time constraints can prevent them from using the value-first SE framework. Detailed analysis is available in Table 3.3.

TABLE 3.3: Evaluation Criteria in Value Research (Ferrario et al.)

Literature	Criteria	Category	Why?
Ferrario et al. [153]	C1: Operationalised requirements	Partial	Only non-functional.
	C2: Re-usability	Partial	In mapping requirements to values, they used [148] which is based on Schwartz values
	C3: Generalisable concepts	No	No definition that can guide instantiation in operationalisation
	C4: Expertise	No	Mapping skills might not be common with novice practitioners
	C5: Used with different RE practices	Yes	No restrictions imposed.

Greef et al. [154] proposed a sCEthics design methodology that couples value sensitive design (VSD) with the situated cognitive engineering (sCE) methodology to incorporate values. The sCE comprises three stages: generation, evaluation and refinement. According to Greef et al., the proposed methodology should include guidance in ethical values and ways to reach consensus among stakeholders on values. In addition, it should include legislative documents and policy guidelines to help define restrictions and constraints in addition to extracting requirements from ethical values. To explain how values are translated into requirements, an extended scenario system should form part of the sCEThics. The suggested scenario system is the Benyon's layered scenario-based design method [155]. The system includes the use of a scenarios layer, conceptual scenarios layer, concrete scenarios layer and use-cases layer. Additionally, design patterns are to be used to show what the implementation of requirements look like. Moreover, an overview of the design process elements is to be provided to show the connections between the elements such as the requirements and the use-cases using matrix overview and radial visualisation. sCEThics was evaluated by eleven participants, all of whom had a basic knowledge of user-centred design. The evaluation revealed that having the scenario system is not useful because it requires extra effort and offers no remarkable benefits (relative to the extra work required). Analysis of the study is illustrated in Table 3.4.

TABLE 3.4: Evaluation Criteria in Value Research (Greef et al.)

Literature	Criteria	Category	Why?
Greef et al. [154]	C1: Operationalised requirements	Full	Combining VSD and sCE allowed this using scenario system.
	C2: Re-usability	Partial	Design patterns are used within the method
	C3: Generic view	No	No generic concept is specified and to be re-used in different projects.
	C4: Expertise	No	Specific knowledge with scenario system is required
	C5: Used with different RE practices	No	Use of scenario system is a restriction.

Studies have discussed how diversified values originating from different stakeholders can be opposed and do cause conflicts (sometimes called value tensions) [131, 151, 153]. As values are related to requirements, this is discussed in Appendix A.5.

Table 3.5 presents the overall analysis of the studies discussed above. It confirms that there is a lack of generic, operationalisable, reusable view of social sustainability. In order to define software's social sustainability requirements, we need to have clear representation of what social sustainability is.

TABLE 3.5: Overall Analysis- Value Research

Literature	Operationalistion	Re-usability	Generic View	Expertise	RE practices
Friedman et al. [17, 136]	Full	Full	No	Yes	Yes
Thew and Sutcliffe [46]	Full	Yes	NA	Yes	Yes
Ferrario et al. [153]	Partial	Partial	No	No	Yes
Greef et al. [154]	Full	Partial	No	No	No

3.3 Core Social Sustainability Indicators Mapping to Values

In Table 3.6 below we show how our 12 social indicators map to the values inventory compiled by Cheng and Fleischmann (i.e. freedom, helpfulness, accomplishment, honesty, self-respect, intelligence, broad-mindedness, creativity, equality, responsibility, social order, wealth, competence, justice, security and spirituality) [129]. This value inventory is an aggregation of 12 value inventories (see Section 3.1).

Our social sustainability indicators mapped to 14 out of the 16 value categories of Cheng and Fleischmann's values. The remaining unmatched categories are spirituality and wealth. Wealth could be related to equality if we are to consider the economic benefits. Spirituality could also be viewed as part of cultural indicator.

Given that our common indicators mainly came from projects assessments, they are primarily based on tangible aspects that can be evaluated. This might be a reason for not finding a match for spirituality. *Nevertheless, addressing social sustainability through values is a legitimate path.*

We observe that values are related to social sustainability in previous work from several domains. For instance, values are used for social construct evaluations (e.g. happiness and well-being [146], poverty [148], and sustainability [149]). There are also some pieces of research that directly link social sustainability and values. For examples, employment as a social sustainability value was investigated in the product - service system, where the product is a building, while service is a "social value" employment [156]. Factors contributing to a sustainable product-service system were researched in [157]. It was observed that changing customers' behaviour contributes to social sustainability [157]. To foster such improved social sustainability, it is suggested that it is important to educate the customers and allow them to participate in product design and similar processes [157] (in accordance with our social sustainability structure, shown in Figure 2.3, participation falls under the social cohesion in the social indicators).

Others [158] conducted a survey on sustainability documentation (such as the Millennium Declaration [159]) and reported that values can be mentioned directly (e.g., Millennial Declaration), appear as principles (e.g. in the Earth Charter), or "be inferred from adopted goals, targets, or even indicators". However, "values are always abstract ideals that define or direct us to goals and provide standards against which the behavio[u]r of individuals and societies can be judged" [158]. Indeed, the "positive and long-term condition within communities" (i.e., social sustainability) [2, 3] occurs when the "behavio[u]r of individuals and societies" [158] is judged (by other individuals within these societies, and societies themselves) to be in line with the "value standards" for such a "positive condition". Therefore, we conclude that our ***12 sub-categories of the social sustainability concerns are indeed the representative aggregated "behaviour standards" motivated by values.***

We have identified the structure of the social sustainability concern as consisting of (at least) 12 areas as per our literature review (Chapter 2), and possibly 16. However many, each area is considered to be a value. In order to address social sustainability concerns, all (12 or 16) areas need to be addressed. We illustrate how these areas can be addressed by treating Equality - one of the key areas of social sustainability.

We start with a targeted literature review on the topic, as presented in Chapter 4, then modelling of pattern/template (Chapter 4), and its evaluation (Chapters 6–8). But next, we present equality as a social value.

TABLE 3.6: Social Indicators Mapping to Values

Social Indicators	**Cheng and Fleischmann's values**
Employment	Accomplishment, helpfulness
Health	Helpfulness
Equality	Equality, Justice
Education	Accomplishment, competence
Security	Security
Social cohesion	Broad-mindedness, honesty, creativity, responsibility/accountability
Services and facilities	Competence
Resilience	Security (as survival)
Human rights	Freedom
Social acceptance of technology	Although this indicator is related to technology/project, if we thought of it in terms of human, it will map to self-respect and broad-mindedness.
Cultural	Helpfulness, Broad-mindedness, intelligence/wisdom
Political	Social order

3.4 Equality

O'Brien [160] looked into social justice as the structure framing social workers' practices. Social workers were asked to define social justice and give an example of a social justice case they had experienced and how they acted either to maximise justice or minimise injustice. Social workers have linked social justice to equality and/or fairness[1]. The results showed that equality has a broad definition (see Figure 3.1). One definition was related to equal treatment and equal access. This definition relates to fairness because not treating everyone equally causes complaints of

[1]Fairness is defined as "the quality of treating people equally or in a way that is right or reasonable". Cambridge Dictionary. `https://dictionary.cambridge.org/dictionary/english/fairness`

unfairness [160]. The other definition which is considered by the author as "radical" suggests that equality means providing different treatments to accommodate the diversified groups and individuals. "At this end of the spectrum, failing to provide different treatment is itself unequal because of the unequal social and economic position of different groups. . . . in this position, diversity enhances equality by ensuring that unequals are treated differently and unequally" [160]. Participants also translated equality into different sets of practices such as equal treatment, equal access, equal rights, equal opportunity and varied programmes and services to achieve equal outcomes. Additional equality concerns were related to polices against inequality and the redistribution of resources and power. The author concluded that:

> Some of the practitioners have quite explicitly defined equality and fairness as requiring different treatment, policies and programmes while others have emphasized sameness. Arguments for differential approaches, reflecting differences in economic, cultural, ableness and gender locations make arguments for identical treatment difficult to sustain, reflected in expressions such as 'one size doesn't fit all'. At the same time, difference alone is insufficient unless it is linked with and built on a base of equality.

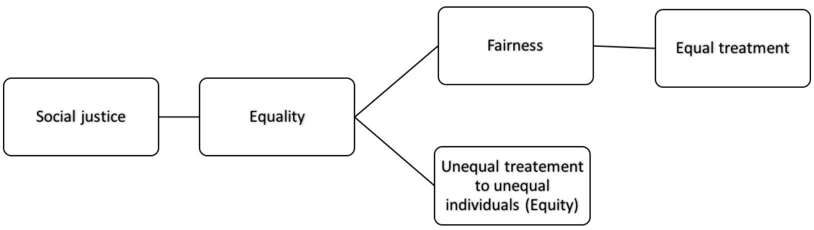


FIGURE 3.1: O'Brien's Justice concept

Eckhoff addressed equality as a principle of justice [161]. Objective equality is distributing the same amount of value/resource/benefit, etc. to each person [161]. Another view of this equality is distributing the same kind of value/resource/benefit, etc [161]. Here, recipients' characteristics are not a concern [161].

Subjective equality takes into consideration the needs of the recipients of value/resource/benefit, etc. as well as what they want or deserve (reward/punishment) to achieve [161]. Relative equality is satisfied by having the same ratio for everyone [161]. The ratio is between the distributed value/resource/benefit, etc. and recipients' characteristics or performance [161]. In this equality, recipients' fitness criteria and what they need or deserve (reward/punishment) is taken into consideration [161]. Fitness criteria refers to recipients' ability to "utili[s]e or take care" of what they receive [161]. Alternatively, it may be related to their ability to learn from what they receive or the ability to handle the imposed burden by the receiver, etc [161]. Relative equality reflects equity where the ratio between inputs/outputs is the same for all [162]. Rank order equality ensures that the distribution of value/resource/benefit, etc. are based on the condition of consecutive order (e.g. rank, worthiness, seniority, position in a queue, etc.) [161]. For example, eldest recipients first and then the youngest [161]. Equal opportunity is about recipients having equal chances to get an indivisible value/resource/benefit, etc. by chance (e.g. "drawing lots") [161]. In this form of equal opportunity, no consideration is given to recipients' characteristics [161]. The other form of equality of opportunities is related to the recipients getting an indivisible resource based on their efforts, needs or fitness [161]. An example given for this type of equality regarding the government's allocation of educational services: "everyone has the same opportunity of developing himself" [161]. This category can come in two forms; one form takes into consideration the needs of the recipients and their fitness criteria [161]. Eckhoff noted that recipients, characteristics can be combined in some cases and there are no clear divisions between them [161]. Eckhoff's view of justice is depicted in Figure 3.2.

Cook and Hegtvedt studied justice in terms of equity and distributive justice [163]. A common equity definition is the fair exchange of valued resources between parties where the benefits (relatively evaluated as the outcome/input ratio) received by each

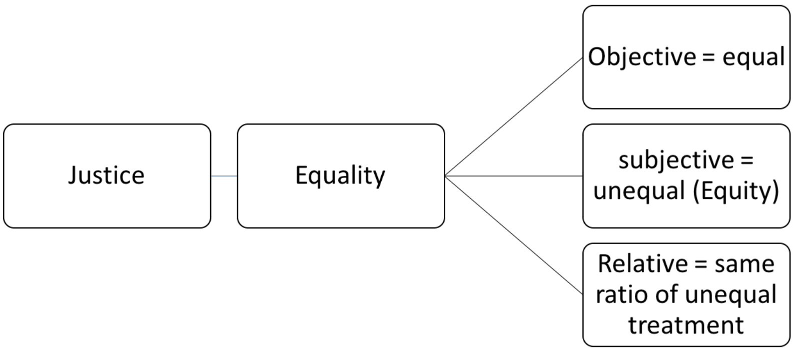


FIGURE 3.2: Eckhoff's Justice concept

party are equal [163]. Cook and Hegtvedt also defined distributive justice as fair allocation (by an allocator) of valued resources, rewards, rights, etc. (to different recipients). Procedural justice looks into the fairness of the procedures/processes of exchange and allocation. Cook and Hegtvedt also defined retributive justice as fair compensation in terms of punishments and victims' compensation. Incorporating the different justice concepts (equity, distributive justice, procedural justice, retributive justice) is known as multiple justice principles or the distribution rule which was explained using Eckhoff's framework of justice (through different equalitiy classifications) [161]. Using Eckhoff's framework, Cook and Hegtvedt classified justice into two levels: dependent and independent on recipients' features. Thir work has shown that equality points out justice, fairness and equity terms [163] (see Figure 3.3).

All those studies [160, 161, 163] have revealed that the equality is a concept with different explanations. Thus, equality is a term that encompass equity and fairness. In addition, equality serves the bigger goal of justice. The idea of equality is depicted using the dependancy graph in Figure 3.4. This is a summarised view based on the previously discussed studies [160, 161, 163]. This view is utilised in Chapter

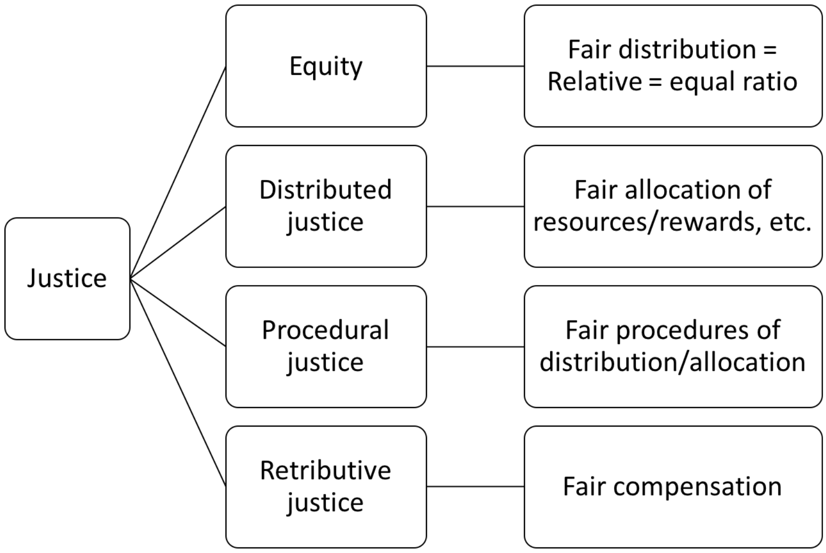


Figure 3.3: Cook and Hegtvedt Justice concept

4 to build the equality value pattern. *This emphasises the need for a model that allows software practitioners to address equality and tailor it to the software under consideration.*

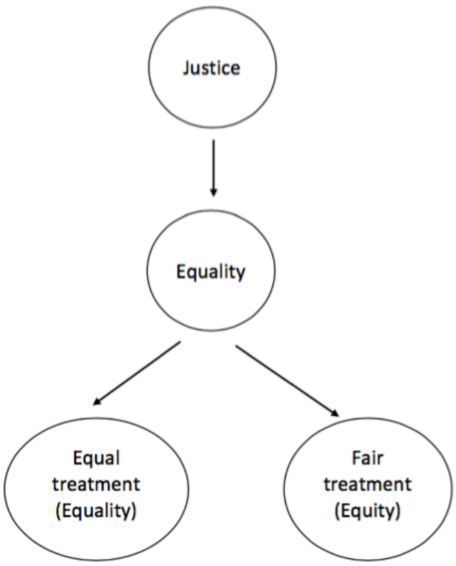


Figure 3.4: Equality concept

3.4.1 Equality and Standards

According to the Equality and Human Rights Commission [164]:

> Equality is about ensuring that every individual has an equal opportunity to make the most of their lives and talents, and believing that no one should have poorer life chances because of where, what or whom they were born, what they believe, or whether they have a disability. Equality recognises that historically, certain groups of people with particular characteristics e.g. race, disability, sex and sexuality, have experienced discrimination.

Equality Act 2010 aims at reducing soci-economic inequalities [165]. The act highlights important points that should be followed by employers and service providers to support equality. The act identifies protected characteristics: age, disability, gender reassignment, marriage and civil partnership, pregnancy and maternity, race (colour, nationality, ethnic or national origins), religion or belief, sex and sexual orientation. People with protected characteristics shall be prevented from possible inequalities and prohibited conducts. Prohibited conducts include discrimination, harassment and victimisation [165].

Al Hinai and Chitchyan [166] summarised ISO 26000:2010 [167] and SA8000:2014 [29] and how the standards embed equality issues[2]. ISO 26000:2010 [167] is an international standard guideline that provides guidance to organisations to allow them to work in a socially responsible manner [166, 167]. This standard's principles cover equality issues such as gender equality in areas of recruitment, training, payment and community health and safety [166, 167]. In addition, underrepresented groups (e.g. women, girls, people with disabilities, children, indigenous people, elderly,

[2]This was presented in the Fourth International Conference on ICT for Sustainability 2016 [166].

poor, etc.) should be considered and provided fair treatment and opportunities so as to achieve higher positions in an organisation [166, 167]. ISO 26000 also mentions labour practices to "ensure equal opportunities for all workers and not discriminate either directly or indirectly in any labour practice and eliminate any arbitrary or discriminatory dismissal practices" [167]. In addition, organisations should "provide equal pay for work of equal value" [167]. Furthermore, workers should have access to skills and career development opportunities without discrimination [166, 167]. The guidelines also note that organisations should undertake fair operating practices that include supporting fair competition between value chain members, respecting property rights, and paying fair compensation to the owner of any acquired or used property [166, 167].

Social Accountability Standard 8000 (SA8000: 2014) [29] provides organisations with guidelines regarding child labour, forced or compulsory labour, health and safety, freedom of association and collective bargaining, discrimination, disciplinary practices, working hours, remuneration and management systems [29, 166]. In terms of discrimination, the standard suggests that organisations shall not be part of any discriminatiory activity whether in terms of "hiring, remuneration, access to training, promotion, termination . . . any other condition that could give rise to discrimination" [29, 166]. Additionally, organisations are to ensure that workers do not face discrimination if they are "union members, representatives of workers and any personnel engaged in organising workers" [29]. Furthermore, the organisation should allow workers to meet their needs related to "race, national or social origin, religion, disability, gender, sexual orientation, family responsibilities, union membership, political opinions or any other condition that could give rise to discrimination" [29].

Being an internationally accepted and advocated concept suggests that equality is a recurrent and universal concern and that is applicable (reusable and repeatable) across domains.

3.5 Threats to Validity

3.5.1 Construct Validity

The analysis conducted in this Chapter was performed by the author which raises a threat to validity. The calculation of Kappa test of inter-rater reliability [126] is unattainable as this review is a traditional review.

The subjective nature of the analysis could introduce misinterpretation as well as under representation of studies.

To reduce the threats, the analysis of the reviewed text was based on the characteristics C1-C5 described in Chapter 2, Section 2.1. The characteristics were defined and decided on prior to the current study.

3.5.2 Internal Validity

Selection of studies using snowballing procedure could have resulted in missing key references. To mitigate this threat, we broadly followed Webster's and Watson's guidelines [168]. This was done by identifying value related studies using digital libraries such as IEEE that are commonly considered the main sources, particularly for ICT communities. Once we came across a relevant study, its reference list was investigated to further identify possibly relevant research studies to be included in the current review. In addition, forward review of studies that cited the initially selected studies was also considered.

3.6 Thesis Agenda

We have seen in this chapter and in the previous chapter (Tables 3.5, 2.9) that there is a need for a methodology for operationalising the notion of social sustainability (values) into software requirements. In addition, this methodology will facilitate reuse of requirements. Additionally, it will be usable by novice and experienced requirements engineers and with different RE practices and processes.

3.7 Summary

In this chapter, we demonstrated the related work in value and ICT domain. We also clarified the mapping of social sustainability indicators to values. Then, we presented the equality value as one of the social sustainability values. We selected this value to instantiate the methodology proposed by this thesis.

Chapter 4

Equality Value Pattern Development

Once social sustainability core values were identified as detailed in Chapter 2, the need to relate social sustainability to software requirements in a practical way became clear.

To operationalise sustainability, all its constituent parts (listed in Figure 2.3) would need to be treated. As each of these constituent topics is a large area on its own, we start with a single topic, aiming to demonstrate how the similar process could be applied to the other constituent topics.

While each of the topic areas (as per Figure 2.3) could have served as an operationalisation demonstrator, one topic had to be selected. We opted to work with equality, which is not only a key characteristic for social sustainability, but is also relatively poorly addressed in Software Engineering domain.

Thus, using the example of equality concern, in this chapter we present a process through which the previously identified topic areas of social sustainability could be transformed into distinguishable patterns and operationalised into practically usable (and reusable) requirements for software systems.

To accomplish the above, we rely on deriving generic patterns, into which social values coalesce. These patterns are identified from an in-depth review of the related literature. The patterns are then operationalised into templates that facilitate relevant requirements identification and documentation. In other words, in this chapter we demonstrate the method used to develop social sustainability pattern i.e. equality.

The aim is to supply software practitioners with guidance to allow them to embed social sustainability concerns in software requirements specification[1].

4.1 Study Overview

This study follows the commonly used practices of pattern development (see Background Concepts in Appendix A, Section A.4).

Starting with the sub-set of papers from the structured literature review (see Section 2.2) that treats equality, we undertake a more in-depth analysis of the equality concern.

This study was conducted to reflect on the notion of equality and gain a clear perspective of the way that the key contributing notions to this concern inter-relate. According to Miles, Huberman and Saldaña [169], in qualitative data analysis: "[c]odes . . . detect recurring patterns". Additionally, because the aim of this research is to observe the "regularly repeated arrangement" [170] of the values (which, as further

[1] This study was presented at the 4th International Conference on ICT for Sustainability in 2016 [166]

discussed in Chapter 6, apply to various domains and are commonly accepted), these values are referred to as value patterns.

Section 4.1.1 is a demonstration of the first step of pattern development were documents are analysed with the aim of finding recurrent issues (as discussed in Appendix A, Section A.4). Next step is to build an understanding of the issue under study and solutions (in our study it is about equality) is demonstrated in Section 4.1.2. Pattern representation is illustrated in Sections 4.2 and 4.3 as the third step in equality pattern development. Pattern application stage is explained in Chapter 6 and pattern evaluation is in Chapters 7 and 8.

4.1.1 Study Data Sources and Method

To undertake this study we used a number of qualitative text analysis [171] techniques that were applied through two rounds.

In round 1 an initial sub-set of 11 papers [1, 172–181] was selected from the previously reviewed literature on social sustainability (as detailed in Chapter 2). For this we chose the papers that, in accordance with our previous indicators and metrics analysis, had flagged equality as a concern (this includes the equality notions discussed in Chapter 3, Figure 3.4) .

We then undertook in-depth content analysis. Starting with the set of codes related to the indicators, we used a mixed approach of inductive and deductive coding [169]. Where the codes were available from the previous analysis exercise (i.e., the indicators categories collected, as per Chapter 2), they were utilised. However, the availability of these categories did not restrict us in inductively defining new codes that described the concepts that were deemed relevant to equality and their interrelationships, for which emergent coding (or open coding as per grounded theory) was used [182, 183].

The initial coding was conducted by the researcher for this thesis, which was then discussed and validated by the thesis supervisor. Where disagreements occurred, the text was re-coded and discussions continued until a consensus was achieved.

The codes were then aggregated into initial value categories and a draft value pattern was derived.

In round 2 Here we used the summative content analysis [171] technique. Whereas in the first round we looked in-depth at the whole of the papers' contents, the focus with summative content analysis was on targeted reading of the text with respect to specific search words i.e. equality, fairness, equity, and justice (Figure 3.4) and their usage in the text [171]. The analysis was conducted to understand the underlying (latent) meaning of these words and their content [171, 184, 185].

Using our four search terms, additional 26 studies were identified and examined. The data sources used in the analysis are listed in Table 4.1. The analysis sought to understand the meaning of equality and to reflect on how equality could possibly be related to software requirements. In addition, memos were used to record any ideas and notes during the coding process.

The resultant codes were grouped into three main categories (as per Table 4.2). Here, the main categories are suggested sub-values that contribute to the equality value.

We have used two methods to report the coding results: a) narrative description presented in Section 4.1.2; and b) visual representation of the results in Section 4.2.

Table 4.1: Data sources for qualitative data analysis

Source	Articles
Scopus	[1, 78–80, 85, 86, 97, 99, 114, 116, 173, 176–178, 186–188]
Springer	[77, 172, 175, 179–181, 189–196]
IEEE	[94, 105, 119, 174, 197]
Web of science	[101]

4.1.2 Study Results

Upon completion, three main categories have emerged: equality with stakeholder variability, fairness to stakeholders goals and equality with access to services and facilities. These categories summarise what equality means and facilitates the idea of relating social sustainability to software requirements. The categories and their definitions are detailed in Table 4.2 and examples of coded text under each category is in Tables 4.3, 4.4. 4.5.

To apply the ideas in this section, we will use the statement "a restaurant should provide food to all customers" as a sample.

The equality with stakeholder variability category refers to factors that cause differentiation between stakeholders. Factors of differentiation can originate from human factors such as age, gender, race, religion and disability or materialistic factors such as income. It can also be attributed to knowledge level and the type of technology used by stakeholders. Applying this category to our sample restaurant statement, we consider customer variability to include gender: male and female, age: young and old, religion (Muslim, Christian, ... etc). Examples of coded text under this category are in Table 4.3.

Table 4.2: Categories definitions

Category	Definition	Coding rules	Sources	References
Equality with stakeholders variability	Refer to variability factors between stakeholders that can be a reason for inequality. Codes under this category includes gender, class, disability, literacy and skills, location, race, religion, power, economic level, age, caste, citizenship.	Point to variability factor or/and equality related to variability	27	125
Fairness in the selection of stakeholders goals	Refers to stakeholders goals that if met, will improve equality between community members. They will feel that they are part of the community and not being marginalized or ignored. Codes under this category includes distribution of information, equal opportunities, fair competition, fair distribution of benefits, fair practices, social benefits, social justice and overcome variability.	Point to stakeholders goals that they would like to achieve. Usually refers to achieving benefits or reducing/avoiding risks. Negative goals that needs to be reversed to achieve equality are also part of this category.	38	254
Equality with services and facilities	Refers to the services that stakeholders needs to access in order to achieve their goals. Codes under this category includes access to information and access to services, resources, and assets.	Point to a service that if accessed, a stakeholder will achieved his/her goal from the application	19	86

Table 4.3: Coded Text Sample for Stakeholder variability category

Categories	Source
The stakeholder variability	
“higher castes (social power), the middlemen (economic power), and males (gender power) . . . to those groups with less power.”	[174]
“knowledge of technology, literacy level, and skill capacity of users”	[174]
“elder teachers . . . younger users”	[174]
“personal disability”	[1]
“discrimination on the grounds of gender, religion, or race”	[177]
“opportunities for the very poor, women, young people, the disabled, families”	[189]
“local and international enterprises”	[80]

Table 4.4: Coded Text Sample for Fairness to stakeholders goal category

Categories	Source
The fairness to stakeholders’ goals	
“prompt information dissemination to relief agencies and affected communities”	[174]
“Fair competition”	[172]
“appropriate resource allocation and priority determination”	[175]
“Economic participation and opportunity”	[175]
“fair distribution of benefits among relevant stakeholders”	[178]
“fair operating practices”	[179]
“women’s power increases at all levels”	[189]
“community participation”	[193]

Table 4.5: Coded Text Sample for Equality for access to services and facilities category

Categories	Source
The equality for access to services and facilities	
“Tourist accommodation facilities such as homestay facilities, resorts and luxury hotels”	[178]
“access to material resource”	[172]
“access to information”	[174]
“access to economic-enhancing resources and livelihood assets”	[174]
“improved telecommunication and transportation facilities”	[181]
“women’s access to adequate water supply, health care, and employment”, “access to credit ”	[189]
“access to jobs and affordable housing”, “convenience shopping”, “primary healthcare”	[97]

Considering stakeholders' differences is important when equality in a community is to be attained. Failing to do so usually leads to inequality. As discussed in Chapter 3, according to the Equality Act 2010, the stakeholders' differences are called "protected characteristics" that specify diversified citizens and how to avoid inequality to those groups [165]. Citizens protected against inequality (or discrimination as used in the act) are those with differences in terms of age, disability, gender reassignment, marriage and civil partnerships, pregnancy and maternity, race, religion or beliefs, sex or sexual orientation [165]. This shows how the concept of stakeholder variability is an important part of achieving equality.

Other studies identify stakeholders differences as social factors such as age and gender [198]. The study has also discussed how age and gender produces different demand for and usage of ICT [198]. Similarly, in addressing disabled, elderly and low-skilled citizens' needs, accessing e-voting software was considered an important aspect of "equity of access" [199]. According to the study: "[a]ll eligible citizens should be able to use and access e-voting systems, including low-skilled, elderly and/or disabled citizens." [199]. The study in [200] referred to the disability as a concept of functional diversity that needs to be considered in software development and implementation.

The fairness to stakeholders' goals category introduces the connection between equality and satisfaction of stakeholders' goals. It suggests that being treated equally is related to considering stakeholders' goals and making it possible to happen (for examples of coded text under this category refer to Table 4.4). However, with the number of stakeholders it is not feasible in terms of time, cost and resources availability to work on all of the requested goals. Additionally, goals from different stakeholders can be conflicting and it is impossible to satisfy all groups. For that reason, fair goals selections between the desired goals (also summarised in Appendix A.5) is essential. Requirements engineers can use different requirements negotiation techniques to resolve the conflict such as [201–203]. One of the possible negotiation techniques is WinWin [204]. The negotiation approach itself implys equality by involving stakeholders in the negotiation [205]. Evaluation of WinWin in [205] suggests that the approach promotes "equalized participants".

Whichever technique is used for negotiation, it is essential to arrive to agreed upon set of requirements, else some stakeholders/groups will be inadequately supported by the resultant software system.

Although detailing the possible fair selection processes and negotiation techniques between the stakeholders' goals is beyond the scope of this study, it is important to remark that the interests of the weak stakeholder groups may be overpowered by stronger groups if their interests are not represented in a balanced way during the requirements negotiation process.

Applying this category to our sample restaurant statement, we observe that the customer variability drives additional consideration where all goals of the customers is to have a meal. To support these goals the restaurant will need to, for instance: provide children's menu for age differentiation (young), provide halal/kosher food for religious variability.

The equality for access to services and facilities category presents the importance of providing services and facilities to stakeholders in order to allow equal opportunities to achieve a specific goal. For example, stakeholders willing to obtain information from a system should be able to view information on a screen. Here, it is recognised that stakeholders' variability requires modification of the display function. For a normal viewer, information is displayed on screen in a textual format with normal font size. For a visually impaired viewer, text information is displayed with a customisation option to adjust the font size and colour. Furthermore, different display formats can also be utilised in the system to accommodate visually impaired users by displaying information in audio or video format. Applying this to our sample statement, a restaurant should provide food to all customers, we observe that in order to support achieving a meal to all customers, it may need to provide high chair seats for young customers and wheelchair accessible seating areas for disabled customers. Examples of these are further discussed in Chapter 6.

In [206], equity was discussed in terms of "equality of choices and equality of utilization" of health and educational services (public services).

The three categories contribute into answering sustainability questions suggested by (J. Tainter 2014) cited in [207]. The question addressed here are Sustain what? and For whom?. The first question answer is to sustain equality within software community. The second question is answered by the value of stakeholder and their variability. The Equality value pattern also answer another question (not from Tainter) which is how will the equality be sustained. This is answered by the values of fairness to stakeholders goals and the equality for access to services and facilities.

4.2 Results Visualisation

In order to visualise equality, the values underneath and the relations between the values, the generic sustainability model by [62, 63] (described in Chapter 2) was utilised. Figure 4.1 depicts the equality value pattern showing the relations between each sub-value and how they contribute to equality and, in turn, to social sustainability.

Magis and Shinn said that the imature sustainability topic is enlighted by the existing well established knowledge on social well-being [208]. As discussed in Chapter 3, values are used in the research area of communities welfare and well-being. In addition, PIRC noted that understanding values allows descovering connection between different concers such as community welfare and sustainability [148]. Thus, we choose to depict the value pattern using the this model, i.e. the generic sustainability model by [62, 63], as it maps sustainability into values.

NFR framework [209, 210] could have been used to represent equality but in that case, we will be assuming that equality initialy results only in non-functional/quality requirements. This is not the case as we demonstrate in Chapter 6 that religion variability introduced a new functionality of providing religious verification (e.g. halal food) in the Health Watchers example. This was also noted by Huber, Hilty and Glinz [211]. They found that considering sustainability concerns in eliciting software requirements can impose new constraints, increase importance of existing ones and introduce new functional requirements [211].

We built the value pattern on this model to:

(a) simplify social sustainability concerns into practical patterns which can fit to any development process (not only where goal modelling is used) and

(b) represent social sustainability requirements regardless of the resulting requirements type.

4.2.1 Equality Value Pattern Semantics

According to Penzenstadler and Femmer [62, 63], a dimension is an aspect that contributes to the overall goal (i.e. sustainability) [62, 63]. A goal can have one or many dimensions [62, 63]. "A dimension is represented by a set of values that express the abstract objectives of the dimension" [62, 63]. Additionally, a value can have subvalues [62, 63]. As the equality value pattern is depicted using Penzenstadler and

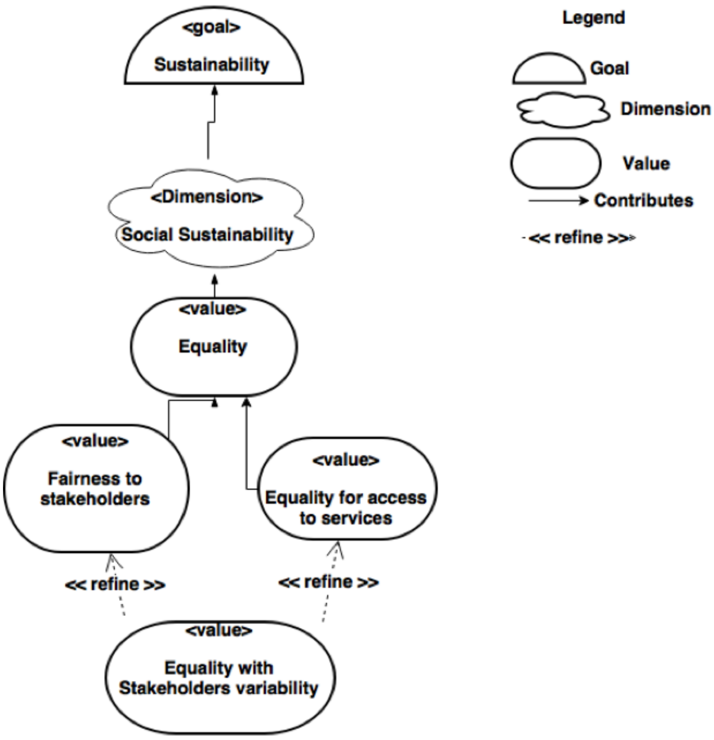


FIGURE 4.1: Equality Value Pattern

Femmer [62, 63] sustainability model, the equality value pattern semantics follows the semantics imposed by the meta model of the generic sustainability model (discussed in Chapter 2, Section 2.1.1). As we can see in Figure 4.1, the main goal is sustainability. Social sustainability is an aspect of sustainability. The social aspect is represented by the equality value which is composed of sub-values of equality with stakeholders variability, fairness to stakeholders and equality for access to services. From our practice (next in Chapter 6), we noticed that the stakeholder variability value can impose refinement on the resulting requirements of another value (i.e. fairness to stakeholders and access to services). For example, identifying visual impairment as a variability associated to a user imposes refinement of the printing services (from equality for access to services value) in an application to include access to Braille printing facilities.

As we are aiming at a simplified method of identifying social sustainability requirements (rather than instantiating Penzenstadler's and Femmer's generic model of sustainability) some parts of the model are omitted.

4.3 Requirements Templates

With the idea of providing software engineers with an effortless and rapid social sustainability requirements elicitation technique and to put the pattern into the context of use, requirements templates are used alongside the equality value pattern.

The templates are simple way of eliciting and structuring requirements, as shown in textual and tabular forms below:

- Textual. The template is ≪*Stakeholder*≫ ≪*with variability*≫ *should get* ≪*function/services/resources*≫ *to accomplish* ≪*goal*≫.
- Tabular as in Table 4.6.

Table 4.6: Equality Templates

Stakeholder	Variability	Software function/services/resources needed to achieve a goal	Goal
e.g. Citizen	e.g. Language	e.g. Select preferred language	e.g. customize application language

We used a set of questions to help operationalise the value pattern into equality templates (see Appendix A.3). Operationalising goals into requirements is a common practice in Goal-oriented methods [212, 213] (see Appendix A.1). In addition, responding to the questions starts pattern instantiation that allow pattern reuse in different domains as discussed in Appendix A.3.

For the equality with ***stakeholder variability*** value, two questions need to be answered;

a) Who are the stakeholders?

b) What are the differences between the stakeholders?

Despite stakeholder identification being common practice in requirements engineering [44], variabilities among each group of stakeholders are less addressed. Identifying the eligible stakeholders will help software developers in making trade-off decisions between the different sustainability domains requirements as noted by Lago, Koçak, Crnkovic and Penzenstadler [61]. As the proposed value pattern depicts that variabilities could refine or add goals and service, this is an important aspect of equality that cannot be neglected.

As this work comes under the umbrella of sustainability and values the creation and maintenance of good conditions in the community now and over the long-term [2, 3], it is important to not only consider direct and current stakeholders but also look at the big picture. This includes examining the systemic social effects of software. Moreover, consideration is given to: (a) the complete set of stakeholders whether they are weak or strong entities (with direct, indirect, intentional and unintentional use), (b) the long-term uses of software because many of social issues only arise after frequent use and (c) the widespread use of software that raises issues associated with the increase in the numbers of users [214–216].

For complete requirements, stakheloder identification is important [217] and examining all possible variability issues will help in identify a more comprehensive set of requirements.

We must note that the identified stakeholders need to correspond to the scope of the software project. In real situations, the identified list should be checked by the development team and project managers to decide eligible stakeholders to ensure completing the project within the allocated time and budget. Additionally, conflicts between sustainability dimensions might affect the final stakeholders list. This will depend on which dimension is more valued and pursued by the project managers, project development teams or even the project sponsors.

In this work, the onion model of stakeholders [218] (described in Appendix A) is also utilised during this stage.

The responses to those two questions can be represented by the template *[stakeholder] has a [difference/variability of..].*

In order to instantiate ***fairness to the stakeholders' goals*** value, two questions must be answered:

(a) What are the stakeholders' goals when using the software?

(b) Which goals are (directly, indirectly) affected by the use of software?

Allowing stakeholders to achieve their goals is bounded by the positive and negative effects as well as the scope of the software application.

In order to instantiate equality with ***services and facilities***, the following question must be answered:

(a) What are the system's functions/services and facilities needed for the stakeholders' to achieve their goals?

The response forms a requirement that can be presented through a template stating: *[stakeholder] should get [service] to accomplish [goal].* This template combines the responses to stakeholders' goals and the services required to achieve the goals. A similar idea was discussed by Dalpiaz, Souza and Mylopoulos [212]: "The association of a function to a goal is called *operationalization* in the sense that the function specifies how a goal can be made operational" (italicised in the original). In addition, operationalising goals helps in defining concepts in a measurable manner [212]. The following Figure (4.2) summarises the used questions to operationalise the values into requirements. This Figure is also incorporated in the next Chapter as part of the usage guidelines of the equality requirements elicitation method in Section 5.3. The equality value pattern and templates are means of 'operationalism'.

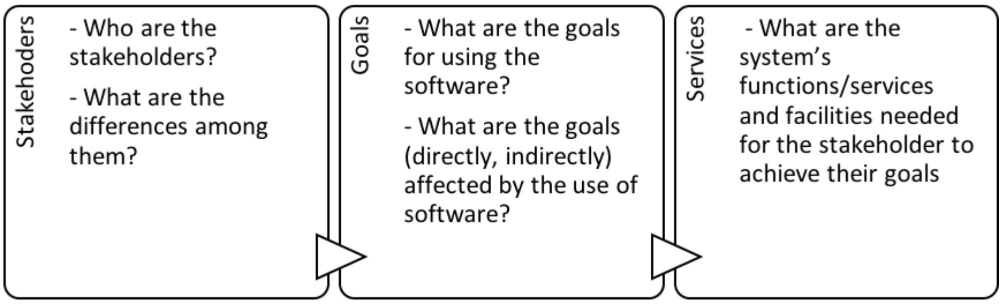


FIGURE 4.2: Operationalism of Equality Value Pattern

The textual templates can also be represented in tabular format. Liu and colleagues in [31] reported that the third top requirements representation (after non-UML and UML diagrams) is the tabular format representation.

Note that equality is not only achieved by considering stakeholders with variability factors and allowing them to achieve their desired goals. Equality is also achieved by equally allowing all relevant stakeholders (in accordance with the agreed software scope) to achieve their goals through the software. Thus, variability factor is an important aspect of equality but its not the only important as achieving goals (fairly selected) is also important.

Use of the requirements templates makes it possible to move social sustainability from the abstract soft-goal notion to concrete requirements based on a set of stabilised societal values. These can later be used as the basis for social sustainability measurement and assessments in software. This will be of considerable help to the requirements engineers who are unfamiliar with social sustainability.

4.4 Discussion

In this work, equality is considered at the requirements level. As it was explained above, the equality requirements are derived from equality values that are important for social sustainability. Thus, the identified requirements are categorised under value-based requirements. As we have mentioned earlier, the equality value pattern was built by reviewing and analysing literature on equality, equity, justice and fairness (see Section 4.1.1). So the backbone of the equality value pattern is composed of those concepts. With the equality value pattern, equity is met when different stakeholders with variability factors are considered during requirements identification. Fairness is met by recommending the use of fair means of selecting stakeholders goals to be implemented and supported in the software application. Equality as equal treatments is suggested by ensuring equal access to software services required.

Previous studies have revealed that variabilities among stakeholders can affect the designed software. The study in [198] revealed how gender and age affect stakeholders demand from ICT. For example, young women want to have software that allows them to organise their leisure time, manage their contact, manage their availability and share emotions.

Similarly, initiatives such as the Web Accessibility Initiative (WAI) [219] confirm the importance of considering variability factors.

As for the future, similar steps can be re-taken and applied with other social sustainability values such as social cohesion to understand and identify the software requirements needed to support the value.

In the value pattern, the aim is to provide access to services for those who need to achieve a goal and at the same time we want to remove any constraints that can prevent stakeholders from enjoying the benefits of using the software. Thus, the suggested value pattern is viewed as a bridging tool between early software development stages and the final usage stage because we are striving to avoid sustainability failure during use by considering sustainability issues at an early stage.

Furthermore, the pattern and template are viewed as a tool for ensuring equality among different stakeholder groups (specifically by the fairness among stakeholders' goals) as well as within similar groups of stakeholders (through variability among similar groups). This tallies with the idea in [220]:

> The incorporation of diverse concerns and perspectives of all stakeholders into the design process is central to PD [participatory design]. Wrapped up in these beliefs is an assumption of equality between stakeholders - not just between designers, developers, and end users, but also within these groupings.

Moreover, in our view, this study presents a form of positive action [165, 221] that moves us closer to socially sustainable software. Positive action initiatives are permitted when people with variability factors (protected characteristics) face disadvantages due to the factors, have different needs than those who do not have such factors or participate less in activities due to the factors [165]. The positive action initiative is to help them in reducing the disadvantages, meeting their needs or encaurage participation [165]. Examples of such initiatives includes involving black, Asian, and minority ethnic BAME [222] individuals in the workplace [223, 224] and involving women in higher education under the Athena SWAN initiative [225]. There is an argument that positive action initiatives can raise issues of reverse discrimination [226] also called positive discrimination [227]. Positive discrimination results from favouring individuals who have erlier suffered from discrimination [228] and reverse discrimination [229] is an equivalent term used. Reverse (or positive) discrimination implies that discrimination is against the majority and dominant group rather than the minority previously suffered from inequality [227, 230].

This work is a proactive attempt to better represent stakeholders and their variabilities in software requirements (early stages of software life cycle). At the same time, possible positive/reverse discrimination is avoided by also integrating stakeholders goals into requirements whether they poses variability factor or not. This will result in software products that accommodate and encourage more stakeholders to use and benefit from these products.

4.5 Threats to Validity

4.5.1 Construct Validity

In this study, a possible threat to validity could be raised by the coding as this is a subjective activity. For this reason, evaluations in Chapters 7 and 8 were carried out to check the trustworthiness of the findings. In addition, and as previously mentioned in Section 4.1.1, the initial coding was discussed and validated by a second coder (supervisor).

Confirmation bias [231] could also be a possible threat as the analysis was done in two rounds. It could be that the identified pattern is built on what the researcher thought to be right from the first round. To mitigate the confirmation bias threat, the coding did not only rely on the pre-defines codes but it also allowed new codes to emerge inductively as described in Section 4.1.1.

Another threat could be related to reporting the results [232] (using the generic sustainability model by [62, 63]) which is relatively new. To mitigate this threat in Chapter 5, Section 5.1, we will use a widely known and common representation of a requirements pattern to present our results.

4.5.2 External Validity

External validity could be threatened by the method of paper selection. Papers were selected and limited to the results of the literature review (Chapter 2) which were about social sustainability and not about equality per se. So, we could have missed key references on equality. In future, this study can be altered to include snowballing procedures where the selected set of papers are further investigated to a) examine the reference lists to identify additional and relevant equality papers to be included in the study and b) find additional equality papers that have citied the selected set of papers using Google Scholar. Then, the additional set of papers are analysed in the same way done in round 2 of the study (summative content analysis, Section 4.1.1).

4.6 Summary

This chapter presented the approach followed to build the equality value pattern. The approach comprised qualitative analysis of a set of 37 social sustainability papers derived from the structured literature review (Chapter 2).

As a result of the chosen approach, an equality value pattern and templates were built. These could be used by software practitioners for social sustainability requirements identification.

The equality value pattern is perceived as a stepping stone towards exploring social sustainability and its effects on software domain.

Chapter 5

Equality Requirements Elicitation Method

Chapter 4 described the development method of equality value pattern. The pattern was a result of an in-depth review of existing literature to identify equality values. Those values are then operationalised into equality requirements templates.

In this Chapter, we first present the equality value pattern (discussed in Chapter 4) using insights from the anatomy of a requirement pattern discussed by Withall [33] in Section 5.1. Then in Section 5.2 we suggest a method of equality requirements elicitation using the equality value pattern. This is followed by usage guidelines (Section 5.3) to assist requirements engineers in identifying and writing equality requirements. Next, discussion on possible work to be build upon the equality value pattern is presented. A possible integration model between equality and its constituents values and software quality attributes is discussed in Section 5.4.1. Then, we present a possible strategic dependency model depicting the role of stakeholders in achieving equality in Section 5.4.2.

5.1 The Anatomy of Equality Value Pattern

The equality value pattern is a requirement pattern. It contains:

1. **Pattern Name:** Equality value pattern

2. **Applicability:** Use the equality value pattern to specify equality supporting requirements. This is used as a supplementary materiel to the equality value pattern diagram (see Figure 4.1).

3. **Content:** An equality requirement contains the following:

   (a) **Stakeholder:** represents an entity that is interested in a software or that exert an influence on it [36]. It is important to identify related stakeholders as this will allow equally considering all/majority of them while designing the software and producing a software that supports social sustainability. Stakeholders identifications models are to be used for this e.g, the onion model of stakeholders [218] (see Table 5.1).

   (b) **Variability(s):** represents natural/human factors (e.g. disability, age, gender, etc.) as well as material factors (e.g. income) that could differentiate stakeholders. Neglecting these factors during software development could result in inequality of stakeholders that have such factors. Variability check-list (Table 5.2) is suggested to be used to identify related variability factors and to ensure completeness.

   (c) **Goal(s):** describe each stakeholder's desired benefits. Goals/benefits are achieved from the software usage (directly or indirectly).

   (d) **Services and facilities** describes how the software will facilitate the accomplishment of the previously identified goals.

4. **Templates:** ≪Stakeholder≫ ≪with variability≫
   should get ≪function/services/resources≫ to accomplish ≪goal≫

5. **Examples:**

   (a) Investors (users) with low investment management knowledge should get access to on screen hints/tips to accomplish the goal of using the application regardless of knowledge level.

   (b) Viewers with visual impairment should get access to alternative information views (text, audio, video, etc.) to accomplish the goal of supporting disabled users.

   (c) Instructor should get access to the records of training activities to accomplish the goal of following up trainees' progress.

As other requirements templates [33], the equality value pattern and as described above does not force a specific RE process (i.e. traditional or agile). It can be used with any RE process already in use. The equality pattern and at any point where RE process does stakeholder identification, asks what is the possible variability? For each variability as any requirement is defined, the pattern asks: how can this variability be supported in achieving the goal that the given requirement supports and what additional services would be needed?
All else is handled as per the used RE method. We do not make any demands on how conflicts will be identified, or resolved, or how cost will be managed. All these are part of the used RE process and not part of the pattern.

5.2 A Method for Eliciting Equality Requirements

Our equality requirements elicitation method is built on the equality value pattern (in Section 5.1). The method here is to expand the ongoing RE activity in place by asking additional questions at set points, and writing up answers into a template form. The method is comprised of four activities (depicted in Figure 5.1). The method activities are explained next.

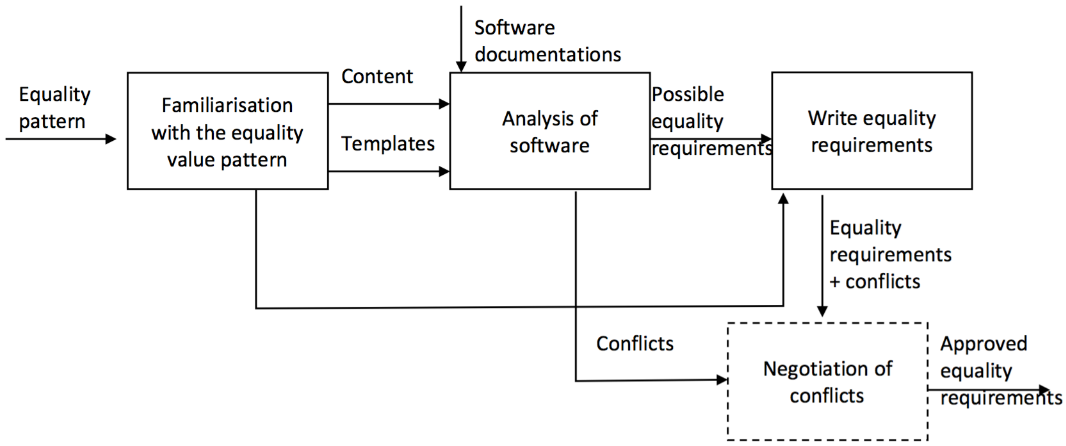


FIGURE 5.1: Equality requirements elicitation method

1. First, become familiar with the equality value pattern (including the pattern diagram in Figure 4.1). Understand what needs to be said (pattern content) and how (pattern templates). This is denoted as familiarisation with the equality value pattern activity in Figure 5.1.

2. Next, analyse the software under investigation. This includes studying software documentations or relevant descriptions. At this stage, use the equality pattern to identify existing equality requirements or recommend new. Its recommended to utilise Alexander's onion stakeholders taxonomy [218] (see Table 5.1). Also, variability check-list is recommended (see Table 5.2). This is indicated as analysis of software activity in Figure 5.1.

3. Then, use the templates in the equality value pattern to guide you in writing equality requirements.

4. Negotiation of conflicting requirements and issues (such as eligible stakeholders to be considered, conflicts with other sustainability requirements, etc.) is a recommended activity. This activity is done by using a negotiation technique chosen by the requirements engineers. The aim is to achieve an agreement on a set of equality requirements. Note that this step is out of the scope of this research, thus, it is indicated using dashed line rectangle (see Figure 5.1). The output will be an approved list of equality requirements prepared by the requirements engineer and to be used during design and implementation of the intended software product.

Table 5.1: Stakeholders List

Type of Stakeholder	Description
Normal Operator	Do routine commands, entering and monitoring output of product. Communicate with the maintenance operator and operational support as well as functional beneficiaries (e.g. providing them with processed information, and receiving instructions from them).
Maintenance Operator	Product maintenance (hardware, product faults)
Operational Support	Help desk, trainers (help and training normal operators)
Functional Beneficiary	This type benefits from the output/result created by the product. They contact the operators.
Interfacing System	This represent other systems that interface with the product.
Purchaser	Product manager on behalf of consumers or procurement.
Product Champion (aka Sponsor)	The product champion is critical from before the start of a development, and remains important throughout. The role does not necessarily or even desirably contribute to product requirements: it functions mainly at a political rather than a technical level.
Negative Stakeholder	Anything/one that can be harmed by the product (financially, physically, etc.) and they can harm the system. E.g. householders living close to the route of a planned railway.
Hostile Agent (type of negative stakeholders)	Any role that actively seeks to hinder or harm the development and operation of the System. 'Actively' means using some degree of intelligence and creativity to oppose the System. Examples include military enemies, political and commercial spies, hackers, spammers, virus writers, thieves, fraudsters. Clearly the degree of harm intended by such agents varies from complete destruction through malicious pleasure to unauthorised acquisition of assets (with essentially unintended harm as a side-effect).
Political Beneficiary	Any role in public office or private business that can benefit in terms of power, influence and prestige through the success of the Product.
Financial Beneficiary	This type gets financial gains from the product success.
Regulator	Governmental or other regulators, e.g. ISO
Developer	Develop the system or undertake maintenance role in maintenance contract.
Consultant	From outside the development organization.
Supplier	Responsible for components of the products.
Source (I. Alexander, "A Taxonomy of Stakeholders. Human Roles in System Development ", 2005)	

TABLE 5.2: Stakeholders Check-List

Stakeholders Variability Check-List

Below is the check-list of variability factors that can assist you in identifying relevant equality requirements. Please check if any of following variability factor(s) affect stakeholders identified earlier.

Variability list:

1. Age
2. Current/potential stakeholder
3. Disability
4. Education/Knowledge level
5. Gender
6. Income status
7. Language
8. Location
9. Position/status
10. Race
11. Religion
12. Technical literacy level
13. Technology used
14. Others

5.3 Usage Guidelines for The Equality Requirements Elicitation Method

The following instructions are intended to guide requirements engineers for easy elicitation of equality requirements. This is a guideline to allow users start utilising the elicitation method (using the pattern). However, requirements engineers repetitive usage of the method and their previous experience in using requirements patterns can form new/updated way of usage [34].

1. Familiarise yourself with the equality value pattern. You will need to understand what is it about, what are the specific elements needed in the pattern and how will you use the templates to write the requirements.

2. Study the stakeholders list (Table 5.1). Understand the different categories of stakeholders.

3. Study the variability factors list (Table 5.2).

4. Go through software documentations. Identify the existing stakeholders. Think of any missing stakeholder that can be added to ensure that equality is achieved (use the stakeholders list in Table 5.1). Ensure that different stakeholders types are represented. For example, in dispute management software, all dispute parties need to be part of the software application. Another example could be indirect stakeholders who will be indirectly affected by the system. For example, in a game application, the parents of a player are indirect stakeholders who are affected by their child's addiction to the game. Engineers need to consult the development team and sponsors to identify and reconcile conflicts.

5. For each stakeholder, look for existing variability factors. Also, look for missing factors to be suggested. Variabilities can be identified by the definition of persona (role profiles) in agile framework [234]. In software documentations, this can be found (explicitly or implicitly mentioned) in the non-functional requirements. Use the check-list in Table 5.2.

6. Based on identified stakeholders and their associated variability factors, look for existing equality requirements. If equality requirements are not existing, look for goals to be achieved by the stakeholders (existing or suggest new).

Ensure that whether stakeholders possess variability factors or not, goals need to be identified. Goals can be directly or indirectly affected by the use of the software. A direct goal would be to safely travel using a travel application. An indirect goal can be related to the importance of maintaining social relationships with a game player to the game's player family.

7. Based on the identified goals, identify how the stakeholders achieve the goals through the software's services, functions and facilities. Are those services, functions and facilities existing? Do you need to add new ones?

8. Rethink. Does any variability factor affects exiting goals and software's services, functions and facilities. For example, a user's visual impairment implies the need for alternatives to information display (e.g. audio).

9. Rethink. Does the variability factor of one stakeholder affects the services related to another stakeholder? For example, in a training application and because of trainers different languages, online tutorials prepared/uploaded by trainers and help desk staff need to be customised and displayed in different languages.

10. Use the equality requirements templates to write the identified equality requirements.

11. Consult the development team and project sponsors on the equality requirements and conflicts. Negotiate to arrive to an agreement on equality requirements.

12. Prepare an approved equality requirements list to be included in the software specification document and then to be used for design and implementation.

5.4 Discussion

5.4.1 Equality and Software Quality Attributes

In this section, we discuss and illustrate a possible integration between equality and its related values (as a sustainability issue, see Figure 4.1) and software quality attributes (ISO/IEC 25010 [42]). Nevertheless, this is not an easy or a clear-cut task. This is due to the fact that in the equality value pattern and its templates, we use the three equality sub-values in writing an equality requirement (see Section 5.1.)

In the proposed integration model, we used the i* notation to depict dependencies. Similar to Cabot et al. [68], the proposed dependencies model in Figure 5.2 is domain independent. We also treat sustainability and equality values as softgoals.

The integration model suggests that to ensure that the diversified software stakeholders are supported and their differences are respected, a software application needs to be usable, portable and compatible. Additionally, allowing software's stakeholders to be fairly considered by allowing them to achieve their goals (agreed on) can be achieved by developing a software that supports functional stability, performance efficiency and security. A software is functionally stable when it "provides functions that meet stated and implied needs when used under specified conditions" [42]. Doing so to all eligible stakeholders helps them in achieving their goals (fairly selected and agreed on by negotiation as discussed in Chapter 4, Section 4.1.2) as well as accessing the facilities to achieve their goals. A software with security issues could hinder stakeholders from achieving their goals (if this happens, some affected stakeholders will not achieve what they want while others can still achieve their goals). For example, an insecure banking application could stop some of the bank customers from successfully paying utility bills while others can still manage. Even worst, it could lead to cyber crimes or frauds (negative goals that stakeholders want to eliminate rather than achieve). Reliable and maintainable software applications ensures that stakeholders are able to access the needed software resources [42]. Doing so will support all users access the software.

5.4.2 Stakeholders Role in Equality

In this section, we model social sustainability (i.e. equality) and illustrate possible relationships between stakeholders (Table 5.1) in light of equality. This is done with the inspiration from the strategic dependency model (i* framework) [233]. This strategic model in Figure 5.3 is an abstract model that can be used with different software projects. It can be used to highlight the roles of software stakeholders in achieving equality supportive software.

Starting with the operator, he/she depends on the developer to build a software that supports equality. After having a software that supports equality, an operator depends on maintenance personnel to keep the software functioning as desired (i.e. supporting equality). An operator will also depend on operational support staff to ensure that training materials are customised to fit equality (i.e considering operator's variability factors). A developer needs a supplier to provide equality supporting equipments (e.g. Braille printer). Additionally, a developer needs a regulator to enforce equality by laws and regulations. This standardise the ways of addressing equality. A regulator depends on a developer to produce software products complying with equality regulations. Functional beneficiary depends on operators on equally accessing services and information they need to achieve their goals (which were already fairly selected).

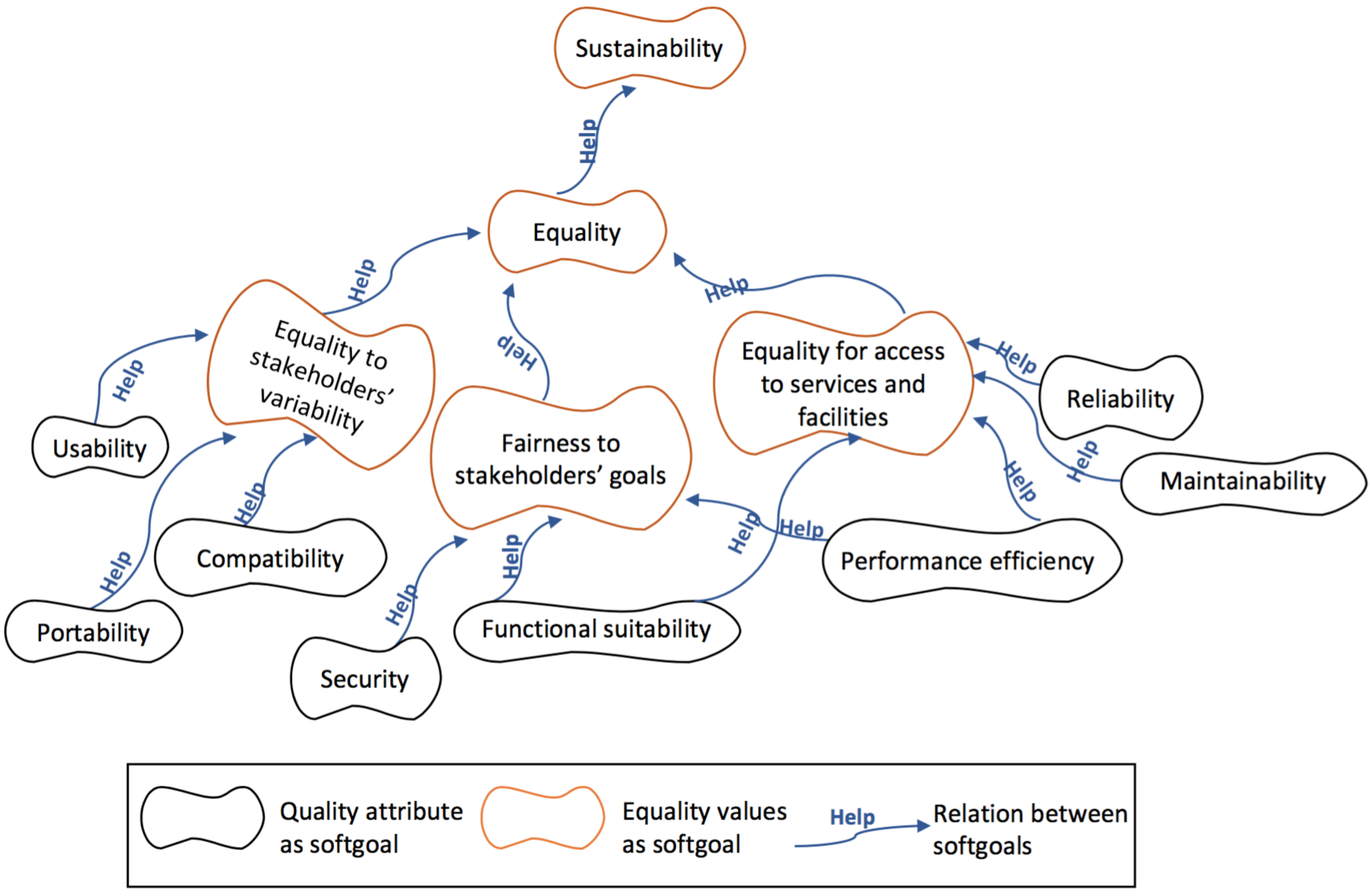


FIGURE 5.2: Equality and Software Quality Attributes

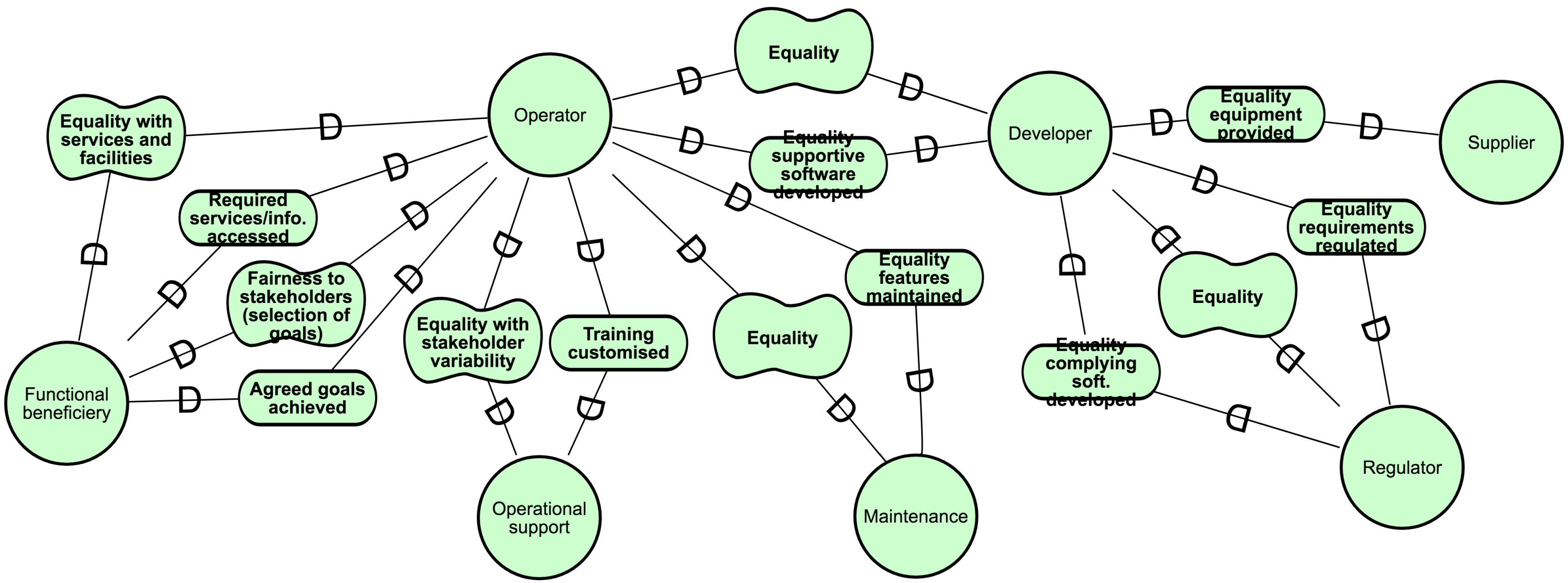


Figure 5.3: Strategic Dependency Model for Equality

5.5 Summary

In this Chapter, the equality value pattern was presented using requirements pattern anatomy. Then, we proposed a method of elicitation of equality requirements and provided a usage guideline. We also discussed a possible relationship model depicting dependencies between equality and software quality attributes. Finally, the potential role of stakeholders in supporting equality was introduced. All the proposed and discussed artefacts are to assist software engineers in their responsibility of integrating social sustainability concerns in software domain and identifying equality requirements.

Chapter 6

Equality Value Pattern Application

Having elicited the value pattern in Chapter 4 and proposed an elicitation method in Chapter 5, this chapter demonstrates the capability of the proposed elicitation method in integrating social sustainability in software through operationalising the social concerns into software requirements. In this chapter, the applicability of the equality requirements elicitation method is provided through seven examples. The study is outlined in Section 6.1 and the application of the equality the requirements elicitation method is presented in Section 6.2. A small scale survey is presented in Section 6.3 followed by common equality requirements in Section 6.4. Additionally, a discussion on how the pattern fits within the agile software development methodology is demonstrated in Section 6.5. Finally, threats to validity are presented in Section 6.6.

6.1 Study Outline

The proposed method was applied to

1. a set of examples for which software documentations are freely available and open for validation.
2. agile software engineering (in Section 6.5[1]).

Because software is embedded in many aspects of life, applicability of the value pattern (and the associated elicitation method) in diverse domains implies that the pattern and the method are reusable and domain-independent.

The examples were studied by the researcher. The researcher then applied the elicitation method to identify the equality requirements for all the examples.

In addition, the equality value pattern was shared with another researcher [234] who worked on integrating the equality value pattern and templates in a process model for agile requirements management.

The questions addressed in this study are as follows:

$RQ1$: Does the use of the equality requirements elicitation method facilitates equality requirement identification?

$RQ2$: Are the identified requirements relevant to equality and in turn social sustainability?

$RQ3$: Is the equality pattern usable with current RE practices?

[1]This study is a independent study by Monica Bahl [234] built on the equality value pattern and templates

6.2 Application to Existing Requirements Documentations

6.2.1 Selection Criteria

The software documentations were selected based on the following criteria:

1. Written in English. English is one of the most spoken languages in the world.
2. Sourced from a reputable research or software practitioner organisations.
3. Representative of the RE documentations (e.g. textual documentation, use cases).
4. Prepared by practitioners independently from our study.
5. Originating from different domains such as health, travel, etc.

6.2.2 Study Method

The application of the value pattern and templates includes

- Selecting a corpus of studies, following the criteria discussed in Section 6.2.1.
- Applying the equality requirements elicitation method and the guidelines presented in Chapter 5.
- Conducting a small-scale survey with experts that integrates findings from the Smart Garden example.

The selected examples for this study are the Health Watcher (HW)[235], Virtual Art Viewer (AV)[236], Travel App (TA)[237], Campbell Prediction System (CP) [238], Arcade Game Maker (AGM)[239], Personal Investment Management System (PIMS) [240] and Smart Garden (SG)[241]. The examples are summarised in Section 6.2.3 and in Table 6.1.

6.2.3 Subject Software Systems

Health Watcher (HW) is a system developed to allow citizens to register health complaints against organisations such as restaurants [235, 242]. In turn, health institutions are to use the system to investigate such complaints as well as to disseminate health-related information to the population. [235, 242].

Virtual Art Viewer (AV) is an application that allows people to explore a wide range of paintings and find information about any piece of art listed on it [236]. According to the project description, system administrators create, edit and store digital representations of paintings and information about these paintings [236]. Interested users can search for the listed paintings, view them and print the ones they like [236].

Travel App (TA) is an application in which travellers act as information agents and share their travel experiences [237]. Such experience reporting helps other travellers to structure their trips, connect with other travellers, and helps to enhance public services [237].

Campbell Prediction System (CP) is a training tool and decision support system based on the Campbell Prediction System methods [238]. The aim is "to compute, project and visuali[s]e the potential fire behaviour, trigger points and alignments-of-forces on the fire-ground." [238].

TABLE 6.1: Subject Software Systems

Example	No of functional Requirements	No of non-functional requirements	Domain
Health Watcher (HW)	9 use cases	9	Health
Art Viewer (AV)	11	19	Art
Travel App (TA)	11 - 5 use cases	7	Travel
Campbell Prediction System (CP)	not complete document but with context data flow diagram 2 use cases diagrams	4	Fire force training
Arcade Game Maker (AGM)	12 use cases	4	Educational game
Personal Investment Management System (PIMS)	21 use cases	6	Investment management
Smart Garden (SG)	11	8 (quality requirements)	Gardening

Arcade Game Maker (AGM) is a set of single player games [239]. These games are designed to assist with the learning and application of the software product lines method [239].

Personal Investment Management System (PIMS) is a single user application that allows investors to manage their investments in different institutions [240]. It is mainly a bookkeeping application [240].

Smart garden (SG) is a software application for home gardeners [241]. The application is designed to control water consumption [241].

6.2.4 Results of Application

The results presented in this section[2] address the first research question:

$RQ1$: Does the use of the equality requirements elicitation method facilitates equality requirement identification?

As the researcher is already familiar with the equality value pattern, the second step of the elicitation method (fourth point in the guidelines) commenced. Here, the onion model [218] (Table 5.1) is utilised. In addition, the full set of weak and strong stakeholders (direct, indirect, intentional and unintentional use), the long-term use of software as well as the widespread use of software [214–216], were considered while identifying the stakeholders [166]. This resulted in identifying a wide set of stakeholders including new previously omitted ones [166].

For example, in the Health Watcher system, food standard agency, animal protection agency, environmental agency and local businesses such as restaurants were identified (see Table 6.2). Stalker, help desk and trainer are examples of additionally identified stakeholders in Travel App (see Table 6.2). For Campbell Prediction System, training organisation, and an instructor's family/friends are examples of additional stakeholders (see Table 6.2). Similarly, for the Arcade Game Maker, examples of the additional stakeholders include player's family/friends (see Table 6.2). Artist, thief, gallery administrator and help desk and trainers are examples of those missing in the Virtual Art Viewer (see Table 6.2). In the Personal Investment Management System, trainers, investor's family/friends and hacker are additional stakeholders who were found to be relevant (see Table 6.2). In the Smart Garden, plantation authorities and pesticide control authorities are examples of relevant stakeholders (see Table 6.2). The full list of stakeholders (provided in the documentation or newly identified) is in Table 6.2.

[2] This part was partially presented at the 4th International Conference on ICT for Sustainability in 2016 [166]

An interesting finding was that even with simple stand-alone software with a single user such as the Personal Investment Management System, there are indirectly affected stakeholders such as the investors family and friends.

Equality requires that all interested and affected stakeholders are identified. This leads to equally considering them while eliciting the software requirements.

Next the related variability characteristics were elicited (in Table 6.3). The listed factors are the most relevant ones to case under investigation. Practitioners are advised to use the variability check-list (Table 5.2) at this stage to assist them in identifying possible variability factors (as per point 5 in the usage guidelines in Chapter 5). In real word situation, the relevant factors will be investigated and confirmed from actual stakeholders.

Interestingly, although equality or social sustainability were not explicitly discussed in the documentations under study, existing services embedded consideration of variability factors. For example, the multilingual interface to accommodate language differences among country partners who will replicate Travel App [237] is related to the language variability factor (see Table 6.4). Another example is the service of the help function and short tutorial guide in the Virtual Art Viewer system [236] which relates to the level of technical literacy (see Table 6.4). In addition, running the system on different platforms (UNIX and Windows) [240] relates to the technology variability in Personal Investment Management System (see Table 6.4).

Equality is achieved by defining variability factors that differentiate stakeholders groups. This will lead to identifying refinements or additions of goals and services to accommodate their variability factors.

Table 6.2: Stakeholders in Subject Software Systems

Example	Stakeholder	Provided in documentation	Newly Identified
Health Watchers	Citizen	☑	
	Staff	☑	
	Local business		☑
	Trainers and help desk		☑
	Tourism department		☑
	SSVS (Interfacing System that will communicate with the health system)	☑	
	City Hall	☑	
	Company X (consultant)	☑	
	Developer	☑	
	Maintenance	☑	
	Environmental agency		☑
	Financial institution		☑
	Animal protection agency		☑
	Food standard agency		☑
Virtual Art Viewer	Website Admin.	☑	
	Viewer	☑	
	Apogee Arts (sponsor)	☑	
	Paragon Software (maintenance comp.)	☑	
	Help desk and trainer		☑
	Gallery administrator		☑
	Art exhibition organizer		☑
	Thieves		☑
	Artist		☑
	Consultant	☑	
Travel App	Traveller	☑	
	City planner		☑
	Marketing agency		☑
	Help desk, trainers		☑
	European Commission	☑	
	Replication partners	☑	
	Stalker		☑
Campbell Prediction	Trainee	☑	
	Instructor	☑	
	Community planner	☑	
	Domain Expert	☑	
	Training organization		☑
	Trainee's family/friends		☑
	Instructor's family/ friends		☑
Arcade Game Maker	Player	☑	
	Sponsoring company	☑	
	Administrator		☑
	Players family/friends		☑
Personal Investment	Investors	☑	
	Trainers and help desk		☑
	Investors family/friend		☑
	Thief/hacker		☑
Smart Garden	Plant nursery (as garden vendor)	☑	
	Gardner/user	☑	
	Equipment supplier	☑	
	Pesticide control authority (for home gardens)		☑
	Plantation authority		☑
	Water and irrigation authority/ministry/department of water resource	☑	
	Environmental charity/authority/ministry/green organisation	☑	

Once the stakeholders and the variability factors were identified, we moved to the next point of the guidelines, identifying stakeholders goals. At this point, we attempted to address the goals behind/for using the software or/and the goals (directly/indirectly) affected by the use of the software.

We noted that variability factors can result in refinement of existing goals/services. For example, impaired visual ability of a citizen in Health Watcher system will result in a service of displaying information in different formats (audio,video, etc.). We view this as an added service that cause refinement of the services that are already available in the system (see Table 6.4).

On the other hand, variability factors can introduce new goals/services. For example, in Travel App, a traveller with age variability (i.e. child) requires having a service of auto-delay for movement information posting for children's accounts in order to ensure their safe movement around the area (see Table 6.4). Other instances include the Health Watcher system where citizens' religion variability introduced a service of food certifications and in the Travel App system it introduced religious filters for locations and events to allow travellers to plan religious travel (see Table 6.4). Similarly, in the Smart Garden system, age was identified as a variability factor for gardeners. Consequently, a goal to support customisation of the display fonts was defined to allow improved readability (see Table 6.4).

We identified a set of goals from the requirements documentations. Some goals were elicited from the software functionalities, others from the software quality features (non-functional requirements) and others from the provided general description. Where new stakeholders were identified that were not previously discussed in the requirements documentation, goals related to these stakeholders' key interests were also listed. This is the case, for instance, with the goal of mass training within

Table 6.3: Stakeholders Variability in Subject Software Systems

Example	Variability	Provided in documentation	Newly Identified
Health Watchers	Technology used	☑	
	Religion		☑
	Disability		☑
	Language		☑
	Age		☑
	Gender		☑
	Information media		☑
	Storage medium	☑	
Virtual Art Viewer	Technical literacy level (Novice user)	☑	
	Language		☑
	Technology used		☑
	Disability		☑
Travel App	Technology used		☑
	Disability		☑
	Information media		☑
	Age (child)		☑
	Language	☑	
	Religion		☑
Campbell Prediction	Technology used	☑	
Arcade Game Maker	Technology used	☑	
	Language		☑
Personal Investment	Technology used	☑	
	Investment Management knowledge level		☑
	Technical literacy level (expert user)		☑
Smart Garden	Age		☑
	Language		☑
	Technology used		☑
	Current/Potential stakeholder		☑

less time and effort for help desk and trainers stakeholders in the Travel App system. Similarly, the goal of preparing marketing plans for marketing agency (new stakeholder) in the Travel App system.

As a result of adopting the stakeholders list [218], a number of hostile stakeholders were also identified, such as hackers in Personal Investment Management System, Virtual Art Viewer and Travel App systems (see Table 6.2). In the Personal Investment Management System, a hacker's goal is to gain investors' financial information and use it for frauds or blackmailing. An opposite positive goal (to the investor) is

TABLE 6.4: Excerpt of Equality Requirements

Subject system	Stakeholder	Variability	Goal	Services/function
Health Watchers	Citizen	Religion	Get religious verification (e.g., on kosher, halal food) (N)	Provide religious verification (N)
Smart Garden	Gardner (E)	Age (elders) (N)	Simple and easy to use application (N) Improve readability (N)	View help function and tutorials (N) View help instructions 'how to install and fix' information displayed as textual, audio and video instructions (N) Customize display font/Select font size (N)
Virtual Art Viewer	Viewer (E)	Technical literacy level (Novice user) (E)	Usability (E)	Help function with short tutorial guide on how the AV can be used (E)
	Artist (N)	-	Maintain relationship with the public (N)	Access to viewers comments on (his/her) paintings (N) Note: This will lead to an additional service to viewer which is add comments to paintings. (Completeness of requirements)
Travel App	Traveller (E)	Language (E)	Multi-lingual support (E, NF)	Provide language customization (at least English, Turkish, Finnish, and Italian because of replication partners) (E, NF)
		Age (child) (N)	Safely move around the area (N)	Auto-delay the movement information posting for child account (N)
		Religion (N)	Plan religious travel (N)	Filter religious locations/events (N)
	Help desk, trainers (N)	-	Mass training with less effort and time (N)	On screen hints/tips (N) Online help (N) Online tutorials (N) in different formats such as text, video, audio, speech, language [traveller variability]
	Marketing agency (N)	-	Prepare marketing planning (N) Involve travellers in future marketing plans (N)	View busiest route reports (N) View active travellers (more post, likes) (N)
Personal Investment	User (E)	Technology used (E)	Use the provided services regardless of technology used (N)	Run on UNIX and Windows based Platform (E, in description)
	Thief/hacker (N)	-	Get financial information of the user, can be used for later blackmailing or acquisition (N)	Encryption to avoid security breach. (N, put more emphasis on the importance of security goal of a user)
	Investors family/friend (N)	-	Maintain social relationships (N)	Usage time alarms/notification (N)

to secure their financial information. These two opposite goals will lead to the same service of encryption (see Table 6.4).

This implies that equality is also about protecting the goals of the legitimate stakeholders.

Questioning the goals (indirectly) affected by the use of the software provided the goal of maintaining social relationships in the Personal Investment Management system (see Table 6.4). This goal was identified as important to the investors family/friend. They will not be using the system (directly), however they will be affected by the long usage of the system by their family member (i.e. investor). This goal introduced a new service of displaying usage time alarms or notifications to the investor about the time spent on the software.

Stakeholders have diverse goals. Hence, it is apparent that it will be necessary to select the goals to be satisfied through the software in a fair manner (as suggested in Chapters 4).

Here, equality is achieved through considering all stakeholders groups and for each group, the most agreed on goals (direct or indirect) are to be implemented in the software (fair selection) as it is not practical to implement all the goals due to time and cost.

The seventh point of the guidelines required determining a set of services to support the previously identified goals.

Due to the diverse language and physical (dis)abilities considered, services such as customising language settings (HW, TA, AV, AGM) and providing voice-based and braille interfaces may be necessary e.g. HW and TA [166]. Such services are necessary to ensure that variability aspects will not disadvantage the diverse set of users in achieving the (agreed upon) goals they expect to accomplish through the software [166]. Furthermore, we noticed that a variability of one stakeholder can affect the services related to another group of stakeholders. For instance, in the Travel App, the trainers' service of providing online tutorials is refined by the variability of the travellers language. In this case, the online tutorial is refined to include tutorials in different languages (see Table 6.4). This emphasises the importance of re-examining and amending services in relation to variability factors (point 9 of the guidelines).

Besides, we observed that a service can introduce another service. In the Virtual Art Viewer, a service to allow artists (stakeholder) to view comments about their paintings was identified (see Table 6.4). These comments are supposed to be generated by viewers (stakeholders). Prior to this, allowing viewers to add comments about pieces of art was not part of the identified requirements. Organising services and functions in this format helps in identifying missing and related services. This allows requirements engineers to check requirements completeness. Requirements patterns helps in identifying missing requirements [33] (see Appendix A.3).

Here, equality is achieved by allowing all stakeholders to access and use the service(s) required to achieved the previously agreed on goals.

The next step of the elicitation method is to write equality requirements. Here, we used the tabular format of the pattern (Table 4.6 to represent equality requirements (see Table 6.4).

The elicitation method using the equality pattern was done manually by the researcher. In future, a software requirements tool is to be developed to facilitate the use of the elicitation method and pattern. The tool shall allow maintaining social sustainability patterns (catalogue that will evolve from further researches) and requirements. For that the tool will display the social sustainability catalogue to the requirements engineers and they will be able to select the value patterns applicable to the project under investigation. Requirements engineers will be allowed to add different projects and associate them to the applicable patterns. Based on the selected pattern, the tool will display the pattern information. This can be for example by providing different tabs for the different pattern elements (e.g. applicability, discussion, content, template(s), etc.). For the template element, the tool can prompt the engineers with the specific elements of requirements temples and asks them to fill the required information. Then the tool will generate requirements statements. The tool shall be able to track requirements and allow modifications. The tool shall also allow engineers to view previously identified social requirements and when applicable, re-assign them to another project. This will facilitate reusability of requirements. The tool can also prepare reports on which social pattern is applied most or least to help identifying the usefulness of the patterns. Having a tool to support social sustainability patterns and requirements elicitation can increase the rate of adopting the patterns and methods in RE practices [49].

6.3 Experts Survey Results

The results in this section address the question $RQ2$: Are the identified requirements relevant to equality and in turn social sustainability? (Please refer to Appendix C for the full questionnaire used in this study.)

Five experts in requirements engineering and systems analysis were invited to evaluate the identified equality requirements survey for the Smart Garden system. Convenience and purposive sampling [243, 244] were used. The feedback included the relevance of requirements to equality using a scale with the following options (Directly and highly relevant, Definitely relevant, Probably relevant, Possibly relevant - likely of indirect relevance, Possibly not relevant, Probably not relevant, Definitely not relevant). In addition, experts were asked to justify and explain why a requirement is being considered as related or not being related to equality. The frequencies of statements ranking are dipected in the diverging stacked bar [245] (see Figure 6.1). Qualitative analysis [169] of the reasons behind categorising requirements as relevant or not relevant to equality was adopted. Furthermore, experts were asked to address any missing equality requirements that they thought may be relevant.

6.3.1 Statements Relevant to Equality

Generally speaking, all the identified requirements were related to equality for at least one reviewer. The only requirement that was rated as being not relevant by four of the reviewers was (number 31 in Appendix C) about allowing the "planting authority" (using the smart garden system) to prepare reports on plants to be removed or added per garden or area. We note that our poor choice of term in this statement is the key reason for this, as the term "plantation" has a very specific meaning of a large farm for growing crops. For the smart garden context a term "plot" would have been appropriate.

As shown in Figure 6.1, statements 21-28, statements 12-16, statements 9-10, and statements 2 and 3 were found to be decisively relevant to equality by all experts. Statements 28 and 27 are related to equipment suppliers. Statement 28 is concerned about potential supplier that we have used this as a variability factor and statement 27 is about current suppliers. Statements 26-21 are concerned about gardeners with different variability factors i.e. information media, age, hardware and language. We observe that the statements that address features/functions that allow diverse stakeholders of the software to utilise the software are highly considered relevant to equality. This indicates that equality is perceived to be achieved by equitable benefits from the software.

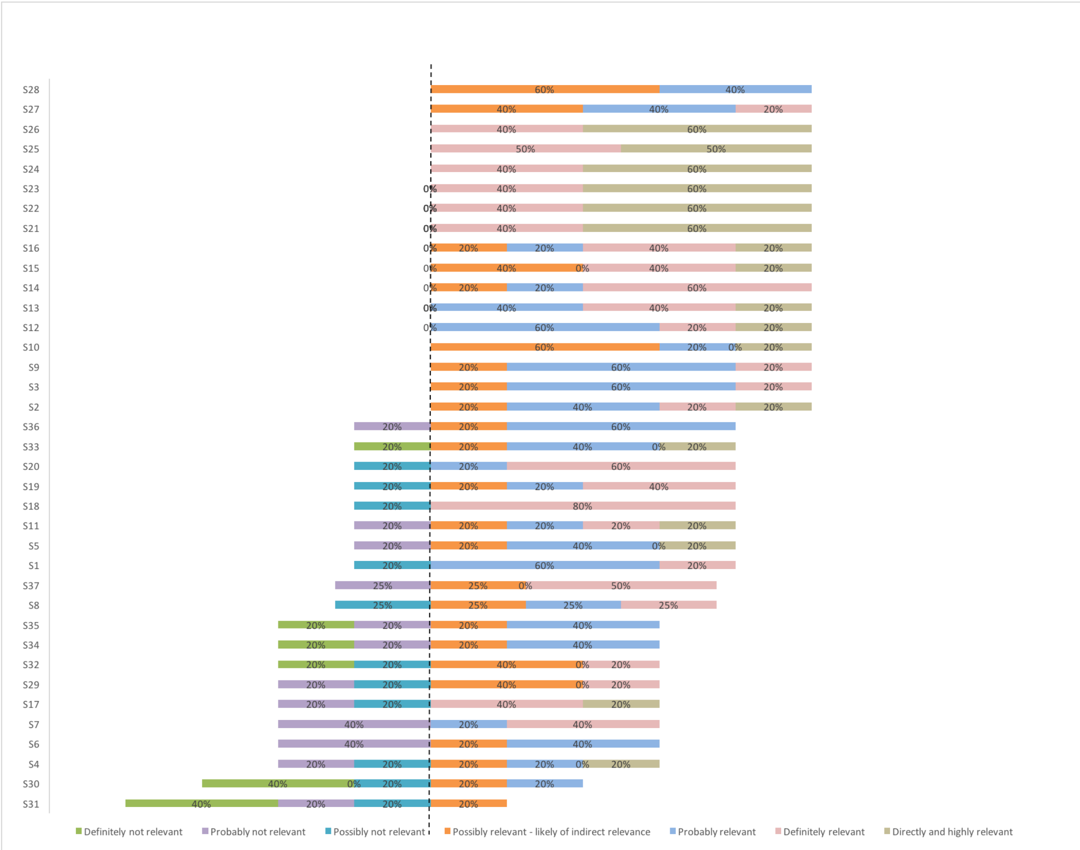


FIGURE 6.1: Smart Garden Statements Evaluation/Ranking

6.3.2 Relevant Requirements to Equality

According to the experts' points of view, the most frequent justification for ranking a requirements as relevant to equality is to ***allow participation and knowledge management***. The knowledge management category is an aggregation of sub-categories; i.e. update (according to changing circumstances or due to variations in water needs according to plant(s) type, location, etc.), share, improve general knowledge, equal access to information and reduce gaps among users (Table 6.5).

Inclusion, accessibility and usability were grouped under ***participation and empowerment*** [246, 247] (Table 6.6). In addition, experts evaluated requirements that imply or suggest that different stakeholders needs and interests are taken into consideration as relevant to equality (see Table 6.5). Privacy as protected need was also thought of as justification for relating a statement to equality (see Table 6.5). Experts have also justified relating requirements to equality if the requirement is important to groups of stakeholders other than the main group; i.e. the gardener (see Table 6.6 for examples of Relevant to several stakeholders). Statements under this category are 9, 12, 13, 16, 22, 25. Experts have also rated statements relevant to equality when they relate it to ***regional equality*** (1 respondent justified S1, 29, 30, 31). Here, regional equality is to consider all regions/areas in public plans (see Table 6.6). Additionally, requirements that illustrate public benefits are also rated as relevant to equality (see Table 6.6 for examples). Requirements that allow ***adaptation to external changes*** (R7: allowing the gardener to set an auto watering function and R8 allowing the gardener to cancel the auto watering function) are considered relevant as well. Additionally, requirements ensuring ***equality in standards' application*** (such as in requirement 32: the SG should allow the plantation authority to view garden/plant information to ensure standards are followed) are considered relevant.

Experts feedback showed that decision on which requirements are related or not related to equality is not clear cut. Discrepancies in relevancy to equality evaluation could be attributed to subjectivity about equality concerns. For example, requirement 18, about keeping information private was reported as being related by 3 experts but was considered not to be relevant by 2 experts. Therefore, the decision on which requirement is contributing to equality will vary according to the application domain needs, stakeholders preferences, project budget and schedule, etc. As this survey was experimental, different views on what could be relevant depended on how each expert understood and defined the software system. Similar variations will exist in real situations, however, stakeholders views would then the deciding factor (negotiation step of elicitation method Figure 5.1).

6.3.3 Non-Relevant Requirements to Equality

Experts have identified requirements as being irrelevant to equality when they are only important to one group of stakeholders. For example a respondent rated statements (R4, 5, 6, 7, 8, 11) as irrelevant and justified it by stating: "I consider this to be a key task of the gardener, where no other stakeholder group has an interest in" and another one stated that R4 is irrelevant because "not [] all stakeholders are concerned of such service". This justification category is similar to what is suggested by the relevant category of being important if groups of stakeholders other than the main group benefit/use the feature Table 6.6. However, we argue that this is not the case. In our view, as long as we are allowing all stakeholders to accomplish their desired goals than this is equality. Else, the equality of opportunity to all is not supported.

Irrelevant categories are summarised in Table 6.7.

By associating requirements to equality, the experts' feedback has marked the possible conflicts between equality and other values such as privacy, security and freedom. R28, R29, R30 to one of the experts were rated as not relevant because they sound " against personal freedom . . . why should these authorities be allowed to do that?". Another expert rated R14 as relevant but noted that it is: "a very tricky requirement. On the one hand we have the privacy needs of the gardener, on the other hand we have stakeholders such as the nursery or also the general public which would need the data to achieve their goals". A respondent rated R18 as not relevant to equality as it could raise data security issues by stating: "all stakeholders shall not be involved for the SG [smart garden] data security". This infers the importance of balancing social sustainability values as well as other sustainability aspects.

6.3.4 Missing Requirements of Equality

Only one expert have stated that:

> "Equality among software stakeholders" seems to be very different from equality between users or the general public. For example, while the system tries to close a knowledge gap between experienced gardeners and novices, the system is not concerned with the economic success of watering supply manufacturers.

However, in this study, the aim is to achieve equality through software requirements in terms of software functionalities and features. In addition, suppliers were part of the requirements in statements 28 and 27. The "watering supply manufacturers" mentioned by the expert is similar to the one we listed. Statements 28 and 27 actually stated that market demand estimations and market feasibility reports needed by the suppliers shall be facilitated by the smart garden system. Those type of functions will help achieving economic goals of the suppliers.

6.4 Common Equality Requirements

We noted that variability factors have resulted in common requirements among different software applications (as detailed in Table 6.8) though the examples are of software from different domains. However, templates can support identification of unique goals and services as seen with the factor of religion described in Section 6.2.4.

Table 6.8: Emerging Equality Requirements

Variability	Common services
Technology used (HW, TA, CP, PIMS, AGM, SG)	Compatible software for hardware and operating systems
Disability (HW, TA, AV)	Different information display format, Braille interface
Language (HW, TA, AV, AGM, SG)	Multilingual interface
Information media (HW, TA)	Speech synthesizer
Technical literacy and knowledge (PIMS, AV)	Keyboard shortcut functions, help function and tutorials
Age (TA, HW, SG)	Modified functions to suit age differences, suitable information details for different ages
Religion (HW, TA)	Added/modified functions to suit religious differences, acceptable and not offending information
Gender (HW)	Suitable information without offending any gender

We observed that many of the equality requirements are routinely used in RE practice without explicit recognition that they relate to equality concerns. For example, many requirements derived from stakeholder variability are normally related to accessibility (e.g., service availability as text, braille, speech, sound reproduction etc.), compatibility (e.g., hardware/ operating system variability) and user interface (e.g., colour scheme customisation). Similarly, gender and religion related requirements are also often classified as non-functional requirements [210, 248] that can affect the acceptability of software [248]. Additionally, equality requirements were found within the already described requirements.

The results of the experts' evaluation suggest that the equality requirements elicitation method is useful for identifying equality requirements.

According to results in Sections 6.2.4, 6.3, we can infer that the pattern and elicitation method are reusable and applicable to several types of software applications. Additionally, they are applicable to software aimed at the general public (AV, TA, HW, SM) and to confined users (PIMS, AGM, CP). Similarly, the last column of Table 6.1 shows that pattern and the elicitation method are applicable to different software applications serving different domains.

Looking at the experts responses and justifications on relevant and irrelevant requirements to equality (Section 6.3), we can say that to them, equality is about accommodating different stakeholders in the software system. This is done by serving their needs from the system and to do so, their differences are to be respected and the software should be flexible and allow them to get what they want. Although the equality value pattern was not part of the survey and they don't know how the requirements were identified, their justifications came to the same point.

6.5 Application in Agile Software Engineering

This study is a independent study by Monica Bahl [234] built on the equality value pattern and templates. In [234], a requirement management tool was designed based on a requirements management framework that supports social sustainability. The framework was an integration between an agile requirements management process [249], and the social sustainability value pattern and templates suggested by this research.

The agile framework originally consisted of several steps [249] that were maintained and accompanied with social sustainability activities (mapped from our value pattern) [234]. For example, the first step in the agile framework is to define the project objectives. Adding social sustainability activity to this step, the framework suggested that at this stage sustainability dimentions and related values are to be identified. Here, equality is defined as a related value that needs to be supported. In the next step, the framework suggested mapping and making the equality sub-values applicable to the project. This step encourages requirements professionals to think about what are the possible equality related requirements that needs to be implemented. Another step is definition of personas (role profiles). Here, the equality with stakeholders variability value Figure 4.1 influences identification of more specific and variable category/roles of stakeholders (e.g. female buyer insted of only concidering general buyers). This leads to identifieying specific needs that could have been neglected. This was then implemented in a tool [234].

The tool allows adding sustainability dimensions, values and relate it to software requirements. As for now, the tool allows practitioners to include the social dimension through the equality and its sub-values that are included as part of the tool and can be viewed by the practitioner. The tool lists equality and its sub-values. The tool allow practitioners to assign related values to the project. Then tool asks the practitioner to identify stakeholders and relate them to the value/sub-value. It allows linking stakeholders to roles (influenced by the equality value pattern). The tool also allows relating the identified stakeholders to elicited requirements. It also allows relating requirements to indicators (those are inspired by the equality value pattern Figure 4.1 which is displayed in the tool). For example, indicators could be gender equality, income equality and equal distribution of resources). The tool also allows assigning priorities to requirements. For more details, please see [234].

The integrated agile framework was then evaluated via an online questionnaire [234].

6.5.1 Study Design

The study was conducted to evaluate the usefulness of having equality pattern in sustainability requirements elicitation [3]. 12 respondents participated in the study and they were asked first to identify equality requirements without having the equality pattern and templates[4]. Then, they were asked to use the equality pattern and templates in identifying requirements. Then they were asked to evaluate their experience with the equality patterns and templates. They were asked to report the usefulness of the pattern and templates in identifying equality objectives, stakeholders and requirements. They were also asked the effect of the pattern and templates on the requirements identification process (i.e. changes the process or improves the process or makes it more difficult or makes it easier). The study was conducted as an online questionnaire.

6.5.2 Study Results

The results in this section address the research question:
$RQ3$: Is the equality pattern usable with current RE practices?

The majority (58.3%) of the respondents were software engineers or business analysts. The rest were having other IT backgrounds such as Information, data, process, project management and others. The respondents were divided in their knowledge of software engineering into strong (25%), good (33.3%), medium (25%) and low (16.7%). Half of the respondents were with Masters degree and half of them were aged 41-60. The feedback confirmed that the equality value pattern and templates (incorporated into the framework) are a reflection of real equality requirements that are valued by IT practitioners [234]. This was derived by analysing the respondents

[3] This study is conducted by Monica Bahl as an MSc thesis

[4] The equality pattern used in this study represented the early results of our research. This did not include the elicitation method as described in Chapter 5 nor the usage guidelines.

answers to a question about equality objectives before and after using the template [234].

Additionally, the templates are found to be useful for identifying broader stakeholders lists that does not exclude less important stakeholders [234]. The study concluded despite the limited evaluation: "[w]e have shown that the meta model provided for the [e]quality core value of social sustainability can be successfully integrated with an agile process." [234] The author also added that:

> All respondents answered positively. Most of them gave the reason that the templates gave new ideas and added broader aspects than what they would have thought of themselves . . . Two of the respondents stated that the templates "made the process more complete" . . . According to one respondent, the templates "provided a framework for the answers without constraining the ideas".
>
> There were only two critical comments. One respondent alerted us to the fact, which we acknowledge, that though such templates "can be useful as inspiration, there did lie a danger that they would trigger people to repeat these instead of coming up with their own ideas". Another reminded "the values and cultures of the community cannot be easily captured due to the diverse nature, and is thus difficult to automate".
>
> That too is true, however, the purpose of the templates is not to automate the requirements process but to facilitate it in its focus on sustainability, in this case Equality, values. We believe the positive responses to this question indicate that the templates do help in serving this purpose.

6.6 Threats to Validity

Its worth noting that the described threats to validity are related to the researcher work only (excluding the study in Section 6.5). The limited results of the equality requirements elicitation method in agile methods is a limitation of our research.

6.6.1 Conclusion Validity

A possible threat to validity arises from the fact that the application of the equality requirements elicitation method on requirements documentation is conducted solely by the researcher which may cause errors. Thus, a second reviewer randomly sampled and cross-validated the application and the results ensuring that the drawn conclusions are effective and adequate. In addition, conclusion validity is increased by involving other software practitioners to apply and use the equality pattern and requirements templates (detailed in Chapter 8.)

6.6.2 External Validity

In the application of the equality requirements elicitation method to requirements documentations, the sampled requirements documentations were created by independent entities and we selected them to reduce the researcher bias (influence) on the results. Yet, threats are inevitable. As discussed in Section 6.2.1, the results of the study applies to English documentations.

In the experts survey, the respondents representation is limited to the small respondents group as they have voluntarily chosen to participate in the study. To mitigate the threat, the experts invited to take part in the survey were from different academic and industrial institutions around the world to ensure the inclusion of different opinions and views.

6.7 Summary

In this study, the equality elicitation method usage is demonstrated through seven case examples. The application reveals the usefulness of the method in the process of eliciting equality requirements. This is also confirmed by experts evaluation of the identified requirements of one example and their relevance to equality. Although results are limited, the use of the value pattern in agile development was discussed showing that the value pattern is usable alongside other RE practices.

Table 6.5: Equality relevant requirements categorisation (1)

Categorisation	References
Related to equality	
Participation and Empowerment *Accessibility* *Inclusion* *Usability*	74 38 34 4
Examples: “Gardeners may have different interests and capacities to manage the garden, which need to be accommodated. Plus functionality should be available to gardeners with different digital literacies and physical abilities” “Auto-watering allows gardens to be watered, independent of the physical abilities of the gardeners” “If automated, functionality allows stakeholders with different physical abilities to check the soil moisture, without obligatorily and directly interacting with the garden” “Functionality should help gardeners to meet their goals, whatever they are” “Gardeners should have access to the application in a language they feel comfortable with” “Elder gardeners, who might have less digital literacy, should be able to understand and use the application as anyone else” “not all people can get mobile devices and not all of the are comfortable to work on PC” and “Usability and inclusion of elderlies”.	
Knowledge management *Equal access to information* *Improve knowledge* *Knowledge sharing* *Reduce knowledge gap* *Update knowledge*	49 3 8 29 5 1
Examples: “all stakeholders shall be aware of such information” “[a]ll gardeners should have the option to benefit from the gardening knowledge” “potential gardeners should have access to the gardening knowledge in order to become a gardener if desired, independently of their previous knowledge, experience, or physical ability” “it might be allowed for the knowledge sharing” “because of different level of knowledge” “These usage features are what makes users closer to educated gardeners” “User might not have expertise in that”.	
Different stakeholders goals and needs	12
Examples: “It could be relevant in the sense that estimation of demand should should attend the need of different stakeholder” “Does “plant nursery” include all the people which have an interest in the seasonal plantation history? If so, we are fine. However, if there are stakeholders which are not considered this might lead to inequality” “it allows supplier to have future prediction for their business benefits” “Gardeners may have different interests and capacities to manage the garden, which need to be accommodated. Plus functionality should be available to gardeners with different digital literacies and physical abilities” “Functionality should help gardeners to meet their goals, whatever they are.” “Gardeners should be able to access the application in the medium they feel more comfortable with.”	
Privacy	13
Examples: “Critical because of potential privacy needs” “Protection is needed for equality”.	

Table 6.6: Equality relevant requirements categorisation (2)

Categorisation	References
Related to equality	
Relevant to several stakeholders	6
Examples: “What about the nursery, this feature could be relevant from them as well?” “ So I wonder why only gardeners should have that option, what about the plant nursery people?” “Highly critical that only elder[l]y garderns should have a help fu[n]ction. Why not young gardeners.”	
Regional equality	4
Examples: “because it considers regional equality to different parties” “it could be relevant in the sense that all areas should be equally considered for public plans, and recommendations should be applicable independently of the gardeners’ social status or other influential characteristics.”	
Public benefits	3
Examples: “it allows them to have future prediction for the public benefit of following standards” “it allows them to have measurements for water consumptions for the public benefit”.	
Accommodate external changes	2
Examples: “The type of weather is not same in different countries and even within the same countries it vary between area so the system should allow them to change the parameters accordingly” “because of changing in environment.”	
Equality in following standards	1

Table 6.7: Equality irrelevant requirements categorisation

Categorisation	References
Not-Related to equality	
No reason provided	8
Only needed by one group of stakeholders	7
No clear link to equality	6
Privacy and personal freedom	5
Not related to equal treatment	2
Security	1
Usability	1

Chapter 7

Software Users Equality Survey

Previous chapters have demonstrated social sustainability value pattern and requirements identification templates. Chapter 6 illustrated the application of the elicitation method and value pattern in deriving equality requirements. This chapter studies *how equality and its related requirements (from the previous chapter) are perceived and valued by software users.*[1]

In order to understand software users' attitudes towards the equality requirements suggested by this study, a web-based survey was used to allow us to reach different groups of software users across the world in a relatively short period of time [251]. The guidelines in Van Selm and Jankowski [251] and Kitchenham and Pfleeger [243] were followed in this study.

7.1 Objective

The objectives of this study are:

[1]This study is accepted at the 6th International Conference on ICT for Sustainability in 2019 [250]

(i) To define a ranking of requirements important to equality, as perceived by the general software user community;

(ii) To observe possible agreement among the user community on equality in software

(iii) To investigate the effects of demographic factors on perceptions of equality.

7.2 Study Method

A cross-sectional [243] survey was conducted. As we were interested in general software users' perspectives with varying demographic factors, we chose to utilise an online survey [251] format which was distributed widely through different communities and lists. The data were collected via a web survey tool named BOS[2].

The respondents were presented with a set of 21 requirements statements (Q1) and where asked to evaluate each statement's importance in supporting equality. The statements are general statements that are applicable to different software systems. This was done in order to avoid user familiarity problems with a specific piece of software. The statements are based on the requirements identified as a result of social sustainability value pattern and templates application, as explained in Chapter 6, that present similar solutions for different software. Table 7.1 shows the list of the statements. Each participant was asked to rate the importance of a requirements statement for supporting equality.
Participants evaluated each statement using a Likert scale ranging from 'Not at all important to software equality', to 'Neither important nor unimportant', 'Slightly important to software equality', 'Important to software equality', 'Very important to software equality'. Because this is an importance scale, we did not aim to produce

[2] https://www.onlinesurveys.ac.uk

balanced levels. Knowing that a statement is not important to equality is more relevant than knowing the degree or depth of unimportance. In the case of unimportance, the direction is what we are looking for and not the depth. On the other hand, knowing the degree of importance can help requirements engineers to prioritise equality requirements and decide which should be included in the first release and which to be kept for later releases taking into consideration resource availability (e.g. time, money, skills, etc.). In this case, "discrimination . . . between the positive scale positions" is important [252].

Statements 1, 4, 5, 17 do not relate to equality; they were used as red herrings to identify how well the respondents distinguish the notions of equality from other requirements. These statements relate to security (S1), performance (S4), robustness (5) and availability (S17).

Statement 9 (ability to accommodate new types of users is important for supporting equality) was initially introduced as a non-equality (scalability) requirement but, after re-examination it was agreed that this statement supports equality by allowing new types of users to benefit from the software. This is also a concern in sustainability (i.e. current and future generations sustainability).

Its worth noting that we are not claiming that the listed statements are the complete equality requirements but as what we have observed from application of the equality value pattern in Chapter 6, those are the common ones and they are applicable to different software. We can have more specific equality requirements depending on the application domain and the case under investigation.

The respondents were also presented with two questions (Q2-3) to understand what concerns drive the notion of equality for software systems among the respondents: from profit, to usability, functionality, and user priorities. While question 2 asked them to make a choice for the key equality-conducive goals, question 3 asked which user groups should be supported. Goals and users here arise from the different definitions of equality (as discussed in Chapter 3, Section 3.4) as well as the application of the equality value pattern in Chapter 6. In future, questions 2 and 3 can be updated to reflect the goals and roles deriving from the proposed models in Figures 5.2 and 5.3 (Chapter 5).

Table 7.1: Requirements Statements

	Requirements Statement
S1:	User authentication is important for supporting equality
S2:	Usability of software to users from different age ranges is important to support equality
S3:	Suitability of software to users from different age ranges is important to support equality
S4:	Short response time to user enquiry is important for supporting equality
S5:	Short recovery time after system failure is important for supporting equality
S6:	Suitability of software to users from different genders is important to support equality
S7:	Considering direct stakeholders' goals behind using a software is important to support equality
S8:	Fairly selecting which goals will be implemented in the software is important for supporting equality
S9:	Ability to accommodate new types of users is important for supporting equality
S10:	Multilingual interface is important for supporting equality
S11:	Different information presentation formats (e.g., audio, video, text) is important for supporting equality
S12:	Compatibility of software application with different operating systems is important for supporting equality
S13:	Compatibility of software application with different hardware devices is important for supporting equality.
S14:	Availability of softwares usage guidance (e.g., help, tutorials, and tips) considering users with no/little prior knowledge of this software is important for supporting equality
S15:	Availability of software's shortcuts to accomplish tasks for experts and fast learners is important for supporting equality
S16:	Availability of software application on different web and mobile platforms is important for supporting equality
S17:	Availability for use 24 hours per day, 365 days per year is important for supporting equality
S18:	Allowing stakeholders to equally access software services to achieve their goals is important for supporting equality
S19:	Suitability of software for users from different religious beliefs is important for supporting equality
S20:	Accepting information from different media (e.g., voice, text, braille) is important for supporting equality
S21:	Considering indirect stakeholder goals that are affected by the software is important for supporting equality

7.2.1 Survey Design

Due to the fact that social sustainability (i.e. equality) requirements were derived from the value pattern and templates suggested by this research study, it was necessary to construct a new survey and not use an excisting one in the requirements engineering domain [243].

The respondents were allowed to save their partial response and return to complete the survey at a later date through the 'Finish later' link.

7.2.1.1 Population and Sampling

The availability of the questionnaire was advertised through invitation messages posted in LinkedIn, Instagram, Facebook, Twitter, ResearchGate and WhatsApp to ensure that a wide range of respondents were able to access the web questionnaire. This is known as ***unrestricted sampling*** [251]. Additionally, invitation emails were sent to the PhD students and staff list in the Informatics Department at the University of Leicester and through academic colleagues in other countries such as (the USA, Brazil and Germany). Invitations were also sent to academic colleagues in the Omani colleges and universities via randomly selected staff members' emails available online, asking them to distribute the participation request locally. This was done to encourage various background representation. In both approaches, ***convenience sampling*** [243] was used. Moreover, ***snowball sampling*** [243] was also utilised by asking respondents in the invitation letters to forward the survey onto whoever they thought may be interested in participating in the survey.

7.2.1.2 Questionnaire Form

The survey comprised four pages as explained below.

A **An information sheet and consent page.** This page introduces the participants to the research, its objectives and completion time. Additionally, the page asks the participants to confirm their age as well as requesting their consent to voluntarily participate by clicking on the 'Agree' option. Failing to do so results in the participant being redirected to a screening out message. This feature was used to ensure that all participants are aged 18 years or above and also to obtain the participants' consent electronically.

B **An equality and software page.** This page encompasses the main equality questions. It starts with equality and stakeholder definitions. This is followed by:

(a) Question 1: In this question, participants are requested to evaluate 21 requirements statements as described above in Section 7.2. Participants assess each statement and decide its level of importance in supporting equality among software stakeholders.

(b) Question 2: This question presents several goals to the participants and requires them to evaluate and select goals that support equality.

(c) Question 3: Presents a small scenario involving online shopping software with different types of stakeholders; each with different goals. The participants were asked to decide on the stakeholder goals they would select to support equality.

C **Background information.** In this section, participants provide their gender, age, religion, education and employment details. They were also asked to provide their level of proficiency in using software. This page was partially adapted from a survey by Osho et al. [253].

D The **end page** is an automatically generated page created by BOS. The page is a simple thank you note.

7.2.1.3 Pilot Test

To pilot the study, PhD students and two academics were invited to complete the survey and provide input for its improvement. They were requested to provide their feedback and comments. Participants were asked to provide notes regarding the layout, the wordings, the clarity and understandability of questions and instructions as well as the length and relevance of the questions and instructions.

The feedback flagged a number of concerns that lead to changes to the wordings of the information sheet, corrections of the grammatical mistakes and the demographic (background) information page was moved from the beginning to the end of the survey.

The survey was opened on 26th December 2016 and closed on 23rd February 2017. There were 164 responses, of which 155 were complete and valid.

7.2.2 Data Analysis

The data coding used in this study is explained in Appendix D. The data were quantitatively analysed using frequency analysis and inferential analysis.

A Frequency analysis is used to summarise and describe the sample using tables and charts [254] such as Table 7.2 and Figures 7.1 and 7.2. This was also used to analyse the requirements statements frequency (see Table 7.3 for an example).

B Inferential analysis is used to test the hypothesis of the study [254]. Given that all variables for statements ranking in Q1, respondents' background (questions 4 to 9) and equality goals (questions Q2-3) are categorical (i.e., nominal and ordinal [255]), a chi-squared test would normally be performed [254] to examine the relationships. However, in some cases, when a crosstabulation of variables was created, the data contained a high percentage (more than 20%) of cells with fewer than 5 counts. Consequently, the Fisher's exact test [255] is preferred to the chi-square. Additionally, because the compared variables are with more than 2 categorical levels (e.g., employment status has 4 options: employed, student, unemployed and retired), the Fisher-Freeman-Halton Test[3] was performed. Furthermore, the Monte Carlo Exact test was utilised (where needed) as an alternative to Fisher's test to resolve the issues of hight-memory intensity posed by the exact tests [256].

7.3 Results

The following sections present the the results of the survey. First, the respondents are profiled based on their demographic information. Then the above stated objectives in Section 7.1 are addressed.

7.3.1 Respondents' Profile

Table 7.2 summarises the respondents profile. It list respondents background factors, levels of the factor, number of responses as well as valid response percentage for each factor.

The respondent's sample was nearly even in terms of gender with slightly more responses from female participants (by 1.2%), as illustrated in Table 7.2. The the largest group of respondents were aged 35 to 44 years (36.8%), followed by participants aged 25 to 34 years (25.8%) with only one aged over 65 (see Table 7.2).

[3]http://www-01.ibm.com/support/docview.wss?uid=swg21479647

Most respondents were Muslim (Figure 7.1). This could be a result of the more direct invitations to Omani participants as discussed in Section 7.2.1.1. Christians and Hindus were the next two largest religious groups (see Figure 7.1).

The respondents' education level is depicted in Table 7.2 and in Figure 7.2. There were no unschooled respondents and the majority had a postgraduate education (70.3%) with 34.2% holding a PhD and 36.1% holding a Masters degree. In addition, only 1.9% of the respondents were novice software users and the majority (45.8%) described themselves as having advance proficiency (see Figure 7.3). This shows that the sample is leaning towards highly educated software users. This in itself is not entirely surprising because the topic of the survey (software and equality) as well as the method of data collection already presumes some minimum education and technological literacy levels.

Furthermore, the majority of the participants (115 respondents, 74.2%) were employed, while students comprised 23.2% of the respondents. There were no retired respondents (see Table 7.2).

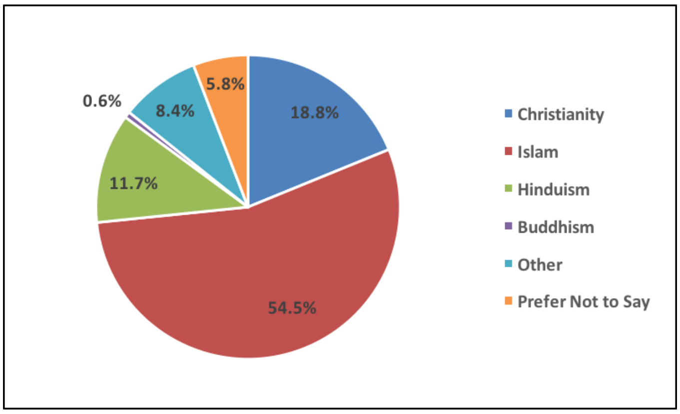


FIGURE 7.1: Respondents Religion, N = 154

Table 7.2: Respondents Profile

Background		No (Valid %)
Gender	Male	76 (49.4)
	Female	78 (50.6)
Age	18 to 24 years	21 (13.5)
	25 to 34 years	40 (25.8)
	35 to 44 years	57 (36.8)
	45 to 54 years	29 (18.7)
	55 to 64 years	7 (4.5)
	Age 65 or older	1 (0.6)
Religion	Christianity	29 (18.8)
	Islam	84 (54.5)
	Hinduism	18 (11.7)
	Buddhism	1 (0.6)
	Other	13 (8.4)
	Prefer Not to Say	9 (5.8)
Highest level of education	PhD (or equivalent)	53 (34.2)
	Masters Degree (or equivalent)	56 (36.1)
	Undergraduate (or equivalent)	24 (15.5)
	A college degree (diploma and equivalent)	21 (13.5)
	High school degree or less	1 (0.6)
Employment	Employed	115 (74.2)
	Student	36 (23.2)
	Unemployed	4 (2.6)
	Retired	0 (0)
Software use proficiency	Novice	3 (1.9)
	Intermediate	28 (18.1)
	Advanced	71 (45.8)
	Expert	53 (34.2)

7.3.2 Frequency of Equality Statements

The frequency of statements ranking is summarised in Table 7.3. This table lists the frequencies of responses on the 21 statements. Responses range from 'Not at all important to software equality' 'NAI', to 'Neither important nor unimportant' 'NINU', 'Slightly important to software equality' 'SI', 'Important to software equality' 'IE', 'Very important to software equality' 'VIE'. To depict the ranking of the

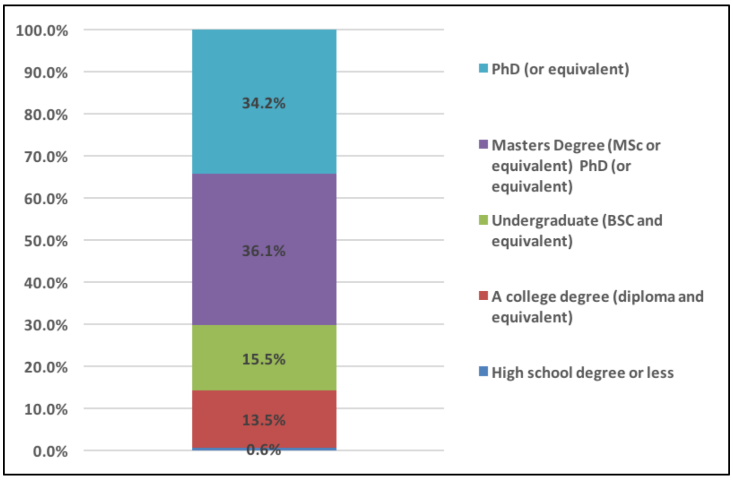


FIGURE 7.2: Respondents Highest Education, N = 155

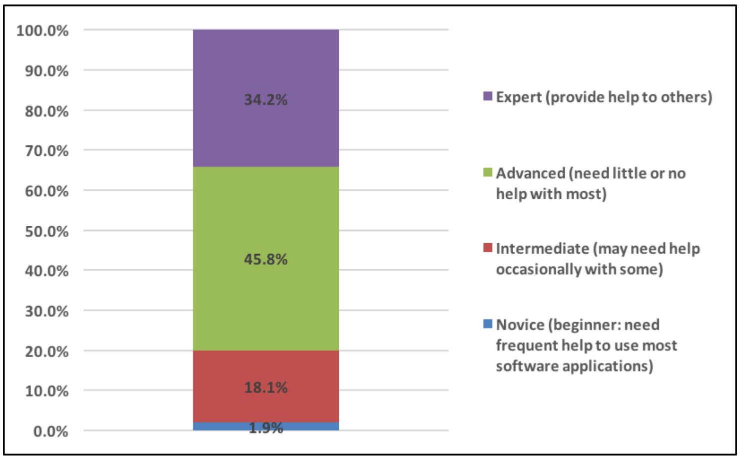


FIGURE 7.3: Respondents Software Usage Proficiency, N = 155

requirements statements based on the frequencies, we utilised diverging stacked bar [245] (see Figure 7.4).

As the figure shows, S10: multilingual interface is the most highly rated statement in terms of its importance. This is followed by S14: software's usage guidance. The third most highly rated statement is S11: different information presentation formats (e.g., audio, video, text) followed by S20: different input support formats and S3: support for users across various ages. These statements also have low irrelevance and indecision. This indicates that there is an agreement (albeit not unanimous) among the respondents that these statements are closely related to equality.

Table 7.3: Statements Frequencies

Statement	NAI	NINU	SI	IE	VIE
S1* (sec)	18 (11.6%)	26 (16.8%)	18 (11.6%)	45 (29.0%)	48 (31.0%)
S2	8 (5.2%)	10 (6.5%)	22 (14.2%)	58 (37.4%)	57 (36.8%)
S3-1	4 (2.6%)	11 (7.1%)	25 (16.2%)	53 (34.4%)	61 (39.6%)
S4* (per.)	6 (10.3%)	29 (18.7%)	32 (20.6%)	41 (26.5%)	37 (23.9%)
S5-1* (rob.)	20 (13.0%)	21 (13.6%)	26 (16.9%)	39 (25.3%)	48 (31.2%)
S6-1	9 (5.8%)	11 (7.1%)	20 (13.0%)	53 (34.4%)	61 (39.6%)
S7	11 (7.1%)	13 (8.4%)	31 (20.0%)	55 (35.5%)	45 (29.0%)
S8-2	6 (3.9%)	14 (9.2%)	22 (14.4%)	64 (41.8%)	47 (30.7%)
S9	2 (1.3%)	15 (9.7%)	18 (11.6%)	61 (39.4%)	59 (38.1%)
S10-2	3 (2.0%)	5 (3.3%)	19 (12.4%)	55 (35.9%)	71 (46.4%)
S11	4 (2.6%)	7 (4.5%)	19 (12.3%)	53 (34.2%)	72 (46.5%)
S12	6 (3.9%)	12 (7.7%)	25 (16.1%)	53 (34.2%)	59 (38.1%)
S13	6 (3.9%)	13 (8.4%)	23 (14.8%)	56 (36.1%)	57 (36.8%)
S14	6 (3.9%)	5 (3.2%)	18 (11.6%)	53 (34.2%)	73 (47.1%)
S15	9 (5.8%)	16 (10.3%)	33 (21.3%)	60 (38.7%)	37 (23.9%)
S16	6 (3.9%)	10 (6.5%)	26 (16.8%)	53 (34.2%)	60 (38.7%)
S17* (av.)	12 (7.7%)	18 (11.6%)	32 (20.6%)	47 (30.3%)	46 (29.7%)
S18	4 (2.6%)	12 (7.7%)	27 (17.4%)	51 (32.9%)	61 (39.4%)
S19-3	5 (3.3%)	13 (8.6%)	18 (11.8%)	50 (32.9%)	66 (43.4%)
S20	5 (3.2%)	10 (6.5%)	26 (16.8%)	57 (36.8%)	57 (36.8%)
S21	5 (3.2%)	20 (12.9%)	39 (25.2%)	65 (41.9%)	26 (16.8%)

Note: * marks statements that are not directly related to equality; S1 relates to security (sec), S4 relates to performance (per.), S5 relates to robustness (rob.), and S17 on availability (av.)

Of the equality statements, the lowest-ranked was S21 (considering indirect stakeholder goals that are affected by the software) and S15 (availability of software's shortcuts). This could be explained by the fact that the key focus of software is normally placed on direct software users, with indirect stakeholders considered thereafter. Clearly, the long-term cumulative effects of a software system could dramatically affect indirect stakeholders. For example, the long-term use of Amazon.com by a large number of individuals has gradually undermined many physical book shops and their customers. Normally, software users first of all focus on the direct effect of their immediate interaction with the software system (e.g. the ability to obtain the desired book at a lower price). This tension between the priorities of

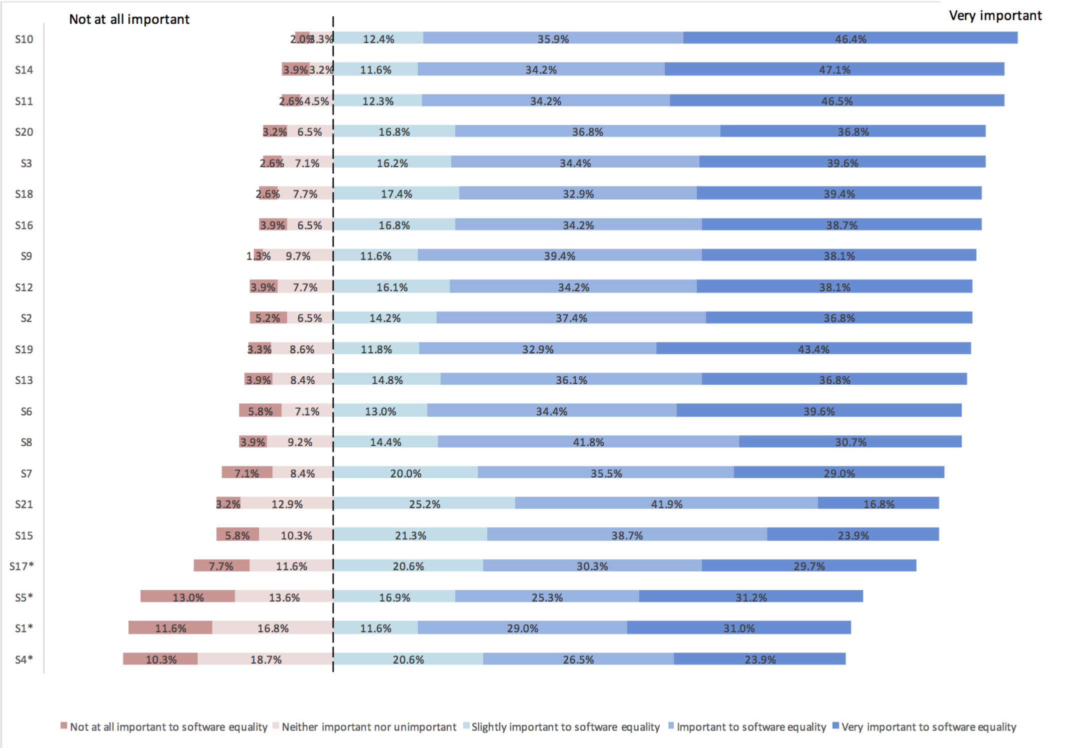


FIGURE 7.4: Statements Ranking

direct and indirect stakeholders is, indeed, one of the characteristics of social sustainability concerns. S15 might have been rated least important because of the extra service supporting experienced users is more thought of as a usability requirement [257]. It might be that the extra support is viewed as a privilege only given to expert users. However, having extra features to be used by experts does not hinder novice users and they are still allowed to use the software as they wish (see Table 7.3 for further details).

Considering the topic ranking among the equality-related statements (except S15), we observed that statements supporting user interaction with the software and considering users variability are ranked highest. The statements that supports stakeholders' goals are at the bottom of the list (S7 and S21).

Although the respondents also rated the non-equality statements as being important to equality, the four statements are ranked as the least relevant to equality and the most doubted (i.e. neither important nor unimportant). The lowest ranked statements (S5, S1, S4) are those related to robustness, security and performance. S17 concerns software availability and it might be thought of as supporting equality providing access at any time for the users' convenience without restrictions.

Despite these discrepancies, the overall ranking of the statements indicates that there is a general agreement among the survey respondents regarding the statements that relate to equality as well as those that do not relate (or only weakly relate) to equality.

7.3.3 Frequency of Equality Goals

Figures 7.5 and 7.6 display the results to questions 2 and 3. The results indicate that the respondents are fairly equally distributed in prioritising support for different groups goals and not only one group; provision of same but equal functionality to all users (without distinction); and simple usability of software as the key drivers for enabling equality through software.

For Q2, the largest group of respondents (37.7%) said that in order to support equality, the most prioritised goals for each group should be integrated into software. In Q3, 72.4% of this respondents group consistently noted that the goals of different stakeholder groups should be implemented to support equality. This suggests they understand that equality is achieved by looking at the different stakeholder groups and finding the best way to allow them to achieve their goals through the system. This will also include identifying variability factors among each group and finding the best way to accommodate those differences. Thus, this group of respondents considers *equality to be the equitable support of various goals of the different user groups* for a given software system.

Another large group of respondents (31.8%) for Q2 said that to them the best way to enable equality through software is by providing the same functionality at same level to all user groups. For Q3, the majority (69.4%) of this respondent group also stated that the goals of different stakeholder groups should be supported.Thus, this group considers *equality to be the delivery of the same service at the same level to all user groups* for a given software system.

The third largest group of respondents (27.9%) for Q2 stated that the provision of the best usable interface is the key driver for equality. In response to Q3:

- 50.0% of respondents in this sub-group indicated that the goals of all user groups should be implemented. Thus, this sub-group think of equality as the usability of the software interface (i.e. effective, efficient, and satisfying design [258]).
- 7.1% of respondents from this sub-group indicated the specialist (disadvantaged) user groups as the ones who should be especially supported via the software system to enable equality. Thus, this sub-group regards equality in terms of accessibility (focus on disability) because "many accessibility requirements also improve usability for everyone" [258].

Furthermore, a small group of respondents (2.6%) considered profit maximisation to be a key direction to enable equality in Q2, and for Q3, half of this sub-group chose prioritising requirements for the "gold user" group (i.e., those who buy expensive packages of service for a given software) as the key in achieving equality, while the other half preferred to support all user groups.

In summary, while (as found from the responses to Q3 responses) the vast majority of respondents (64.7%) think that the goals of all user groups should be supported, perceptions of how equality should be delivered with the software system are varied;

just over one third of survey respondents noted the need to support the goals that each user groups prioritises; another third focused on equal service levels to all - whichever service that may be and to whomever it is delivered; and the other (slightly less than a third) portion of respondents underlined the usability of software as being key to its support for equality.

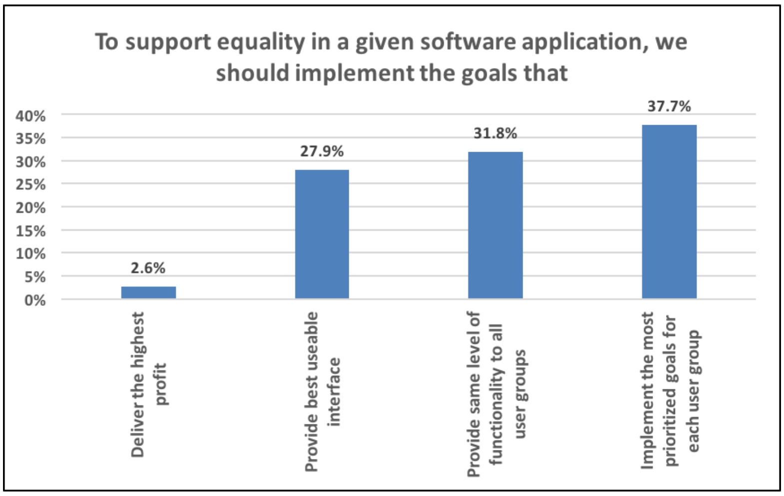


FIGURE 7.5: Question 2 frequencies, N = 154

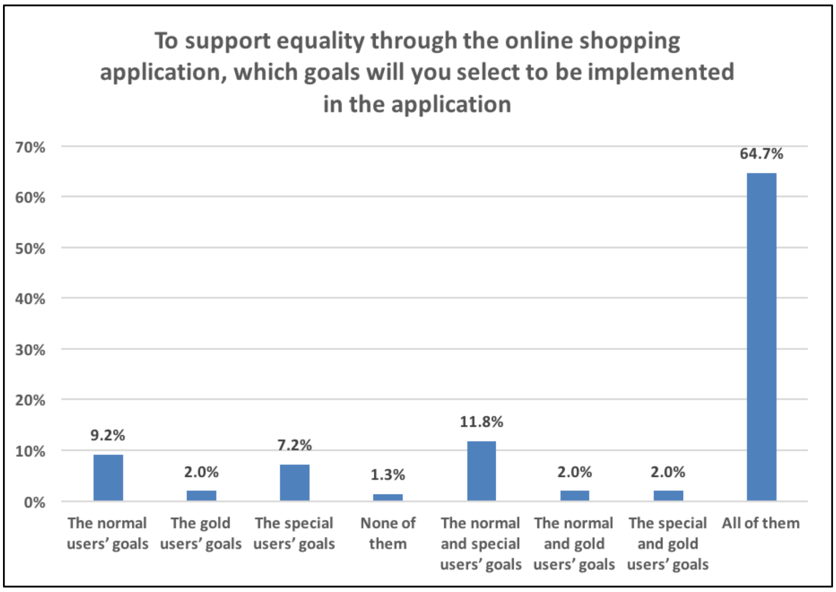


FIGURE 7.6: Question 3 frequencies, N = 153

7.3.4 Impacts of Background Factors

To study what impact background factors may have on perceptions of equality, we consider if and how the demographic factors affect both the rating of equality-related statements and the equality goal priorities (i.e., responses to Q2 and Q3). In this part, inferential statistics as described in Section 7.2.2 were utilised.

We start our study of the demographic factors' impact by formulating a null hypothesis is that the background variables (Q4-9) and equality statements and goals (Q1- Q3) variables are independent, with the alternative hypothesis that they are dependant:

- H_0 : Background variables and equality statements rating and goals variables are **not related**.
- H_a : Background variables and equality statements rating and goals variables are **related**.

By studying the results of the tests (chi-squared or exact), a decision about rejecting the null hypothesis is required. To reject the null hypothesis, the test's p-value should be less than 0.05.

Tables 7.4 and 7.6 depict the results of the inferential analysis. The tables illustrate the results of the factors affecting statements rating and goal priorities listing exact tests scores and p-values.

7.3.4.1 Impact on Statements Rating

Analysis for the role of the background factors on the rating of the statements not-related to equality (i.e. S1, 4, 5, 17) shows that the rating of statements 1 and 17 is not affected by the background factors of the respondents (see Table 7.4). However, we observe a relationship between the religion of the respondents and their rating of the statements for S4 (response time) and S5 (recovery time); p-value = 0.047 for S4 which is under 0.05, and p-value = 0.008, for S5 which is under 0.01).

Many Muslim participants (31.0%) rated statement 4 as being **very important** to equality. Many Christian (24.1%) and Hindu (44.4%) participants also found it to be **important to equality**, and a large number of those with other beliefs (38.5%) rated this statement as being **slightly important** to software equality. Finally, many of the respondents who did not disclose their religious beliefs (44.4%) rated the statement as being **neither important nor unimportant**.

For statement 5, many Christian (34.5%) and Muslim (36.1%) participants found it to be **very important** and many Hindu participants (38.9%) also rated it as being **important to equality**. Participants with other religious backgrounds scored a tie between **neither important nor unimportant** and **slightly important** to software equality with 23.1%. While 33.3% of those who preferred not to disclose their religion rated statement 5 as being **not important**.

We observe that both of these statements are related to time, and previous research has demonstrated that there is a correlation between time valuation and cultural factors [259, 260]. Thus, it is likely that this relationship is a demonstration of such a cultural, time-related influence.

In the equality-related statements, we noticed that gender affects the rating of S14: availability of software's usage guidance (e.g., help, tutorials, and tips) considering users with no/little prior knowledge of this software. While the majority of female respondents (57.7%) rated this statements as being **very important**, 35.5% of male respondents rated the statement as being **very important** and (39.5%) as **important**. Related studies, such as [261], have demonstrated that males generally tend to seek less help than females, which could also explain differences in the perceived importance of help-related statements.

Table 7.4: Statements and Influencing Background Factors

Requirements Statement	Background Factor
S1 (sec)	None
S2	Religion: FET= 31.141, p=.020*
	Software usage proficiency FET=31.869, p=.001*
S3	Age FET=28.867, p=.049*
	Education FET=24.608, p=.048*
	Software usage proficiency FET=24.058, p=0.007
S4 (per.)	Religion FET=28.367, p=.047*
S5 (rob.)	Religion FET=33.670, p=.008*
S6	Age FET=30.254, p=.028*
	Employment FET=17.828, p=0.007
	Software usage proficiency FET=19.916, p=0.034
S7	None
S8	Religion FET=28.347, p=.050*
S9	Age FET=30.270, p=.036*
	Education FET=28.462, p=014*
S10	None
S11	None
S12	Religion FET=32.520, p=.012*
S13	None
S14	Gender FET=11.211, p=0.018
S15	None
S16	None
S17 (av.)	None
S18	Age FET=34.159, p=.008*
	Religion FET=38.008, p=.001*
	Education FET=24.866, p=.048*
	Employment FET=15.242, p= 0.026
	Software usage proficiency FET=21.800, p= 0.017
S19	Religion FET=28.918, p=.042*
S20	Religion FET=32.538, p=.013*
S21	Religion FET=33.154, p= .010*

Note: FET = Fisher's exact test

* Monte Carlo estimates using 10000 sampled tables was used

The results presented in Table 7.4 indicate that:

- Age affects the rating of S3, S6, S9 and S18 which is motivated by the fact that people from different age groups have different preferences (as shown in [262]).
- Religion affects the rating of S2, S8, S12, S18, S19, S20 and S21.
- Education affects the rating of S3, S9 and S18
- Employment status affects the rating of S6 (software suitable for different genders) and S18 (equal access to service to achieve goals).
- Software usage proficiency affects the rating of S2, S3 (software suitable for different ages), S6 (software suitable for different genders) and S18 (equal access to service to achieve goals).

Thus, we observe that there is an overall agreement across the world and user communities regarding the relevance of specific statements for equality, which makes it possible to develop and use generic equality requirements and templates. Despite this, each community/group has also its specific preferences, which is to say that the templates/requirements must be adapted for each user community.

7.3.4.2 Impacts on Equality Goals

To examine the relationships between the respondents' backgrounds (Q4-9) and the equality goals (questions Q2-3), chi-square statistics were meant to be performed [254]. However, due to the high percentage of cells with fewer than five counts, Fisher's exact test [263] was performed. Additionally, because the variables compared have more than two categorical levels, in SPSS the equivalent test is called the Fisher-Freeman-Halton Test[4]. The hypothesis is:

- H_0 : Background variables and equality goals variables are independent (unrelated)
- H_a : Background variables and equality goals variables are dependent

The only indicated relationship was between software usage proficiency and the goals in question 3, as depicted in Table 7.6. In this case, the null hypothesis is rejected and the respondents' selection of goals to be implemented in the software is related to the respondents' proficiency using the software. Table 7.5 shows that half of the novice respondents selected 'all of them' goals and the other half selected the gold user goals. The majority of intermediate, advanced and expert users selected the 'all of them' goals with 37.0%, 70.4% and 71.7% respectively.

7.4 Discussion

In this chapter we presented a survey-based study regarding the perceptions that software users have of equality. We investigated if the wider community of software users has generally agreed upon equality goals, and the equality requirements ranking that should be supported via software. We also looked at whether users' background characteristics affect their perceptions.

The respondents to our survey demonstrated a clear and nearly equal split in their perception of equality goals: roughly a third of them considers equality in terms of equal distribution (same functionality to all). Another third perceive it in terms of

[4] `http://www-01.ibm.com/support/docview.wss?uid=swg21479647`

Table 7.5: Q3 and Respondents Software Usage Proficiency

Q3 options	Novice (n= 2)	Intermediate (n=27)	Advanced (n=71)	Expert (n=53)
The normal users' goals	0 (0.0%)	5 (18.5%)	6 (8.5%)	3 (5.7%)
The gold users' goals	1 (50.0%)	0 (0.0%)	1 (1.4%)	1 (1.9%)
The special users' goals	0 (0.0%)	3 (11.1%)	4 (5.6%)	4 (7.5%)
None of them	0 (0.0%)	1 (3.7%)	1 (1.4%)	0 (0.0%)
The normal and special users' goals	0 (0.0%)	5 (18.5%)	6 (8.5%)	7 (13.2%)
The normal and gold users' goals	0 (0.0%)	2 (7.4%)	1 (1.4%)	0 (0.0%)
The special and gold users' goals	0 (0.0%)	1 (3.7%)	2 (2.8%)	0 (0.0%)
All of them	1 (50.0%)	10 (37.0%)	50 (70.4%)	38 (71.7%)

unequal equality, whereby each group needs to be treated differently by supporting their own priorities and needs to achieve their own goals. Finally, the last third perceives software equality in terms of a narrower scope of accessibility requirements. While all of these topics form part of the present equality discourse, this nearly equal split of priorities was somewhat unexpected. Furthermore, the majority of respondents also indicated that in order to be conducive for equality, a software system should support a wide variety of its user groups.

For software engineering professionals this means that in order to engineer a software system that is perceived to be conducive to the equality characteristic of social sustainability, that system must:

- Provide usability and accessibility support to all its user groups.
- Support unequal equality, i.e., to ensure that more support is provided to those with greater need; more rewards are provided to those who contribute more

Table 7.6: Equality Goals and Background Factors

Background factor	Equality Goal	Results	Interpretation
Gender	Q2	FET=1.577, p=0.677	No relation
	Q3	FET=5.451, p=0.651	No relation
Age	Q2	FET=14.992, p=0.479	No relation
	Q3	FET=41.617*, p=0.280	No relation
Religion	Q2	FET=15.314, p=0.435	No relation
	Q3	FET=29.912, p=0.951	No relation
Education	Q2	FET=10.566, p=0.655	No relation
	Q3	FET=37.014, p=0.157	No relation
Employment	Q2	FET=4.290, p=0.666	No relation
	Q3	FET=14.511, p=0.431	No relation
Software proficiency	Q2	FET=10.474, p=0.286	No relation
	Q3	**FET=31.563, p=0.048**	**There is relation**

Note: FET = Fisher's exact test
* Monte Carlo estimates using 10000 sampled tables was used

(i.e., groups are positively differentiated with respect to the goals they want to achieve and support that they need),

- However, equal equality is also observed, whereby despite the differentiated stakeholder goals, all access to resources and services is perceived to be equal and fair.

This, clearly is not an easy task to accomplish but if either are is not upheld, at least one third of the potential user community is likely to be disappointed.

Furthermore, we observe that although there is no unanimous agreement on which requirement statements are most important for equality, there is a general convergence of views suggesting that user diversity support requirements are paramount, followed by differentiated goal support requirements.

There is also generic convergence amongst the software users around the notions that are not relevant to equality, though demographic factors (i.e. religion) affect these perceptions significantly.

7.5 Threats to Validity

7.5.1 Internal Validity

The study design has a large role to play in ensuring that the results correctly convey the information contained in the study data. In this respect, the internal validity of this study could be threatened if the statements in Q1 and goals in Q2 are poorly related to the equality concern. Although this threat cannot be fully eliminated, we have mitigated it by ensuring that the equality statements and goals are representative and are closely related to concerns expressed in requirements specifications from several independently defined software system requirements documents.

Although we discussed in Section 7.2 the reasons of having an unbalanced scale, we acknowledge that this could raise a threat to validity as it might have miss-leaded responses.

We have also used an English language survey that has been filled in by participants from other countries who are unlikely to be native English speakers. Thus, it is possible that some respondents may have interpreted some of the statements differently to how they were intended. Be that as it may, English is the most widely used language in academic research and publications, and the participants would have had access to translation support (e.g., via paper and online dictionaries, and translators). We believe this was a reasonable choice to make. In addition, we carried out a pilot for the data collection process to improve both the clarity of the questions and statements as well as the structure.

Another possible threat is related to the respondents' maturation [264] because the time factor can adversely affect responses if respondents are tired or bored. To mitigate this threat, the 'Finish later' link was used to allow respondents to continue answering the survey at their leisure without having to finish the survey in one sitting.

7.5.2 External Validity

A potential threat to external validity is related to the respondents being representative of the population [264], which would have been influenced by such factors as the sampling methods used [264] (see Section 7.2.1.1). To ensure that the respondents were representative of different countries, religions, and ages, we posted the request internationally and across various age-groups.

However, because we used distribution methods (e.g., LinkedIn, Research Gate, etc.) accessible to us, it is likely that some populations with very different views were not reached. Indeed, we have reported that the *respondents are rather over-educated compared to the expected average set of software users*, as the request for participation was widely posted using university lists and personal requests to university academics.

Moreover, due to the distribution and data collection format used, it was not possible to calculate the response rate. Our respondents are those who volunteered to participant due to some personal interest; clearly not all who received the participation request have completed the survey.

Thus, we must note that the generality of the results presented in this chapter relate to a sub-section of well educated, English speaking, and technologically literate software users.

7.5.3 Conclusion Validity

A possible threat is related to the reliability of the measures that are affected by poor question wording or a suboptimal survey layout [264]. This threat was mitigated by the pilot test as discussed in Section 7.2.1.3.

7.6 Summary

This chapter has presented an empirical study conducted for the current research. Software users participated in the study with 155 valid responses being received. This study aimed to perceive users' perspectives of equality in relation to software applications.

The results indicated that there is no general agreement on which requirements are most important to equality. There is a general agreement that diversity supporting requirements are more important to equality followed by differentiated goals support requirements.

Chapter 8

Think Aloud Activity

Chapter 4 presented the equality value pattern and the templates associated with the value pattern. The templates are to equip requirements engineers and systems analysts in the process of identifying equality requirements.

In this chapter, the focus is on exploring the process used by requirements engineers in deriving equality requirements and understanding how equality value patterns and templates are adopted in this process.

The participants were asked to use the ***think aloud protocol*** (also known as verbal protocol analysis) [265]. With this protocol the participants would continuously verbally comment aloud on the task they are handling, stating what they were doing, the reasons behind their decisions, their opinions and why [265]. This protocol has been widely used for system design and evaluation [266], engineering design processes [267], usability testing [268–270] and requirements analysis [271]. It is well suited for the current task to explore whether requirements professionals found the equality templates useful, as well as to establish how the use of these templates would influence (either positively or negatively) the elicitation process.

This chapter presents the think aloud activity design and analysis conducted for this research[1].

To evaluate the use and utility of the equality requirements templates, we designed a study whereby two groups of experts undertook the equality requirements identification task for a given requirements brief. In one group, participants carried out the task without having equality templates, while the other group was given the templates and asked to use them during requirements elicitation.

The study is designed with two research questions:
$RQ1$: **What do requirements engineers perceive equality to be?**
$RQ2$: **Do the equality templates facilitate equality requirements elicitation?**

8.1 Study Design

In this study, equality requirements identification is compared between two groups of participants. In one group, participants carried out the task without having equality templates while the other group were given the templates and asked to use them during requirements identification.

8.1.1 Subjects

The subjects were purposively selected [244] as requirements engineering experts and information systems analysts. They represent the intended equality templates

[1]This study was presented at the 5th International Conference on ICT for Sustainability in 2018 [272]

users. The templates were expected to assist them with the identification of equality requirements as well as formulating these requirements for specification documents.

Thirteen participants (6 female and 7 male) were recruited within the age range of 25 - 54 years old. Of these, 8 were academics and 5 were industry practitioners. The participants self-specified their experience levels as 'expert' (2 practitioners), 'advanced' (4 academics and 1 practitioner), 'intermediate' (4 academics and 2 practitioners). The participants come from different backgrounds (Oman, India, Philippines, the UK and Austria).

The participants were allocated into two groups with the best skills and gender balance possible (see Table 8.1). Those in Group 1 undertook the activity *without using the equality templates*, whereas those in Group 2 *used the equality templates.* Participants attempted the activity each at a time since the author was the only facilitator. As the equality value pattern and templates are new in the field, we decided that more number of participants should be in Group 2 to evaluate the usefulness of the equality pattern. In addition, each group undertook the activity only once, this is to reduce the reactive measurement threat [273]. Reactive measurement threat can result from subjects being "aware of the fact of the experiment" [273] and this can occur from repeating the think aloud activity.

For the purpose of readability, the participants are referred to as NoTe-Px (for participant x in 'no template' group) or Te-Px (for participant x in the 'with template' group).

TABLE 8.1: Think aloud participants

Proficiency	Domain	Group 1	Group 2
Advance	Academic	2	2
Intermediate	Academic	1	3
Intermediate	Industrial	1	1
Advance	Industrial	1	0
Expert	Industrial	0	2
Total		5	8

8.1.2 Pilot study

The study was piloted with 4 female software engineers, two of whom were 'advanced' postgraduates in Software Engineering and two were current intermediately-skilled in requirements PhD students. The pilot study was conducted to validate the structure of the think aloud protocol and the set task in preparation for the full study.

The pilot study flagged up several changes to the study design:

1) Simplify the examples provided in the template
2) Use colour coding in the templates to make it simple for participants to follow and track the examples
3) Use the value pattern diagram as extra material.

Upon consideration of the pilot results, the full evaluation study was conducted.

8.1.3 Procedure

The task set before the study participants was to undertake equality requirements elicitation for a given requirements brief. The requirements brief pertained to a smart garden design [241] (refer to Chapter 6).

All participants were given a copy of the information sheet including an overview of the activity and its rationale, as well as the materials for the actual task which included: (i) a form to collect demographic information of the participants; (ii) an instructions sheet detailing precisely what exactly the participants should do (i.e. requirements elicitation for equality concern); and (iii) a stakeholders list. In addition, the Group 2 participants were furnished with the the equality value pattern and its requirements representation templates, which they were asked to use for the elicitation activity (see Appendix E).

As the participants started on the requirements elicitation task, their verbal commentary (with their prior consent) was recorded for further analysis. A brief follow-up interview was conducted upon the task completion.

The study design allowed for up to 30 minutes on the think aloud activity and up to 30 minutes for the follow-up interview. The actual time of each participant's engagement ranged from 30 to 48 minutes. The study was mainly conducted in the English language. However, two participants wished to carry out the verbal commentary in Arabic (their mother tongue). This was acceptable because it helped to remove communication barriers for the participants and made them more comfortable with the activity [274]. Since the author is proficient in Arabic, she translated and transcribed the commentary upon completion.

The activity was run as a relaxed interactive think aloud [269]: at times the researcher could minimise her intervention to acknowledgement tokens (e.g., 'yes', 'ok') in response to the participants' seeking confirmation (e.g., NoTe-P1: "You understand what I'm saying?" Researcher: "Yes."). On occasion, the researcher had to remind the participant to continue verbalising (e.g., telling Te-P3 "Can you please keep talking"). Task continuation interventions [269] were used to encourage participants to continue identifying equality requirements (e.g., "What else do you think?" and "Anything to add?"). With some participants (Te-P6 and NoTe-P4), intervention was used to reduce their anxiety and help them through the activity [275, 276].

8.1.4 Data Analysis

The verbal protocol transcripts were analysed. The analysis was conducted using qualitative text analysis [169]. This was also used by verbal protocol studies in [262, 277–279]. We used directed qualitative content analysis [171] (also known as deductive content analysis by Mayring [280]). This approach is used to support and extend the existing theory. In the current study, the approach is used to provide evidence that notions depicted in the template (discussed in Chapter 4) are considered to be either useful or unhelpful for equality requirements. In addition, we are interested in the relationship between equality template use and the requirements elicitation process; specifically: How do the templates support or impede the elicitation process? Do the requirements identified with the templates differ in any way from those identified without the templates?

Thus, text analysis was initiated with a set of pre-defined codes, where the category codes represented the key notions supported by the template (i.e., stakeholders, variability, goals, services). The pre-defined categories that emerged from the equality value pattern are:

A Stakeholders: refers to "anyone with an interest in, or an effect on," the software is considered as a stakeholder [281].

B Variability: refers to the differences between stakeholders that can cause inequality (also discussed in Chapter 4).

C Goals: refers to what a stakeholder wants to achieve from the software (also discussed in Chapter 4).

D Services: refers to the service or features that will allow stakeholders to achieve their goals (also discussed in Chapter 4).

Additionally, new codes were defined during the coding process to add classifications to the verbalised requirements (e.g., system development, price affordability, inclusion, etc.). This is known as inductive category development [280] where categories/-codes are derived from the data under investigation. Here, there is no hypothesis or theory testing instead, codes are interpretation of the data [282].

The code categories and examples of coded text are listed in Table 8.2. The additional codes (other than those pre-defined) are defined as follows:

A Accessibility: data coded under this category meets the definition and guidelines of accessibility provided in the Web Content Accessibility Guidelines (WCAG)[2]. This includes, for example, alternative input/output medium, alternatives to textual information, readable content and easy navigation.

B Price affordability: data coded under this category refers to the affordability of the software or the affordability of products sold through the software.

C Inclusion: data coded under this category refers to the aim of including stakeholders from different locations, with different languages from different nations

[2] `http://www.w3.org/TR/2008/REC-WCAG20-20081211/`

and other variability factors as defined on the web accessibility initiative page [3].

D Usability: data coded under this category include references to features helping to produce effective, efficient, and satisfying software [3]. In addition, Nielsen usability heuristics[4] formed this category. This includes, for example, error prevention, aesthetics, real-world feel and simplicity.

E Functional requirements: this code refers to the definition provided in Appendix A. It describes software functions as well as the inputs to those functions. This includes, for example, water consumption calculations, watering alerts, inputs and outputs.

F Non-Functional requirements: this code refers to the definition provided in Appendix A such as hardware, software requirements and constraints.

G System development: this code refers to the software project development process such as planning, analysis, testing and programming.

H Knowledge management: data under this category refer to knowledge creation or sharing among software stakeholders.

The final analysis results are reported in Section 8.2 below.

[3] `https://www.w3.org/WAI/intro/usable`
[4] `https://www.nngroup.com/articles/ten-usability-heuristics/`

TABLE 8.2: Additional Category Codes

Code	Group 1		Group 2	
	No Part.	Refs.	No Part.	Refs.
Accessibility	4	17	7	35
Examples:	“some voice recognition technology” “you can Braille I mean the even I am blind I can use the system” “The colours are important so the stakeholders can understand” “the image depict the level”		“should be able to operate the system” “icons which are easily recognizable” “can still use the mouse” “add some sounds” “read the functions for them or get them the message as voice” “ access the history in a faster way. also we can do it as swipe” “gain access, to the core functionality”	
Price affordability	3	8	3	7
Examples:	“providing on affordable device at affordable cost” “people capable to buy”		“should . . . not offer only the expensive plants to buy but should may be also offer cheap things” “will be of cheaper price”	
Inclusion	2	16	6	13
Examples:	“take input from different geographical region” “the people those who are from slum areas or wherever they can come on free basis and they can be able to use this system”		“have a different languages considered in the system” “ I am in Oman so the buttons should be in Arabic”	
Usability	4	51	8	37
Examples:	“very user friendly” “they don’t waste time” “image depict the level” “it will be attractive and easy to use” “2–3 colours not more than that”		“user friendly” “easier for me to use it” “it should make time of showering the plants in a reasonable time” “to use drop-down menus to make it easy to solve the mistakes done while typing” “instructions are important” “explicit tutorial available either inside the application or alongside the application” “we can do all this from one button like instead of having 20 pages I put them all in one”	

Table 8.3: Additional Category Codes (Continued)

Code	Group 1		Group 2	
	No Part.	Refs.	No Part.	Refs.
Functional Requirements	3	29	7	28
Examples:	“need to input how many seedlings are there” “What level of water should create an alarm to the sensor” “view the details of his garden” “(in registration window) will be the details of user name, user ID, email, password, confirmation password, ok and cancel buttons.		“the level of the water and also the time element and how long it should be on this level” “display the report to the user or the history” “acquire or load profiles and watering details” “I should be able to indicate the different kinds of support I expect” “automatically sprinkling the plants” “alerts that a certain location is facing some problems” “inform him that this certain plant is either over watered or needs to be watered”	
Non-Functional Requirements	4	24	7	48
Examples:	“it should be compact . . . (module)” “sprinklers are also attached to the system” “I can go for hybrid application” “(water faucet/valve), there should be a device other than the sensor”		“mobile application”,“proper UPS”,“ get permission” “should be a policy about how much water it can be used for this for this kind of application” “backup server”, “it is wireless”, “the user to only view the people in the same area”	
Knowledge Management	-	-	3	11
Examples:	“provide instruction on how to use the application” “has to know how to do this thing” “have advice, the user might not be gardener and does not know, so the system informs me that the type of soil I have needs 20% watering ” “it can be useful the users view each other experience and experiments, someone can send me a post”			
System Development	4	19	8	23
Examples:	“team of people from different places” “will manually compute also if the actual water level is the same as the expected water level that is generated by this formula” “users are also part of the development” “obtain feedback from the stakeholder”		“If there’s a language barrier so, the analyst should also having the . . . tools to communicate with the one who will use the system” “check if the software is meeting the original requirements” “interaction design model”	

8.2 Results and Discussion

8.2.1 Stakeholders

Although both groups received the stakeholder list, participants with equality templates were more specific when identifying the stakeholders (see Table 8.4). This was expected because the equality template gives instructions to relate stakeholders to various areas of equality requirements. This direct stakeholder identification and relation to requirements within the template is also the reason for the larger number of stakeholders verbalised by the participants from the template-using group (Group 2) than that in the group with no template to use (Group 1), with 156 references to stakeholders vs 39 respectively.

8.2.2 Variability

As shown in Table 8.5, the group with the template (Group 2) verbalised more variability factors (121) than the group with no template (58).

Both groups related equality requirements to education, age, language, disability and income. Although disability was not listed in the template, both groups used it. This might be due to the equality definition provided in the activity materials and due to its commonality in daily life issues.

The group with the template identified additional factors including race, culture, the technology used, knowledge and memorisation capacity. Memorisation capacity was not listed in the template.

Participants Te-P8 and NoTe-P5 verbalised a new variability dimension (previously not considered in the template): the lifestyle of the intended users (fast/busy). This suggests that the template is flexible and does not restrict how participants think.

One participant (NoTe-P2) from Group 1 did not define any human variability factor and did not specify any inclusion, accessibility or usability requirements. This suggests that there is a possible relationship between variability value and the identification of those requirements.

Participants in both groups thought of variability factors not only in terms of stakeholders but also in terms of the plant growth. These variability factors vary based on the plants in the garden and this will affect the calculations and functions performed by the software. For example, NoTe-P2 identified that sunlight, plant age and fertilisers all affect the amount of water that will be calculated by the smart garden and Te-P2 identified the soil type and the location of the garden.

Due to the verbalisation of variability factors within the group without the equality template, it is clear that equality is related to considering diverse people and allowing them to benefit from the system. This conforms to the equality value pattern suggested by this research.

Table 8.4: Stakeholders Elicited Per-Group

Group 1	Group 2.
People, users, team member, local support team, people of (Oman, UK or India), owner (land, garden), [seeds, plantation] analyst, department/ministry of health, consumers [garden produce], society, supplier, gardener, "somebody who will monitor everyday", "people those who are from slum areas", Government, society as indirect stakeholders, "people who are specially monitoring the plants, monitoring the watering in our plants", experts, developer, public compound farmers	Sponsor, consultant, team of operators, Maintenance and operations, we have the people for the sowing and maintaining the garden, procurement department, supplier, help desk, buyers, gardener, flies and insects [as hostile stakeholders], environmentalist, ministry of agriculture, agriculturist, consumer, citizens, legal authority, designers, landscaper, programmer, tester, customer, user, people living within the area of the garden, ministry of environment, consumer, regulator, producers, marketplace, purchaser, owner of home garden, private gardener hired by the owner, maintenance companies, housekeeping companies, household, requirements analyst
No of reference = 39	No of reference = 156

Table 8.5: Variability Elicited Per-Group

Group 1: No Template Used	Group 2: Equality Template Used.
new or existing customers, garden features (size, plant types, number of trees)	Knowledge (technological or gardening), disability (mental capacity), memorisation capacity, race and cultural background.
Common: education, location, language, age, technology used, position, disability (vision, deaf, blind), income status, **lifestyle, e.g.** "don't have time to read", "you are a single Mom, and you have a full time job and a really full plate"	
No of reference = 58	No of reference = 121

8.2.3 Goals

Table 8.6 depicts the goals identified by each group. Table 8.7 depicts the functions, features or services identified by each group. The goals, functions and services overlap with the other categories reported in Section 8.2.4. This is expected because we have structure the equality requirement to include stakeholders, variability, goals and functions, features and services.

8.2.4 Equality Requirements

8.2.4.1 Common Equality Requirements

Both groups identified a number of common groups of equality requirements, including:

- *Accessibility:* e.g. visual representation of information ("[the application should mention the quantity of water and the image depicting the level]" by NoTe-P5); easy navigation ("[access the history in a faster way. Also we can do it as a swipe]" by Te-P6), readability (" we have to make sure that the text is readable" by Te-P4).
- *Affordable price:* the more affordable software is, the wider its user segment (e.g., "providing [software] on affordable device at [an] affordable cost for common people . . . will promote equality" by NoTe-P3; "cheaper price so that the households could use this" by Te-P5; and "the system itself should not be too expensive" by Te-P7). Participant Te-P7 also noted that in case of making commercial exchanges through the software, the exchanged products should be diversified to make the exchange affordable to users. The participant mentioned "the system should . . . not offer only the expensive plants to buy but should maybe also offer cheap things for people that don't have that much income".
- *Inclusion* across geographic locations (e.g. "for instance, I am in Oman so the buttons should be in Arabic" by Te-P5) and languages (e.g. "we must have different languages considered in the system" by Te-P4; and "We can change the language [to] Arabic or English" by NoTe-P1). This also includes preventing exclusion of diversified users communities by facilitating the use of the software. For example, Te-P6 suggested having multiple versions of the

smart garden software so that users who "[need the full features of the software . . . can use the highly priced sensors . . . and if lesser features [are needed] then use sensors costing less]".

Participants also noted the importance of having a flexible system than can be customised to the different gardening practices. Te-P4 noted that "we will have to consider the race, cultur[al] background when it comes to certain plants, . . . In some countries, they have different preferred times to harvest plants; some can prefer to harvest before [they are] totally ripe so we have to consider [this]. The system should have measurements of the plant customised for the country it is being planted in, not all countries". A similar idea was suggested by Te-P8: "the application needs to be flexible when it comes to typical plant requirements in most specific regions".

- *Usability* requirements identified by both groups relate to simplicity, error reduction and efficiency. For example, "to have a very explicit tutorial available either inside the application or alongside the application that demonstrates step-by-step what people need to do to set up their environment" was suggested by Te-P8; "[use drop-down menus to make it easy to solve the mistakes done while typing]" by Te-P6; and "[simple, easy to use, not complicated, not a lot of steps, 5 steps are maximum or 5 clicks even]" by NoTe-P5. Usability was found to be a relevant requirement to environmentally sustainable software [283]. This confirms the overlapping between sustainability dimensions [284].

- *Representative user involvement into software development* was noted as a key avenue that helps to ensure equality (e.g., Te-P5 suggested the use of an interaction design model since it is only by "involving the . . . intended users of these application[s]" where "the users are . . . the centre of the development" that equality would be assured. Similarly, NoTe-P3 stated that "you are going to be selecting the end user representation for agile rapid application or whatever it is. You have to go for some gender equality or some professional equality then the equality will appear in [the] software as well". In addition, participant Te-P7 noted that involving representative stakeholders is important to spot equality issues: "during the requirement identification phase, you need to include stakeholders that are potentially discriminated and need to analyse the system with respect to this". Similar directions for ensuring equality through design models and methods are discussed in related literature on participatory design techniques [22, 285].

- *Platform compatibility* was also noted as a supportive requirement to equality in both groups. Te-P4 noted: "it is going to be compatible to be installed on all . . . operating system[s]". Te-P8 mentioned that "not [to] target the happy few with the most up-to-date operating system but effectively making sure that almost anyone can use it". NoTe-P3 stated: "I can go for android, I can go for iOS development, I can go for hybrid application".

Furthermore, group 2 identified a few group-specific requirements, as discussed below.

8.2.4.2 Equality Requirements With Template

In this group three more requirement categories considered to be relevant to equality were identified:

- *Availability* was noted by Te-P1 (who suggested using hardware components that ensure application availability: "UPS because it has to work 24/7") and Te-P5 (who noted that the application needs to be available for use at any time: "I can just use the application . . . at any time and wherever I am").

- *Knowledge creation and management* category was noted by three participants. Te-P6 noted that equality will mean providing "[instructions on how to use the application]" and allowing gardeners to share their gardening experience through the software where "[the users view each other's experience and experiments]". Te-P2 and Te-P3 suggested that consumers of garden products need to be aware of the smart garden system to know how the produce was grown.

- *Suitability* of the materials used in the software for the user's community background (e.g. race, beliefs, culture, etc.). This was noted by Te-P7 who suggested that images displayed within the software should not offend user community features: "it's very important to for example [to] see pictures of women that are the same race as they have" and they explained that "the figures and picture that . . . [are displayed in the software] should not contain material that discriminates race or gender and religion for example". Te-P6 also noted a similar idea by saying that the software should be built in a way that respects the user's community background: "[a picture is not suitable for us but for other communities it is ok. There are several things similar to the pictures that can affect the use of the software]".

8.2.5 Participants' Viewpoints On Equality

8.2.5.1 Equality as a Maximum Number of Users

Participant NoTe-P1 viewed equality requirements as having the maximum possible number of users. This led to identifying possible usage barriers and requirements that eliminate such barriers, so that the software application can reach the widest possible market (e.g. "anybody can buy and use it"; "many people can use it").

NoTe-P1 considers as barriers what in this research is referred to as stakeholder variability factors (e.g., age, language, geographic location, etc.).

Furthermore, NoTe-P1 viewed supporting equality as a gradual process of addressing the needs of humans: from individuals, to communities, then regions and so on (e.g., "look at the individuals only but when you broaden it you will come to regions or countries").This also necessitated diverse development teams, e.g. "people are there [i.e., in a team] from different geographies", and "local support is needed" because local support teams contributes to equality by better representing local needs and requirements.

Participant NoTe-P5 views equality as allowing all types of gardeners to benefit from the application (e.g. "develop a system that can help them all") by reducing water consumption (e.g."allowing them to use the system based on either the number of trees or the area/size of garden"). The participant started by identifying different types of users and thinking about what each type would need to have in the application. She then sketched a user interface, helping herself to elicit requirements that would relate to interface decisions. Noting that the users "don't have time to read the information" and software use manuals, she suggested representing information and interfaces as game icons: "like the Super Mario game, it shows the water, the water coloured in blue, trees in green in the application. . . . Consumption status indicator with colours: red, orange, yellow and green".

Similarly, Te-P6 used the template to find and anticipate problems that could be faced by users and define requirements to solve them (e.g., "[we can see that language can be a barrier to using the system. The template gives us the functions that the system should have in addition to the goals]"). Te-P6 thinks that the purpose of the template is: "[to make it easier for the developer to know the requirements of an application. He can find what . . . the problems currently faced by a user are and what the suggested solutions and goals are]".

Participant Te-P8 also viewed equality as an "equal number of users or to reach a fair number of users . . . support as much of the market as possible".

8.2.5.2 Equality as a Shared Responsibility for Software Project

Participant Te-P1 defined equality as allowing all stakeholders to share responsibility for making the software development project a success. This could be achieved by undertaking responsibilities and duties in the project: "So here, everyone has equal responsibility", " it is not one person who is running [the project]. All the stakeholders . . . mentioned here are part of this, equally involved in this".

Participant Te-P5 held the same view, with the slight difference that the users are to be considered the central players and engage in "interaction design".

Consequently, the template was primarily used to identify stakeholders who need to be part of the project, whether they represent individuals, departments or other systems.

8.2.5.3 Equality as Indirect Stakeholder Support

Participant NoTe-P3 considered indirect stakeholders to be the main focus of equality, so this category of stakeholders should not be neglected (e.g "for each and every system, certain social stakeholders or certain indirect stakeholders from [a] social aspect have to be added"). Negative stakeholders too must be given a consideration or else the "equality perspective will be in question". However, NoTe-P3 thinks that when software is developed for internal use in a specific company then, as a developer, one should not "bother much about the equality of the society and all those factors". This is because such a system supposedly does not have either any societal impact or any indirect stakeholders. This is clearly an oversimplified view on equality, blinded to the diversity of the software system's stakeholders within the company (whether direct or indirect).

NoTe-P3 further relates equality to non-functional requirements such as affordability and clarity (e.g. " clarity . . . is [a] fundamental requirement for equality. If clarity aspects [are] not there then inequality will come" because the software developers will interpret the requirements on their own, potentially neglecting equality and societal concerns).

8.2.5.4 System Functions vs. Equality

Participant NoTe-P4 stated that there is no relationship between equality and the smart garden. This is because she regards equality to be narrowly defined as economic equality, which is supported by dedicated applications that monitor and report on economic differences between people. The other avenue through which software relate to equality, according to NoTe-P4, is when applications are specifically targeted at people with special needs and the deprived, e.g., for creating job opportunities (e.g. "if I am a disabled person . . . and you give me the software that is really related to me giving me an opportunity to work"). However, in any other contexts, NoTe-P4 emphasised the difficulty of relating the concept of equality to software requirements. Only after viewing the templates as part of the follow-up interview did the participant started to grasp the possible connection between equality and the software. For example, the participant said: "the equal life knowledge to all the house owners because we are giving the video that how to plant you are giving the opportunities for them to take care of the plants".

Participant Te-P7 was of the same opinion. However, she was able to get requirements that support different users through the user interface being suitable for the differences. The participant noted that "I can not really imagine anything in such an application that would discriminate [against] somebody really apart from like the system or user interface being so difficult or not suitable for old people. So, the system itself, with respect to its functionality, I think its neutral".

8.2.5.5 System Functions and Stakeholder Selection

Participant Te-P2 worked by using the template to gather what goals/tasks the current system is to complete. This system "as-it-is" [286] review is a usual starting point in requirements analysis, whereby the identified tasks are transformed into the requirements statements for the software system. Te-P2 proceeded to consider:

- Stakeholders as part of the system analysis team;
- Variability factors as selection criteria for the representative stakeholders to be included in the analysis of the system (e.g."they have that right knowledge in the gardening . . . the knowledge, education and the technology use and a location should be indicated here . . . [as well as] the age."). A similar point was suggested by Te-P7 that to ensure equality, different stakeholders with variability factors should be involved by stating: "during the requirement identification phase, you need to include stakeholders that are potentially discriminated").
- Variability factors as selection criteria for testers to be involved in the testing stage of the software ("the knowledge of the tester, [he/she] should be . . . familiar with the technology to use on the garden but definitely he should also [be] knowledgeable . . . in the garden[ing] because the IT who is knowledgeable of gadget[s] and the person who will test the amount of water placed on the garden is in the garden. So both of them should team up").

8.2.5.6 Equality as an Implied Concern

Participant Te-P3 viewed equality as part of accessibility and usability: "Normally we sum it up in one requirement like 'user friendly' . . . [b]ut when we talk of user friendly, inherently we are talking of equality because regardless of age, of gender or . . . the skill of the person" they should be able to use the system. Thus, equality is not an isolated requirement but "it's implied" as part of several other usability-related requirements. To this participant, the templates served as a reminder: "Normally we tend to omit some . . . variability but using your template it will keep on reminding us that this factor will also be important in the system that you are trying to produce".

Similarly, participant NoTe-P2 considers equality to be an aspect embedded in software engineering principles, stating that equality is similar to the principles followed by engineers, where an engineer must consider the impact of their product "for society, for the customers, . . . with regards to, for example, privacy or patents or IP" which are notions considered as part of the professionalism and ethics. The participant notes that although "equality is there already" within professionalism and ethics, "to make it as one whole concept or one whole section of the requirement, this [is] my first time". To NoTe-P2, equality should be considered not only with respect to humans and "society as a whole" but also with respect to other creatures who may be stakeholders in the system as well. Thus, for the smart gardening system she notes the need to consider the rights of the plants.

Participant Te-P3 stated: "Think this is the first time that you have explicitly mentioned about equality requirements but as I've said, it's there already". Participant Te-P7 stated: "I think accessibility requirement[s] are a subset of equality".

8.2.5.7 Equality as Stakeholders' Goal Support

Participant Te-P4 used the template to identify stakeholders and analyse how the system could support their goals through software functions and features. She noted that variability can help refine a software service (e.g. in discussing how an application alters the user, she starts with smart phone users, then notes that "Some people use the cell phones only for texting and talking so maybe this person['s] education or technology background doesn't help in using . . . software applications. So . . . [the] system can send him a text message if it's needed". She then observed that the age variability will require larger print fits for older people and blind users will need to be supported by voice messages.

8.2.6 Equality Requirements Identification Process

By analysing the process that the two groups of requirements engineers used to identify equality requirements, we note that, irrespective of the template use, both groups looked at the variability factors and considered their effect on software functions and features, as well as at satisfying the stakeholders' goals through a software system. Thus, it is clear that **stakeholders, variability, functions/features/services and goals are relevant contributors to the equality requirements** and are commonly used by the requirements analysts.

At the same time, we observe that the equality **elicitation process can start with any of the named factors and the sequence can vary depending on the analyst's perceptions, preferences and convenience**.

Where the analysts were asked to use the templates, they **tailored the template to their understanding of equality**. For example:

- Using stakeholder variability characteristics as possible obstacles in reaching as many users as possible, and defining requirements to resolve these 'obstacles';
- Using the template in combination with the stakeholder list (taken form the Onion model), whereby some participants alternated between stakeholder selection and goal/function definition, where goals were defined and refined into functions for each stakeholder (e.g., Te-P3, Te-P4), filling in the template row by row, while others started by identifying all possible stakeholders and then reviewing all variabilities related to one stakeholder, i.e., aiming to fill the template in a more of a vertical manner (column by column fashion, as per Te-P5);
- Goal and function were not used on all occasions (e.g., Te-P2), especially when the participants were working on the understanding the current functions of the intended software domain (i.e., gardening). At this point, analysis's focused on stakeholders and what information they hold. More specific goals and functions for the software system would follow on form this. In some cases goals come before and in some cases after functions (e.g. Te-P4, Te-P6).

8.2.7 Template Utility

Overall, Group 2 participants commented that the template was useful for identifying equality requirements. Participants commented that the template is simple (Te-P4, Te-P6), clear (Te-P3, Te-P4), well organised (Te-P6), and sufficiently detailed (Te-P2). Participant Te-P6 noted that the template would be useful for system analysts to "extract" user requirements. Participant Te-P8 noted that the templates "look very familiar and yet they elicit some new thought".

Participants Te-P3 and Te-P4 commented that the templates serve as good reminder of what should be considered by the analyst with regards to equality; e.g. Te-P3 stated that: "Normally we tend to omit some . . . variability, . . . the template will keep on reminding us that this factor will also be important in the system", and Te-P4 said: "It helped me a lot; I was returning each time to it". According to Te-P3 and Te-P8, the most useful part of the template in stimulating and recognising equality issues is the variability value. Te-P3 stated that "normally when we talk of requirements we sometimes don't specify the variability . . . the distinguishing thing that I noticed about this [template] is the variables . . . you consider". Similarly, participant Te-P8 recognised that variability (in the template) is the most important part that helps to spot equality issues: "it is definitely the variability . . . that will determine the range you have to provide in order to give all these people an equal experience or if not equal, at least equivalent". In addition, participant Te-P8 noted that "by using these kinds of templates, you actually make it much more clear what you have to think about so I think the templates really help in focusing the discussion which is really what you need for software development". Moreover, participant Te-P7 mentioned that the template helps in relating and identifying which stakeholders need to take part at the requirements analysis and elicitation phase of RE by noting that: "the link to the stakeholders could be something that is interesting because it could like ask these stakeholders or ask them for further information".

Furthermore, participant Te-P8 found the template to be useful because it "pushes me out of my comfort zone of assuming that I completely understand my user which I think is the biggest difference; the biggest danger in all software development".

A participant from Group 1 (who was not given the template during the experiment) had shown a level of misunderstanding about equality and its relation to software requirements. Participant NoTe-P4 stated that there is no relationship between equality and the smart garden application because to her equality is being "treated as human beings". This could be a common issue due to practitioners' lack of knowledge about social sustainability that might make it difficult to relate sustainability concepts to software requirements as the study in [10] reported. However, templates can remove this difficulty as participant Te-P8 stated: "you would ask ok, What are the equality concerns here? For example, I think in abstract people would not really identify these kinds of things, maybe because in your head, it always makes sense, right? But by using these kinds of templates, you actually make it much more clear what you have to think about". Te-P7 had a similar idea: "I think the smart garden thing is such a neutral I would say gender, race, religion, whatever neutral system". Yet, she was able to continue the task and elicited equality requirements.

NoTe-P2 and NoTe-P1 thought that equality is often implied as part of other requirements (e.g. NoTe-P1 stated: "when you say portability, equality may not come in the first phase but when you dig [deeper], equality is there"). NoTe-P2 further stated that the present study was the first time that she had seen the notion of equality as "whole concept or one whole section of the requirement". This indicates that software developers may omit equality concerns because they do not see the link between equality and software requirements. They may also part-specify equality because it relates to other concerns, but fail to consider the notion as a whole.

Participant Te-P8 identified requirements using user stories. The generated user stories were checked against the equality template to derive equality requirements. The participant stated that: "I will update my requirement specification for my user interface to effectively be . . . very easy to add any language in". The participant also added that after using the template, he will look for possible ways to reach a wider market by "making the application as light weight as possible. So, in other words, not [to] target the happy few with the most up-to-date operating system but effectively making sure that almost anyone can use it, and in addition, also try to see if it's possible within reason to have an interface that can be accessed another way, effectively making it easier for people to access and not necessarily have a fancy smart phone to do it". In addition, the participant added that due to users' variable educational level, a function of configuring the garden and its details was extended to an "auto detect" of garden setup. The same requirement was also extended to include an "explicit tutorial available either inside the application or alongside the application that demonstrates step-by-step what people need to do to set up their environment". Here, we see that the equality template is useful for deriving more detailed as well as missing useful equality requirements.

Likewise, participant Te-P7 used the template to check if a requirement supports equality. The participants said that "if the system also support[s] for example . . . buying something through the system then it should not only contain very expensive things and the system itself should not be too expensive". When asked why is this an equality requirement, the participant replied: "because you have this different income as an example [provided in the template] then I would say it could be such a[n equality] requirement, yeah".

This shows that the equality template can be integrated as part of the usual and familiar requirements engineering practice without the need for having special experts in social sustainability. It also shows that there is no need for special social sustainability sessions or meetings as it can be incorporated into the currently used practice. This will contribute to project time efficiency as more aspects are integrated (equality) within the same available time with no need for specialised/dedicated meetings for social concerns. This suggests template efficiency.

All of this demonstrates the following key areas of the present template utility:

- The template *tackles the issue of conceptual misunderstanding of equality* by depicting a direct and clear association between the equality concept and software requirements;

- The template *consolidates the notion of equality into a coherent structure* rather than leaving it as implied across several other concerns;

- The template acts as a *visual reminder* of the variability dimensions that should be considered for all stakeholder types to assure equality;

- The template can be used as a *tool for checking requirements coverage of equality* requirements.

- One of the uses of the equality template is during *brainstorming activities* (as a technique used for requirements elicitation [287, 288]) with single or multiple participants. Suggestions from participants such as Te-P8 ("I would take the features what we have discussed and set up an environment and create a prototype" and "I think the templates really help in focusing the discussion which is really what you need for software development"); and Te-P5 ("So for this template . . . it would be better if you will use also the cloud to gather the requirement . . . to gather . . . feedback from the users . . . So I think a link with this format would make life easy for you as a researcher") shaped such ideas. Participant Te-P5's suggestion implies a possible use of the equality template in collaborative requirements gathering.

- A possible benefit of using the template is that it will help to *ease communication with higher management* because it makes clear assumptions about equality and shows where are the areas that needs the management support and they are out of the control of the developer. As participant Te-P8 noted: "Essentially it is more of a concern that you want to be able to communicate to higher management . . . Your template will help in identifying this".

  In a way, this use of the template will also contribute to social sustainability as an organisational sustainability factor. Organisational sustainability was defined as a sub-dimension of social sustainability [289]. One of the aims of organisational sustainability is to maintain communications between stakeholders.

- An inherited (from templates) use of equality template is its usage in *documenting equality requirements* as it provides a structure for writhing the requirements.

8.2.8 Other Avenues for Equality Support

During the follow-up interview, the participants were asked how they would suggest improving equality consideration in software engineering. Some of the suggestions provided are outlined below:

- Equality metrics should be used from the early stages of development. These metrics should ideally be "already embedded in the methodology" (NoTe-P3). This is because the metrics that require measurement and monitoring without integration into the originally used methodology will involve other tasks and will not be well received by the developers. Groher and Weinreich reported that the lack of such measuring tools contributes to sustainability challenges in software development [290].

- Equality standards could be defined by the professional bodies in SE "like, for example, when we have the principle with the ACM standard so we can have that as guideline" (NoTe-P2).

- Equality concerns could be taught as part of the software engineering curriculum, ensuring that the students (who are future practitioners) are familiar with equality concerns and solutions, thus integrating these into their own developed products (NoTe-P2).

- Equality criteria checklists can be created to validate that the equality concern is adequately handled in the given requirements (NoTe-P4).
- Governments could legislate to require consideration of equality in all software products (NoTe-P2, NoTe-P4, Te-P7). To the best of our knowledge, there is no specific guidelines/regulation that impose equality on software requirements. Except for the general equality laws that can also be applicable to software development environment (e.g. gender equality in involving software engineers, male/female, not specific to software functional/non-functional requirements) discussed in Chapter 3 Section 3.4.1, we did not come across equality supporting guidelines/regulations for software requirements. This was also revealed during this study as none of the participants referred to software equality laws that they are already aware of.
- Guidelines on scoping would be useful, as pointed in the feedback from participant Te-P8. The participant stated that it "would be really great [ ] to have [ ] some form of guidance on how to deal with these kind of concerns that kind of jump over the boundary of software development and so, it is very easy to get lost in the process where you are trying to please everyone" (Te-P8).

We observe that all of the provided suggestions are complementary to the use of a template, and could be integrated with the template use process.

8.3 Threats to Validity

8.3.1 External Validity

A possible threat to external validity is related to sample size which can affect the generalisability of the findings. However, this is a common practice in think aloud [271]. Nielsen estimated the sufficient number of participants in a think aloud activity to be 4±1 [291]. Other studies have used small sample sizes in think aloud experiments [292] with 5 participants in each group.

As the coding of the activity result was conducted by the author, inter-rater reliability test [126] is not possible to be computed which raises a threat to validity. To mitigate this threat, clear definitions of codes were maintained to ensure consistency as well as utilisation of a mixture of coding strategies i.e. deductive and inductive categories development (also discussed in Section 8.1.4). In addition, a second coder (supervisor) occasionally cross-checked and validated the coding and the inferences.

Another threat arises from the allocation of participants into groups. The two groups were formed to represent a balanced grouping of gender, experience and background. The participants' reported experience was used instead of an experimentally validated level of individual expertise. This raises the possibility that more skilled participants are grouped together in one of the groups. However and based on the participants feedback, the importance and effectiveness of the template was evident.

Another possible threat arises from the software case provided (i.e. smart garden). The case is a simplification of a real-world case with limited resources to the participants to allow activity completion within the allocated time. To restrict this effect, the participants were advised to use assumptions to cover gaps and complete any missing information. In addition, the software case used may limit generalisability. Nonetheless, the reported equality requirements are common in the domain and applicable to different software applications.

A possible threat could be related to the participants' knowledge of the technique used (think aloud). We did not specify prior to the task that participants should have prior knowledge in this area. Among the participants, only one was found to be familiar with this type of activity (Te-P5). We tried to mitigate this problem by sending the information sheet (explaining that we are using think aloud) before meeting the participants but this threat can not be totally avoided.

A possible threat arose from the sample selection because they were from author's personal contacts. Thus, the findings may not be generalisable. We have tried to mitigate this threat by selecting people from various backgrounds and countries. It is also important to note that although they are known to us, we have no personal relationships or shared interests with them.

8.3.2 Conclusion Validity

Because this study included two groups, a possible threat results from the implementation of the activity. As much as possible, the implementation was standardised and the provided materials and minimum timings were followed. In addition, the follow-up interview was a semi-structured interview and an interview guide was used.

8.3.3 Internal Validity

By conducting the pilot test as described in Section 8.1.2, a possible threat related to the instrumentation (i.e. the materials used in the activity) was mitigated.

8.4 Summary

This chapter has presented an evaluation of the equality template. The evaluation was undertaken by experts who participated in a think aloud activity. We set out to investigate two research questions: *What do requirements engineers perceive as equality?; Do the equality templates facilitate the equality requirements elicitation?*

For the first question we found that experts perceive equality in terms of accessibility, price affordability, inclusion and accommodation of users with differences, usability, user involvement and representation in the software development process.

For the second question, the templates helped to bring the otherwise distributed and implicit notion of equality into explicit view. They serve as a reminder to address different variability notions that otherwise could be omitted or forgotten. Most importantly, the templates tackled the conceptual misunderstanding that equality is not relevant to some applications. Moreover, the template can be used as a guideline to be used during different phases of software development.

Table 8.6: Goals Elicited Per-Group

Group 1: No Template Used	Group 2: Equality Template Used.
“bilingual part has to be implemented” “for everybody use” “should be very user friendly” “to sell it abroad on different places” “market it in different regions” “Age should not be a barrier. Both lower side as a well as the aged people upper, higher side” “able to buy this product” “assess how many seedlings will be planted” “safety of the public” “analyse the land area” “ best growth for the plants” “can redirect it to others” “So it shouldn’t be fixed” “able to use the system in the same way” “affordable” “equality in your requirement analysis” “respond to everybody in the same manner” “beautiful and simple not very crowded” “users to be equal in use of the water” “control the amounts” “attractive” “not complicated” “they will be familiar with what kind of system are there”‘user participation” “gender equality” “to satisfy most of the customers, stakeholders” “eyes will not get damaged”	“equally involved in this” “I want our water level to be maintained” “will not manually tender it” “lessen physically the job of the gardener to water” “should be able to operate the system” “can still use the system” “monitor and check” “ text is readable” “same benefits” “be suitable to the countries who will use this software” “anyone can understand” “to provide a more accurate amount of water” “lessening the efforts exerted by the . . . maintenance or housekeeping people” “easy to solve the mistakes done while typing” “decide to water the plant or no” “allow people to communicate with each other” “allow sharing their experience” “usable for different kinds of people” “people . . . speaking different languages should be supported” “treats people with less money equal” “it should also work for people that are new” “modifying their settings” “application needs to be flexible” “more efficiently use water” “avoiding an unnecessarily expensive contract with a single provider” “getting dedicated water and support for my particular garden configuration”
No of reference = 149	No of reference = 168

Table 8.7: Functions/Features Elicited Per-Group

Group 1: No Template Used	Group 2: Equality Template Used.
“change the language Arabic or English” “pictorial representation” “List of values” “smart application or mobile” list of plants to be displayed “local support” “input from different geographical region” “installed on a mobile” “select first a model area the area” “have to calculate . . . number of seeds” “find out the level of water that is being consumed” “color navigation” “registration” “click on ok or cancel” “to upload the picture” “view the details of his garden” “calculate and display the saved water quantity” “2–3 colours not more” “he can print it” “log out” “input how many seedlings are there” “reaches this line then the system will be alarmed” “there is calculation in average water per seed” “calculating for the enough amount of water” “voice recognition technology” “User name, password” “date has to be included similar to bank statements” “can get an email” “Consumption status indicator with colours (red, orange, yellow, green)” “also with charts” “will display numbers” “(in registration window) will be the details of user name, user ID, email, password, confirmation password, ok and cancel buttons”	“proper UPS” “timer is kept, its sprinkling time” “schedule of providing water to the plants” “the type of soil” “measure the amount of water level in the bid” “failsafe mechanism” “trigger” “select location of the devices” “alerts that a certain location is facing some problems” “receive an alert” “sound you can add some sounds” “get them the message as voice” “have a different languages considered in the system” “get summary of . . . information” “amount of water it will sprinkler” “open or to switch the applications” “buttons should be in Arabic” “set the sensor” “display the report to the user or the history” “drop-down menus” “instructions and advice” “pictures should not be something that discriminates” “switch languages” “acquire or load profiles and watering details” “phone in based help desk, is it online chat, is it an engineer coming to your house” “input the details of my garden” “display that says please give me the details of your garden, so say initials start-up of the system or on the main menu configure your garden” “explicit tutorial available either inside the application or alongside the application” “a video or it could be onscreen hints”
No of reference = 118	No of reference = 120

Chapter 9

Conclusion and Future Work

Social sustainability is often defined as the ability of a society to maintain its "social capital" which creates the "basic framework for society", including "cohesion of community for mutual benefit, connectedness between groups of people, . . . standards of . . . ethics . . . , rules, laws, and information" [40]. It lowers the cost of working together and facilitates cooperation; e.g., "trust lowers transaction costs" [40].

As software engineers, we foster continuous and deep integration of the software systems into this very 'basic framework for society', yet, as previously discussed in this thesis, presently there is a serious lack of a generic and reusable methodology that enables systematic analysis and integration of *social sustainability requirements* into the requirements engineering process.

Furthermore, this is exacerbated within the software engineering community by the lack of understanding as to what social sustainability means and how it could be related to specific software projects. Thus, the *main aim of this thesis is to derive such a methodology.*

Given that the present software engineering scene has a wide variety of engineering practices (e.g., from waterfall to agile development), we find it imperative that any proposed methodology should be both compatible with the current engineering processes, easily integrated with such practices, and be equally accessible to both novice and experienced requirements engineers. To this end, we set out the following objectives:

1. **Conceptual:**
    - Develop a methodology that enables requirements engineers to integrate (otherwise implicit) social sustainability requirements into software systems specification.
    - Ensure that the methodology is applicable to various social sustainability aspects.
2. **Technical:**
    - Ensure that the developed methodology can be integrated with the (most) current RE processes (e.g. from waterfall to agile development) without imposing any restrictions.
    - Ensure that the requirements related to social sustainability are reusable yet customisable within different projects/contexts.
    - Ensure that the developed methodology is amenable for use by both novice and experienced practitioners.

In the following, we discuss how these objectives have been addressed.

9.1 Objectives Re-visited

Integrate Social Sustainability Concerns into Software Systems Specification.

In order to integrate social sustainability into software engineering processes, we had to first present the abstract notion of social sustainability in such terms and structures that requirements engineers would be able to relate to. Then we had to provide the mechanisms through which the engineers could act upon the said structures.

In a bid to elicit the structures that underpin social sustainability, we undertook a systematic analysis of the social sustainability literature (as detailed in Chapter 2). By looking at what indicators and metrics are reported to be relevant to the notion of social sustainability, we identified 12 distinct and explicit topics, all of which have been discussed as contributing to social sustainability. However, each one of these topics (e.g., health, equality, security, education, etc.) constitutes a large domain on its own.

We find that: *social sustainability constitutes a number of distinct domains. In order for a socio-technical system to holistically address social sustainability, requirements for each of its constituent domains must be addressed, along with the conflicts and inconsistencies that may arise between them.*

We further observed that the structure of social sustainability concerns that emerged from our systematic literature review is closely aligned with the notions of societal values (as discussed in Chapter 3).

Thus, we find that: The notion of *'social sustainability' is formed around and driven by the societal values.*

As previous work [143, 145, 148] has observed that many social values are universal across various societies (though often adapted to a specific context), we postulate that so will be social sustainability concerns and their respective operationalisations in requirements.

Given the large set and scope of each of the social sustainability constituent topics, we chose to develop a methodology for the operationalisation of each of these value-centred constituent concerns. Given that equality is both a key contributor to social sustainability (through its notions of fairness, justice, and equity) and also a rather poorly addressed topic in current requirements research, we opted to use it as the demonstrative domain for further analysis.

As a side note, we acknowledge that this methodology would be to some degree incomplete as the interdependencies and conflicts between requirements that belong to different topics are not addressed here. Yet, these conflicts and dependencies are not different from conflicts and dependencies between any other requirements. Therefore, we refer the dependency and conflict analysis issues to the established RE techniques (e.g. [204, 293, 294]).

Thus, we turned to the equality-related literature and used qualitative content analysis techniques to understand the sub-values of equality and their interrelationships (as detailed in Chapter 4).

This resulted in the development of an equality value pattern: a pattern into which the varied perceptions of equality coalesce, as per the reviewed literature.

In order to make this pattern directly usable by the requirements engineers, we operationalised it through simple questions into a template (as detailed in Chapter 4).

We also proposed a requirements elicitation method using the value pattern. The method is complemented with usage guidelines as detailed in Chapter 5.

We knowledge that limitation of inter-rater reliability as the analysis and coding was done by the author. This is a limitation.

Upon application of the equality value pattern and templates (using the elicitation method) to seven example requirements specifications (developed by others prior to this work), we identified a set of reusable requirements that support equality (as presented in Section 6.2). A small group of expert requirements engineers commented upon the results for equality requirements derived using these templates for a specific study and found these results to be useful (as discussed in Section 6.3) and the pattern and templates to be usable (as per the think aloud activity discussed in Chapter 8).

Thus, we have presented a methodology (exemplified upon the equality concern) through which all of the constituents of social sustainability could be mapped into patterns and templates that can be operationalised.

However, while the values have been acknowledged to be universal [143, 145, 148], we still needed to validate that the requirements we derived through our value pattern analysis were also applicable across communities and cultures. Thus, the common equality requirements derived through the equality pattern were evaluated by software users from several countries, cultures and religions via an online survey (as discussed in Chapter 7). The survey participants demonstrated a generic agreement that the derived requirements were indeed relevant and conducive to the equality in software. Moreover, there was overall agreement (across religious and cultural boundaries) regarding the ranking and relevance of the requirements, although some differences were observed across the various demographic groups.

Thus, we conclude that the proposed pattern, templates and their respective requirements are generalisable and reusable across contexts and communities, although various aspects of these requirements would need to be customised across different communities.

Indeed, taking a more fragmented view of equality, we note that other researchers have reported on generic guidelines for handling issues that contribute to equality (e.g. education in [295] or treating accessibility and usability characteristics [219]).

Methodology Applicability to Various Social Sustainability Aspects

Finally, although the instantiation of the proposed technique is only conducted with respect to equality, the methodological steps used for are equally reusable for other sub-concerns of social sustainability. Thus, the same analysis and operationalisation approach could be used to address each of the other remaining areas.

Integration with RE Processes

While the use of the proposed pattern, templates and requirements does not proscribe any specific development practices, it can be applied in processes where RE is clearly defined (e.g., in waterfall or spiral development) because it simply augments the requirements elicitation and analysis. However, we wished to evaluate the usability of the patterns and templates within less structured requirement processes. Therefore, a study into the use of the pattern within the agile setting was performed (as discussed in Section 6.5). However, the limited results of the equality requirements elicitation method in agile methods is a limitation of our research.

Furthermore, discussions with a requirements engineering expert (presented in Section 8.2.7, participant Te-P8), highlighted his opinion on the utility of using templates to check user stories coverage for equality requirements within an agile setting.

Thus, we conclude that it is both possible and useful to integrate the pattern and templates use within the development processes (such as agile) where requirements engineering is not a clear-cut stage.

Reusability of Templates and Patterns

Given that equality requirements are derived through application of the pattern and templates, it is not surprising that many of them are repeated and, therefore can be reused in various domains (as was discussed in Chapter 6).

Additionally, RE experts (as detailed in Section 8.2.7) discussed the (re-)usability of the templates to check for user stories coverage of equality requirements, as well as in brainstorming for requirements elicitation, and collaborative requirements gathering.

Equality Pattern and Templates and Requirements Engineers Expertise' level

We demonstrated that the equality value and templates were used by practitioners with different RE expertise during a think aloud activity for requirements elicitation (presented in Chapter 8). The equality template helped to provide a clear and direct path for operationalising equality value into a set of software requirements. In addition, the templates worked as visual reminders of what is to be considered to ensure equality. Furthermore, the templates were used as checking tools for requirements coverage.

9.2 Conclusion

In conclusion, this thesis draws the attention of the RE community to the notion of the social sustainability concern and demonstrates that it is possible to operationalise and integrate it (through its constituent sub-domains) into the requirements engineering proper.

In fact, fragments of social sustainability are already tackled in the RE (such as usability, accessibility, etc.), without realising their contribution to the holistic notion of social sustainability.

In this thesis we argue that social sustainability in software should be approached as the requirements driven by the amalgamation of many common and shared societal values of the software user and developer communities. Thus, integrating social sustainability into software is necessary if the software is to relate to and be accepted by its user communities. The proposed methodology for the integration of social sustainability into software (as exemplified by the case of equality) includes understanding social sustainability values and their manifestation through software requirements. The approach (depicted in Figure 9.1) leads to the identification of equality value patterns and their operationalisation in templates.

The obtained templates for the case of equality were evaluated by practitioners and the resulting requirements were evaluated by software users. From these evaluations, we observe that the use of the templates alleviates the problem of vagueness that comes with the notion of social sustainability in general, and also its specific subparts (e.g. equality), thus hindering the integration of their requirements into the software system specifications.

Moreover, with the value patterns and templates, there is sufficient scope to tailor the values (i.e. equality) and their respective requirements to the specific community and software context, allowing us to use the "preconceived values and a context-specific description of value . . . as complementary to one another" [296].

9.3 Future Research Directions

While this thesis has taken the first step towards de-mystifying the notion of social sustainability for the integration of its requirements into the software specification, we are still a long way away from realising full integration of the holistic social sustainability support. Some issues to be addressed to towards this while building on the present work include:

1. Treatment of other topics within social sustainability

   As demonstrated in Figure 2.3, social sustainability constitutes at least 12 independent topics, of which equality is only one. The current research operationalises the equality topic of social sustainability. The other topics are yet to be addressed and although we are confident that the methodology of value-pattern derivation, followed by their operationalisation through templates will deliver requirements that support each of the identified social sustainability topics, this work is remains to be done.

2. Review of inter-value/quality dependencies

   Once the requirements for individual areas of social sustainability are treated, as per point (1) above, it would inevitably be necessary to consider the inter-value/quality dependencies, conflicts and synergies. Again, we have noted previously that the established conflict identification and analysis techniques in RE could suffice to address these issues but this remains to be confirmed and carried out. Only when these issues are addressed, it will be possible to provide a comprehensive view of social sustainability in relation to software requirements.

3. Practical applicability

   Having presented the requirements that emerge from a single topic (out of at least 12) for social sustainability, we observe that the share volume of the relevant requirements is likely to be overwhelming. An open challenge to be addressed in this respect is: how can the multitude of social sustainability requirements be integrated into each software system in a practically feasible way (i.e. within the time and cost constraints of given, often small, projects)? It is possible that (parts of) such requirements may be supported by reusable frameworks but this is an open topic in sustainability research.

4. Applicability of the proposed pattern and templates in agile methods

   Having said that the study in Section 6.5 have limited results, we would like to evaluate the use of the value patterns and elicitation methods (starting with equality) within agile methodologies.

5. Industrial and broader evaluation

   We would like to evaluate the use of the value patterns (starting with that for equality, as presented in this work) within the end-to-end software development projects. We have already identified a number of areas where the pattern and templates, as well as the related requirements can be utilised.

   For example, while the pattern and templates are directly designed for requirements elicitation and refinement, they could also be used during testing activities [33] and requirements coverage assessment.

   They are also noted to be of value for brainstorming activities and even possibly even for legal compliance checks (e.g. equality legislation relating to non-discrimination of the protected characteristics).

   At present, the collection of a corpus of studies in which the templates are applied and the identification of their further uses remains a matter for future research studies.

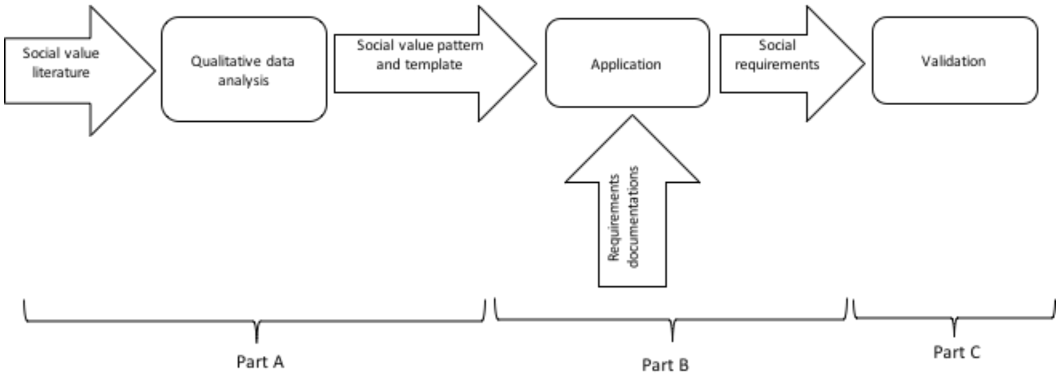


FIGURE 9.1: Social sustainability integration approach

Appendix A

Background Concepts

A.1 Software Requirements Engineering

Requirements engineering (RE) is the process of requirements elicitation, requirements modelling and analysis, requirements communication, requirements agreement and evolution [44]. It starts with preparation where the objectives of the software are identified [33]. Then, requirements are gathered or elicited from people using interviews for example, from documentations and/or from the existing system [33]. Then requirements are documented in requirements specifications document, reviewed and updated [33].

A study involveing 228 companies operating in 10 countries found that incomplete and/or hidden requirements are problems in RE that are relevant to project failure [30]. The problem can also affect customer satisfaction and product acceptance [30]. Moreover, it was reported that lack of experience and the weak qualifications of the RE team are among the common causes of RE problems. *Techniques that support identification of implicit requirements related to social and personal well being can* *thus improve social/individual effect of software systems and improve satisfaction of their user communities.*

Liu, Li and Peng reported the results of a survey conducted in China involving RE practitioners [31]. The survey reported that the third-most common requirements elicitation technique used is by referring to similar existing systems [31]. This provides an insight into the importance of having reusable requirements and requirements elicitation techniques that can be tailored to fit different software systems. Furthermore, the study reported the common causes of RE failure. One reason is the lack of domain expertise of RE practitioner [31].

Software requirements describe what a software needs to do [33]. It is a "measurable objective" to be satisfied by the software [33]. During requirements elicitation, the system boundaries (where the system fits) are identified as well as the stakeholders and goals [44].

Goals represents the objectives that a software needs to satisfy. One important role of goal modelling in requirements engineering is to address future objectives and identify how to operationalise them into system components (functional, non-functional) [297]. Operationalisation is the transformational process of abstract and intangible concepts and properties into measurable specifications that satisfy them [212, 298]. This is achieved in several ways. Letier, and van Lamsweerde suggested a formal approach based on KAOS [298]. KAOS is a formal goal modelling language used for requirements identification [299]. Others have incorporated goals with use cases [300, 301]. Scenarios were also used in operationalising goals such as in [302].

A.2 Stakeholder identification

Requirements originate from stakeholders [36]. Robertson and Robertson define stakeholders as entities interested in a software or that exert an influence on it [36]. Defining the right stakeholders of a software is an essential task during RE. This is because stakeholders provide their opinions about what the software should do or look like. Their role is an important role and requirements engineers work closely with stakeholders to identify software requirements [41]. Studies have suggested several methods for stakeholders identification such as [303, 304].

Alexander suggested a taxonomy of stakeholders called the onion model [218]. The model has four circles representing the different stakeholders with their different roles. The inner circle represents the product. The next circle includes the product circle as well as the system that incorporates different operators (normal, operational and maintenance). Next is the containing system that adds to the previous view the human beneficiaries even those who are not part of the operations. The wider environment circle adds to the previous stakeholders who could represent negative and hostile roles, political and financial roles, developer, consultant and supplier roles.

In this thesis, we used this taxonomy because it provides a wider and comprehensive view of stakeholders.

A.3 Requirements Reuse, Patterns and Templates

Until the 1990s, software systems were developed from scratch and the concept of reuse was not common [41]. Creating software without taking advantage of the available knowledge can be a waste of efforts [36, 41]. Reusing different components in software development projects allows faster software development and better quality to be achieved because the components are already being tested and have been proved to be useful [41].

Requirements reuse is facilitated by requirements patterns [36]. Patterns were defined by Darimont et al. as reusable models that raises efficency in RE [305]. A requirements pattern is used as guidance in specifying a requiremnt [33]. A requirements pattern presents a model for logical grouping of requirements that can be adaptad to fit different software systems [36]. Domain independent patterns (i.e., grouping of requirements that are not related to the core notions of a specific application domain, such as patterns related to security, transaction handling, etc.) are more abstract than the domain specific patterns (e.g., groupings of requirements that detail needs and wishes specific to a single application domain, such as work process of the train signalling system, or workings of the student assignment marking, etc.) to allow reusability accross domains [36] and they are instantiated (mapped into detailed requirements [37]) when being reused [305].

In addition, requirements patterns facilitate comparisons between requirements of the same type among different software as well as enhancing requirements readability [33]. Furthermore, requirements patterns make writing requirements and spotting missing requirements (by comparing against the pattern) easier [33].

Requirements patterns are presented using templates to ensure consistency [36, 41] and reuseability [36]. Requirements templates are used to represent requirements in a structured textual format [41] as well as other possible forms such as modelling notations [34]. To show alternatives in decisions and actions in requirements, tables can be used with the templates [41].

Using templates helps organisations to reduce efforts spent on discovering, organising and communicating requirements [36]. According to Smith and Lai [306], a requirements template presents a reference model that helps in pointing the required information as well as proposes the structure of representing the requirements [306]. Templates also assist in generating adequate requirements documentations [306].

In a survey conducted by Palomares, Franch and Quer, software practitioners and academics reported that requirements reuse is achieved by filling pre-difiend requirements templates and using a requirements patterns catalogue [307]. Although less widely used then such reuse techniques as copy/paste [307], patterns provide a clear path for requirements re-use. It is also reported that using patterns helps to improve requirements uniformity as well as reducing the time spent on requirements elicitation [307].

A.4 Pattern Development

Based on [32–35, 37] pattern development is commonly acknowledged to comprise the following steps:

A Documents are analysed to identify the recurring problem or challenge in the area/domain. They can be from individual's own experiences, software artefacts, software specification, articles, other practitioners experiences, standards etc.

B Analysing the documentations leads to building a consistent and comprehensive understanding of the problem and the solutions to it.

C Pattern representation involves organising and documenting the pattern to allow it to be re-used. This is done by templates.

D Pattern review is a process of testing the clarity and usablity of the pattern. This can be achieved by collecting feedback from experts and users such as software designers, engineers and other pattern writers. The feedback will lead to improvements.

E Pattern application is the step of applying a pattern to solve a problem.

According to Withall [33], requirements patterns can be used when defining requirements. Indications of what should be said, how and what issues should be considered are supplied by templates [33]. In addition, patterns can be used after the requirements being identified and written, in the process of reviewing the quality of the requirements, during requirement implementation to provide an extensive understanding of a requirement's intention and during requirements testing [33].

A.5 Requirements Negotiation

Requirements negotiation is an essential step in RE due to the different stakeholders (with different goals, needs and requirements) involved in a software system that potentially causes conflict [308]. There are a number of techniques that support negotiation such as Game theory [294], [293], and WinWin [204].

Game theory models includes player as decision makers [294]. Each player has a set of strategies [294]. Choosing a strategy results in set of outcomes [294]. With each outcome, payoffs are presented to a player [294]. A player uses his knowledge about other players payoffs to decide which strategy to take [294].

Easterbrook presented a negotiation model that is built on viewpoints [293]. Viewpoints represents conflicts as perspectives originating from person(s) [293]. The model start with conflict exploration to understand the conflict and creates a map of the conflicts [293]. Then, a generative stage takes place where possible resolutions are investigated [293]. The final stage is the evaluation [293]. During this stage, possible resolutions are related to the conflict map and to each other [293]. This will allow choosing the best resolution [293].

The WinWin requirements negotiation approach involves stakeholders in negotiating to reach a consensus on a “mutually satisfactory or winwin set of requirements” [204]. This assures to stakeholders that they will get what they have agreed to rather than them all getting what they want [204]. The WinWin approach follows a collaborative resolution strategy in which the concerns of all stakeholders are to be satisfied by finding alternatives to conflicting issues [308].

The WinWin approach has four artefacts [205]. First is the Win condition in which stakeholders identify their goals and concerns about the new software. If a condition does not conflict with other stakeholders' conditions, then it will be considered as an agreement. If it does conflict with others, it will be documented as an issue. Options artefact documents alternatives provided by stakeholders to resolve issues. Agreements record agreed on options and win conditions. In addition, requirements workshops, group media and groupware tools are also suggested as methods for negotiating requirements [309]. Within the negotiation phase, requirements priorities need to be discovered using, for example, sorting, voting, market surveys and matrix techniques [309].

A.6 Sustainability and Software

Sustainable development is defined by the report of the World Commission on Environment and Development [310] as “ development that meets the needs of the present without compromising the ability of future generations to meet their own needs". The five dimensions of sustainability in relation to software are as follows [9, 311]:

- **Individual** sustainability is related to preserving human capital via for example education, health, skills etc [311]. It involves individual freedom, human dignity, fulfilment, ability to thrive,ability to exercise rights, and to develop freely [9].
- **Social** sustainability is related to maintaining good relationships between individuals in society to maintain the community's solidarity and services.
- **Economic** sustainability is related to preserving capital and added value.
- **Environmental** sustainability is related to preserving natural resources through, for example, waste management and energy consumption.
- **Technical** sustainability is concerned with maintaining and evolving "artificial systems (such as software) over time" [9].

Penzenstadler et al. remarked that "sustainable software is energy-efficient, minimi[s]es the environmental impact of the processes it supports, and has a positive impact on social and/or economic sustainability" [312]. Calero and Piattini [313] defined sustainable software as:

> software whose direct and indirect negative impacts on economy, society, human beings, and the environment resulting from development, deployment, and usage of the software is minimal and/or has a positive effect on sustainable development.

De Souza and colleagues defines a sustainable software as the software that is used now and in the future with reusable components [314]. In additon, they noted that creating a community of users and developers ensures the software continuity [314].

Assessing the effects of ICT on sustainability is achieved by assessing three levels of effects [315]. First order effects relate to the opportunities resulting from the existence of ICT and the process of making it [315, 316]. Second order effects are related to the impacts and opportunities of the ongoing use of ICT such as changes in the way of doing things "and what were capable of "[315–317]. Third order effects are related to the opportunities or impacts of the large and ongoing application (medium-and long-term usage) of the outcomes [315, 316] such as "energy demand, mass surveillance, etc." [317]. According to the Karlskrona Sustainability Manifesto [317], ICT impacts and opportunities are to be evaluated not only by focusing on the direct effects (first order) but also the indirect (second order) and systemic (third order) effects.

Mahaux, Heymans and Saval remarked on the need to support the process of sustainability RE [67]. Penzenstadler et al. noted the need for 'domain-independent' guidelines to provide support to sustainability within the software engineering domain [318].

In a recent survey-based study by Condori-Fernandez, and Lago [319], the contribution that software quality (e.g. usability, security, maintainability, etc.) requirements makes to sustainability was investigated. The software quality requirements were based on ISO/IEC 25010 [42]. Software and ICT practitioners with sustainability expertise responded to the online survey. As part of the survey, social sustainability was defined as the "ability to allow current and future generations to have equal and equitable access to the social resources in a way that preserve[s] their socio-cultural characteristics and achieve healthy and modern society" [320]. The study investigated 8 quality requirements and their relation to social sustainability. The quality requirements are: 1) freedom from risk, 2) effectiveness (accuracy and completeness with which users achieve specified goals), 3) efficiency, 4) satisfaction, 5) security, 6) usability, 7) accessibility and 8) compatibility. The study results suggests that usability, accessibility and compatibility contributes to social sustainability.

Johann and Maalej [321] remarked on the importance of including social sustainability (as defined in Chapter 1) in requirement engineering "as a basic non-functional requirement". The paper discussed examples of the social impacts of software such as access to data and civic participation in the requirement engineering life cycle which is known as social software engineering. Another example of the social impacts of software is accessibility. The paper mentioned software that allows users in rural areas with limited Internet access to receive market information via email and SMS. The authors noted the need for clear characteristics that help to create a uniform model of a social sustainable software.

Mahaux in [322] discussed how participation in requirements engineering supports social sustainability by referring to studies from other disciplines. This is achieved through empowering participants with distinct interests and allowing them to be part of the requirements engineering process. Empowering is achieved by successful participation in decision-making that, in turn, leads to project acceptance and longevity. In successful participation, the collaborative environment has to be configured in a way that mitigates the risk of a ruling strong opinions. In addition, successful participation requires involving non-experts in discussions without ignoring the important roles to be played by experts in maintaining the quality of the resulting work from the discussion. Experts act as facilitators, consultants and validators. The author also argued that involving the right stakeholders (which is a common practice in RE) who care about sustainability concerns supports sustainable software engineering.

Cabot et. al [68] proposed a modelling and integration method of environmental sustainability in software and business decision-making. The method employed requirements modelling using the goal-oriented techniques (GORE) i.e. i*. In this method, sustainability is treated as a soft-goal that is supported by "domain-independent" sub-goals of reduce, reuse and recycle. In turn, those sub-goals can be further decomposed into sub-goals. In the same spirit, we look for the patterns that relate to the key concerns of social sustainability, without immediate focus on specific software projects or domains.

Appendix B

Systematic Literature Review Sources

1. Knowledge Representation in Virtual Teams: a perspective approach for Synthetic worlds. By Fiumara G., Maggiorini D., Provetti A., Ripamonti L.

2. Nations in social LCA. By Ekvall T.

3. Social life cycle assessment of palm oil biodiesel: a case study in Jambi Province of Indonesia. By Manik Y., Leahy J., Halog A.

4. Sustainability of European maize-based cropping systems: Economic, environmental and social assessment of current and proposed innovative IPM-based systems. By Vasileiadis V.P., Moonen A.C., Sattin M., Otto S., Pons X., Kudsk P., Veres A., Dorner Z., van der Weide R., Marraccini E., Pelzer E., Angevin F., Kiss J.

5. Accounting for the social dimension of sustainability: Experiences from the biotechnology industry. By von Geibler J., Liedtke C., Wallbaum H., Schaller S.

6. Comparative evaluation of rice production systems for their sustainability. By Chandre Gowda M.J., Jayaramaiah K.M.

7. Ergonomics and sustainability - Challenges from global supply chains Hasle P., Jensen P.L.

8. Assessing the agricultural sustainability of conventional farming systems in Samsun province of Turkey. By Ceyhan V.

9. A bio-economic indicators suite for the appraisal of the demersal trawl fishery in the Southern Adriatic Sea (Central Mediterranean) Ceriola L., Accadia P., Mannini P., Massa F., Milone N., Ungaro N.

10. A concept for assessing sustainability impacts of forestry-wood chains. By Paivinen R., Lindner M., Rosen K., Lexer M.J.

11. Social sustainability, flexible work arrangements, and diverse women. By Blake-Beard S., O'Neill R., Ingols C., Shapiro M.

12. Reverse logistics and social sustainability. By Sarkis J., Helms M.M., Hervani A.A.

13. Exploring Social Sustainability: Learning from Perspectives on Urban Development and Companies and Products. By Weingaertner C., Moberg A.

14. Development of fishery indicators for local management initiatives - A case study for Plettenberg Bay, South Africa. By Smith M.K.S., King C.M., Sauer W.H.H., Cowley P.D.

15. Evaluating social sustainability in historic urban environments. By Landorf C.

16. Environmental and social impact considerations for sustainable project life cycle management in the process industry. By Labuschagne C., Brent A.C., Claasen S.J.

17. Evaluation of the urban renewal projects in social dimensions . By Lee G.K.L., Chan E.H.W.

18. Rainforest Alliance Certification of Kenyan tea farms: A contribution to sustainability or tokenism? By Ochieng B.O., Hughey K.F.D., Bigsby H.

19. Measuring Social Sustainability: A Community-Centred Approach. By Magee L., Scerri A., James P.

20. Sustainability indicators for the assessment of nuclear power. By Stamford L., Azapagic A.

21. Sustainability of forestry and timber industry in Ghana. By Eshun J.F., Potting J., Leemans R.

22. Sustainable homeservices? Toward household services that enhance ecological, social and economic sustainability. By Halme M., Jasch C., Scharp M.

23. Social Capital and Walkability as Social Aspects of Sustainability. By Rogers S.H., Gardner K.H., Carlson C.H.

24. Towards a global criteria based framework for the sustainability assessment of bioethanol supply chains: Application to the Swiss dilemma: Is local produced bioethanol more sustainable than bioethanol imported from Brazil? By Corbire-Nicollier T., Blanc I., Erkman S.

25. Towards a new welfare state: The social sustainability principle and health care strategies. By Garces J., Rodenas F., Sanjose V.

26. Urban form and social sustainability: The role of density and housing type. By Bramley G., Power S.

27. Social indicators for sustainable project and technology life cycle management in the process industry. By Labuschagne C., Brent A.C.

28. Social sustainability and social acceptance in technology assessment: A case study of energy technologies. By Assefa G., Frostell B.

29. Adaptation of sustainability community indicators for Malaysian campuses as small cities. By Saadatian O., Sopian K.B., Salleh E.

30. Alternative agricultural policy scenarios, sector modelling and indicators: A sustainability assessment. By Lehtonen H., Aakkula J., Rikkonen P.

31. Changes in rural-urban sex ratio differences in the young professional age group as an indicator of social sustainability in rural areas: A case study of continental Spain, 2000-2010. By Gurrutxaga M.

32. Connecting Municipal and Regional Level Planning: Analysis and Visualization of Sustainability Indicators in Bergslagen, Sweden. By Andersson K., Angelstam P., Axelsson R., Elbakidze M., Tornblom J.

33. Developing statewide indices of environmental, economic, and social sustainability: A look at Oregon and the Oregon Benchmarks. By Schlossberg M., Zimmerman A.

34. Effects of CAP policy on farm household behaviour and social sustainability. By Manos B., Bournaris T., Chatzinikolaou P., Berbel J., Nikolov D.

35. An appraisal of social aspects in project and technology life cycle management in the process industry. By Brent A.C., Labuschagne C.

36. An exploration of measures of social sustainability and their application to supply chain decisions. By Hutchins M.J., Sutherland J.W.

37. From brown to green? Assessing social vulnerability to environmental gentrification in New York City. By Pearsall H.

38. Characterization of sustainable development indicators for various power generation technologies. By Genoud S., Lesourd J.-B.

39. Assessing stakeholders' influence on social performance of infrastructure projects. By Doloi H.

40. Analyzing scale independence in jobs-housing and commute efficiency metrics. By Niedzielski M.A., Horner M.W., Xiao N.

41. Assessing the sustainability of the US food system: A life cycle perspective. By Heller M.C., Keoleian G.A.

42. Agroforestry as a sustainable land use option in degraded tropical forests: A study from Bangladesh. By Alam M., Furukawa Y., Harada K.

43. Assessment of ecological sustainability of a building subjected to potential seismic events during its lifetime. By Menna C., Asprone D., Jalayer F., Prota A., Manfredi G.

44. Small scale fisheries in Europe: A comparative analysis based on a selection of case studies. By Guyader O., Berthou P., Koutsikopoulos C., Alban F., Demaneche S., Gaspar M.B., Eschbaum R., Fahy E., Tully O., Reynal L., Curtil O., Frangoudes K., Maynou F.

45. MARGOV - building social sustainability. By Lia Vasconcelos, Ursula Caser-Maria Joao Ramos Pereira, Grasa Gonalves, Rita S.

46. More Than Wires, Pipes and Ducts: Some Lessons from Grassroots Networked Communities and Master-Planned Neighbourhoods. By Mark Gaved, Marcus Foth.

47. Decision making for social sustainability: A life-cycle assessment approach. By Hutchins, M.J.; Gierke, J.S.; Sutherland, J.W.

48. Developing a social capital metric for use in an educational computer game. By Gennett, Z.A.; Isaacs, J.A.; Seager, T.P.

49. Community Resilience: An Indicator of Social Sustainability. By Magis, Kristen

50. Impact assessment of CAP policies on social sustainability in rural areas: an application in Northern Greece. By Basil Manos, Thomas Bournaris, Parthena Chatzinikolaou.

51. Social Sustainability in Selecting Emerging Economy Suppliers. By Ehrgott M., Reimann F., Kaufmann L., Craig R. Carter.

52. Social sustainability in certification schemes for biofuel production: an explorative analysis against the background of land use constraints in Brazil. By Anna Mohr, Linda Bausch.

53. Critical factors for improving social sustainability of urban renewal projects. By Edwin Chan, Grace K. L. Lee

54. Cloud Management: Challenges and Opportunities. By Forell, T.; Milojicic, D.; Talwar, V.

55. Achieving sustainability through a combination of LCA and DES integrated in a simulation software for production processes. By Widok, A.H.; Schiemann, L.; Jahr, P.; Wohlgemuth, V.

56. An investigation of indicators for measuring sustainable manufacturing . By Chengcheng Fan; Carrell, J.D.; Hong-Chao Zhang

57. The use of multi-actor systems for studying social sustainability: Theoretical backgrounds and pseudo-specifications. By Faber, N.R.; Jorna, R.J.

58. Social Sustainability Considerations during Planning and Design: Framework of Processes for Construction Projects. By Valdes-Vasquez, Rodolfo; Klotz, Leidy E.

59. Exploring sustainability research in computing: where we are and where we go next. By Bran Knowles, Lynne Blair, Mike Hazas, Stuart Walker

60. A Generic Model for Sustainability with Process- and Product-specific Instances. By Penzenstadler B., Femmer H.

61. Human numbers, environment, sustainability, and health. By McMichael AJ, Powles JW.

62. Novel methods to assess environmental, economic, and social sustainability of main agricultural regions in China. By Liu, Fang; Zhang, Hongqi

63. General sustainability indicator of renewable energy system based on grey relational analysis. By Liu, Gang; Baniyounes, Ali M.; Rasul, M. G.; Amanullah, M. T. O.; Khan, M. M. K.

64. Sustainability in hydropower development-A case study. By Liu, Jian; Zuo, Jian; Sun, Zhiyu; Zillante, George; Chen, Xianming

65. Autonomy and sustainability: An integrated analysis of the development of new approaches to agrosystem management in family-based farming in Carnaubais Territory, Piaui, Brazil. By Oliveira, Francisco das Chagas; Calle Collado, Angel; Carvalho Leite, Luiz Fernando

66. Towards a triple bottom-line sustainability assessment of the U.S. construction industry. By Murat Kucukvar, Omer Tatari

67. Introduction: Challenges in Adult and Vocational Education for Social Sustainability . By Peter Willis, Stephen McKenzie, Roger Harris.

68. Embedding sustainability in capstone engineering design projects. By Hasna, A.M.

69. MOTIFS: a monitoring tool for integrated farm sustainability. By Marijke Meul, Steven Van Passel, Frank Nevens, Joost Dessein, Elke Rogge, Annelies Mulier, Annelies Van Hauwermeiren

70. A participative procedure to select indicators of policies for sustainable urban mobility. Outcomes of a national test. By Gerardo Marletto, Francesca Mameli

71. An interdisciplinary project in sustainable development based on modern visual programming environments and web 2.0 technologies. By Giordano, D.; Maiorana, F.

72. Assessing Green IT Initiatives Using the Balanced Scorecard. By Jain, R.P.; Benbunan-Fich, R.; Mohan, K.

73. Combining sustainability criteria with discrete event simulation. By Widok, A.H.; Wohlgemuth, V.; Page, B.

74. Identifying and ranking attributes that determine sustainability in Dutch dairy farming. By Klaas J. Van Calker, Paul B. M. Berentsen, Gerard W. J. Giesen, Ruud B. M. Huirne

75. Development and application of a multi-attribute sustainability function for Dutch dairy farming systems. By van Calker, KJ; Berentsen, PBM; Romero, C; Giesen, GWJ; Huirne, RBM

76. Social and Cultural Sustainability: Criteria, Indicators, Verifier Variables for Measurement and Maps for Visualization to Support Planning. By Robert Axelsson, Per Angelstam, Erik Degerman, Sara Teitelbaum, Kjell Andersson, Marine Elbakidze, Marcus K. Drotz

77. Sustainability indicators for livestock farming. A review. By Lebacq, T., Baret, P. V., Stilmant, D.

78. Adult and Vocational Education for Social Sustainability: A New Concept for TVET for Sustainable Development. By McKenzie, S.

79. From "Localized" to "Networked": A transformation in community structures, By Towne, W.

80. Urban Social Sustainability: A Study of the Emirati Local Communities in Al Ain. By Ahmed K.G.

81. Evaluation of the effect of recycling on sustainability of municipal solid waste management in Thailand. By Menikpura S.N.M., Gheewala S.H., Bonnet S., Chiemchaisri C.

82. Employment and Environment in a Sustainable Europe. By Hinterberger F., Omann, I., Stocker A.

83. A PCA-based method for construction of composite sustainability indicators. By Tao Li, Hongchao Zhang, Chris Yuan, Zhichao Liu, Chengcheng Fan.

84. Measuring the Social Sustainability of Urban Communities: The Role of Local Authorities. By Neamţu, B.

85. Structural analysis of factors affecting agricultural sustainability in Qazvin Province, Iran. By Asadi A., Kalantari K., Choobchian S.

86. Towards assessing the sustainability of European logging operations. By Berg S., Fischbach J., Bruchert F., Poissonnet M., Pizzirani S., Varet A., Sauter U.H.

87. The Virtual Workplace: A new alternative merging technology and policy for more environmental, economic and social sustainability. By El Dessouky N.F.

88. The impact of lean manufacturing on environmental and social sustainability: A study using a concept mapping approach. By Silva C., Vaz P., Ferreira L.M.

89. Evaluation of sahand new town based on sustainability indicators. By Asli M.F., Khojasteh Ghamari M.A.

90. Effects of pervasive computing on sustainable development. By Koehler, A.; Som, C.

91. Finding the Right Balance Between Standardisation and Flexibility: A Compendium of Indicators for Measuring Child Well-Being. By Carboni I., Morrow N.

92. Sustainability impact assessment on the production and use of different wood and fossil fuels employed for energy production in North Karelia, Finland. By den Herder M., Kolstrom M., Lindner M., Suominen T., Tuomasjukka D., Pekkanen M.

93. European Union agricultural policy scenarios' impacts on social sustainability of agricultural holdings. By Bournaris T., Manos B.

94. Comparison of fuzzy-based and AHP methods in sustainability evaluation: a case of traffic pollution-reducing policies. By Rossi R., Gastaldi M., Gecchele G.

95. Higher sustainability performance of intensive grazing versus zero-grazing dairy systems. By Meul M., Van Passel S., Fremaut D., Haesaert G.

96. Creating a "values" chain for sustainable development in developing nations: where Maslow meets Porter. By Walsh P.R.
97. The social pillar of sustainable development: A literature review and framework for policy analysis. By Murphy K.
98. Sustainability Analysis of Ecotourism on Yao Noi Island, Thailand. By Jitpakdee R., Thapa G.B.
99. Sustainability assessment: An integrated approach with inhabitant participation. By Mattarozzi S., Antonini E.
100. A Methodology for assessing the sustainability of hydrogen production from solid fuels. By Gnanapragasam N.V., Reddy B.V., Rosen M.A.
101. Never Waste a Good Crisis: Towards Social Sustainable Development. By Bijl R.
102. Measuring sustainable urbanization in China: a case study of the coastal Liaoning area. By Liu S., Zhang P., Jiang X., Lo K.
103. Technology sustainability assessment to support decision making on energy production at local scale. By Sala S., Castellani V.
104. Evaluation of sustainability using fuzzy association rules mining. By Vinodh S. , Selvan K.E. , Prakash N.H.
105. Building communities of place: Habitual movement around significant places. By Hargreaves A.
106. The social in change: property rights contradictions in Finland. By Salmi P.
107. Social life cycle assessment for material selection: a case study of building materials. By Hosseinijou S.A , Mansour S., Shirazi M.A.

108. Sustainability Assessment and Reporting for Nonprot Organizations: Accountability “for the Public Good”. By Jones K.R., Mucha L.

109. Evaluation of sustainable development of cities in Shandong province based on fuzzy BP-ANN & GIS. Shan B., Li-e W.

110. Extending the Technology-Community-Management model to disaster recovery: Assessing vulnerability in rural Asia. By Chib A., Komathi, A.L.E

111. Biometrical Models as Tools for Forest Ecosystem Management. By Pretzsch, H.

112. Impact of telecentres on Sri Lankan society. By Rajapakse J.

113. Measuring the Sustainability Performance of Software Projects. By Albertao, F.; Jing Xiao; Chunhua Tian; Yu Lu; Kun Qiu Zhang; Cheng Liu.

114. Locast Civic Media: Extending Civic Engagement Boundaries through Mobile Media and Hyper-Local Conversations. By Boardman, D.; Casalegno, F.; Pomeroy, S.

115. Role of telecenters in empowering citizens: a case study of ‘Mahiti Mitra’ project in Kutch district of Gujarat, India. By Mishra, G.

116. On the Sustainability Impacts of Cloud-Enabled Cyber Physical Space. By Wlodarczyk T.W., Rong C.

117. Towards a Procedural Model for Sustainable Information Systems Management. By Schmidt N.-H., Erek K., Kolbe L.M., Zarnekow, R.

118. On building sustainable broadband networks in rural areas. By Nungu, A.; Knutsson, B.; Pehrson, B.

119. ToSIA-A tool for sustainability impact assessment of forest-wood-chains. By Lindner M., Suominen T., Palosuo T., Garcia-Gonzalo J., Verweij P., Zudin S., Paivinen R.

120. Assigning results of the Tool for Sustainability Impact Assessment (ToSIA) to products of a forest-wood-chain. By Palosuo T., Suominen T., Werhahn-Mees W., Garcia-Gonzalo J., Lindner M.

121. Comparison of sustainability issues in two sensitive areas of China. By Zhen L., Cao S., Wei Y., Dilly O., Liu X., Li F., Koenig H., Tscherning K., Helming K.

122. Strengths and weaknesses of common sustainability indices for multidimensional systems. By Mayer A.L.

123. Social sustainability and the art of engagement - The small towns: Big picture experience. By Rogers M.

124. Methodologies for assessing performance of irrigation and drainage management. By Bos M.G., Murray-Rust D.H., Merrey D.J., Johnson H.G., Snellen W.B.

125. Ecological and agro-economic study of small farms in sub-Saharan Africa. By Onduru D.D. , Du Preez C.C.

126. MASC, a qualitative multi-attribute decision model for ex ante assessment of the sustainability of cropping systems. By Sadok W., and Angevin F., Bergez J. and Bockstaller C., Colomb B., Guichard L. and Reau R., and Messéan A. and Doré T.

127. A comparison of the sustainability of public and private transportation systems: Study of the Greater Toronto Area. By Kennedy C.A.

128. Supplying urban ecosystem services through multifunctional green infrastructure in the United States. By Lovell S.T., Taylor J.R.

129. Participatory impact assessment for poverty alleviation: Opportunities for communities and development agencies. By Jackson E.T.

130. Education in Post-conflict Environments: Pathways to Sustainable Peace. By Spence R.

131. Assessing farm-level agricultural sustainability over a 60-year period in rural eastern India. By Sharma D., Shardendu S.

132. Industrial ecosystem indicatorsdirect and indirect effects of integrated waste- and by-product management and energy production. By Korhonen L., Okkonen L., Niutanen V.

133. The Analytic Hierarchy Process (AHP) Approach for Assessment of Urban Renewal Proposals. By Lee G.K.L., Chan E.H.W.

134. Adaptive Ecosocial System Sustainability Enhancement in Sub-Saharan Africa. By Gilioli G., Baumgärtner J.

135. Environmental impact assessment of an automotive component using eco-indicator and CML methodologies. By Vinodh S., Jayakrishna K., Joy D.

136. Is There Room at the Bottom for CSR? Corporate Social Responsibility and Nanotechnology in the UK. By Groves C., Frater L., Lee R., Stokes E.

137. The Emotional City: Refugee Settlement and Neoliberal Urbanism in Calgary. By Burke Wood P., McGrath S., Young J.

138. Sustainability Assessment of Future Scenarios: Methodology and Application to Mountain Areas of Europe. By Sheate W.R., Do Partidário M.R., Byron H., Bina O., Dagg S.

139. Regulating sustainability in the coffee sector: A comparative analysis of third-party environmental and social certification initiatives. By Raynolds L.T., Murray D., Heller A.

140. Life Cycle Inventory of Particleboard: A Case Study in the Wood Sector. By Rivela B., Hospido A., Moreira T., Feijoo G.

141. Agroforestry as a sustainable landuse option in degraded tropical forests: a study from Bangladesh. By Alam M, Furukawa Y., Harada K.

142. Moving towards urban sustainability in Kenya: a framework for integration of environmental, economic, social and governance dimensions. By Mutisya E., Yarime M.

143. The disappearing sustainability triangle: community level considerations. By Chang C.T.

144. Renewable energy for sustainable agriculture. By Chel A., Kaushik G.

145. Integrating Social Development and Financial Sustainability: The Challenges of Rural Computer Kiosks in Kerala. By Kuriyan R., Toyama K., Ray I.

146. Semantic Knowledge Management to Support Sustainable Product Design. By Yang Q.Z., Song B.

147. Managing Sustainability with the Support of Business Intelligence Methods and Tools. By Petrini M., Pozzebon M.

148. Workgroup issue paper: Indicators and assessment of agricultural sustainability. By Huffman E.

149. Reframing social sustainability reporting: Towards an engaged approach. By Magee L., Scerri A., James P., Thom J.A., Padgham L., Hickmott S., Deng H., Cahill F.

150. Public-private partnerships and information technologies for development in India. By Kuriyan R., Ray I.

151. Disruptive innovation in China's BoP market. By Zhou J, Tong Y., Li J.

152. Who is the advocate? Stakeholders for sustainability Penzenstadler, B., Femmer, H., Richardson, D.

153. Sustainability requirement patterns. By Roher K., Richardson D.

154. The milk run revisited: A load factor paradox with economic and environmental implications for urban freight transport. By Arvidsson N.

155. Economic, social, and environmental sustainability in development theory and urban planning practice. By Basiago A.D.

Appendix C

Expert's Survey

C.1 Survey Form

C.1.1 Page 1: Introduction

A smart garden application is designed with the goals of preserving water resources consumed in home gardens. The software is to be connected to growing beds. "During their growth period, vegetables receive much more water than needed and this goes unquestioned because it does not harm the plants. However, using the sprinklers versus manually watering a garden consumes water at a much greater and unnecessarily faster rate." The application is needed to "find the balance between water consumed while keeping the healthy nature of the plant. Smart garden application will use sensors that are connected to a board "to measure the water levels in growing beds and ultimately find a solution to reduce the amount of water consumption while maintaining a nourished garden. Gardeners are expected to control irrigation through using the Smart garden application. Source

(Collin, Kaitryn, Nancy, Nathan, Ryan, Resilient Smart Gardens, online: `http://birgit.penzenstadler.de/teach/493a.html`)

Definitions:

- Equality in software is related to treating stakeholders with variability (disability, language, gender, technology, etc.) equally to the other stakeholders. In addition, it is related to fairly considering and selecting (direct, indirect) stakeholders' goals to be implemented in a software. We also relate equality to the equal access to software services/functions and facilities that will assist stakeholders in achieving their goals.
- Relevant to equality requirements represents a need that if satisfied, stakeholders would feel being equally considered, benefited and served by a software application.
- SG refer to Smart Garden application

The following page lists equality requirements identified for the SG application.

C.1.2 Page 2: Equality Requirements

R1. The SG should provide plant nursery with seasonal plantation history that will be used to estimate seasonal demand.
R2. The SG should allow the gardener to create garden profile (Plants type, number, location) to manage their garden and reduce gardening efforts.
R3. The SG should allow the gardener to register plant (sensor, sprinkler, ideal moisture level) to manage their garden and reduce gardening efforts.
R4. The SG should allow the gardener to water plants to manage their garden and reduce gardening efforts.

R5. The SG should allow the gardener to set watering alarms to manage their garden and reduce gardening efforts.

R6. The SG should allow the gardener to cancel watering alarms to manage their garden and reduce gardening efforts.

R7. The SG should allow the gardener to Set auto watering function (when, amount, plant) to manage their garden and reduce gardening efforts.

R8. The SG should allow the gardener to cancel auto watering function to manage their garden and reduce gardening efforts.

R9. The SG should allow the gardener to view garden history to manage their garden and reduce gardening efforts.

R10. The SG should allow the gardener to test sensors and sprinkles to manage their garden and reduce gardening efforts.

R11. The SG should allow the gardener to view environmental advice to manage their garden and reduce gardening efforts.

R12. The SG should allow the gardener to check soil moisture to assess plant need.

R13. The SG should allow the gardener to compare current soil condition to ideal condition and take action (water, not to water).

R14. The SG should allow the gardener to select what garden information to be public to be part of shared gardening knowledge.

R15. The SG should allow the gardener to view similar gardens report (what plants, water consumption, etc.) as part of shared gardening knowledge.

R16. The SG should allow the gardener to post garden enquiry as part of shared gardening knowledge.

R17. The SG should allow the gardener to reply to garden enquiry as part of shared gardening knowledge.

R18. The SG should allow the gardener to select what garden information to be private to protect gardener's privacy.

R19. The SG should allow potential gardeners (current and in future) to view neighbourhood's gardens history to gain knowledge.

R20. The SG should allow potential gardeners (future) to view neighbourhood's gardens trends to gain knowledge.

R21. The SG should allow gardeners with different languages to select preferred language to customize the application.

R22. The SG should provide elder gardeners with help function and tutorial to achieve an easy to use application.

R23. The SG should allow elder gardeners to choose a preferred font size to be able to customize the application display that is suitable for them.

R24. The SG should allow gardeners with different mobile devices (iOS, Android, etc.) to use the application (compatible versions).

R25. The SG should allow elder gardeners to view help instructions "how to install and fix" to achieve an easy to use application.

R26. The "how to install and fix" information should be displayed as textual, audio and video instructions to achieve an easy to use application.

R27. The SG should provide the current equipment suppliers with gardens and equipment information to be able to estimate future demand.

R28. The SG should allow potential equipment suppliers to view gardens history to be able to prepare market feasibility report.

R29. The SG should allow pesticide control authority to view garden-plant information (per area) to be able to plan public pesticide spraying process.

R30. The SG should allow plantation authority to view garden-plant information to be able to manage plantation.

R31. The SG should allow plantation authority to prepare reports on plants to be removed/increased per garden or area to be able to manage plantation.

R32. The SG should allow plantation authority to view garden-plant information to ensure plantation standards are followed.

R33. The SG should allow water and irrigation authority to view water consumption reports to be able to monitor consumption.

R34. The SG should allow water and irrigation authority to send overwatering note to gardeners as part of monitoring water consumption.

R35. The SG should allow environmental authority/charities to view water consumption reports to be able to monitor consumption.

R36. The SG should allow environmental authority/charities to post environmental advice for home gardens to be able to share environmental best practices.

R37. The SG should allow environmental authority/charities to reply to garden enquiry with environmental concern as part of sharing environmental best practices.

For statements 1-37, the responses were elicited via the following:

A In your opinion, how Relevant is the statement to equality?

○ Directly and highly relevant ○ Definitely relevant ○ Probably relevant
○ Possibly relevant - likely of indirect relevance ○ Possibly not relevant
○ Probably not relevant ○ Definitely not relevant

B Why?

Missing Equality requirements

Do you think there are any missing requirements that support equality among software stakeholders? Kindly explain.

Appendix D

Equality and Software Survey

D.1 Invitation Letter

Dear [friends/colleagues/...etc]

Do you think equality should be supported through software applications? Please spend 15 min. of your time to fill in this online survey

`https://leicester.onlinesurveys.ac.uk/equality-and-software-_survey`

to help us learn what is relevant for equality in software applications.
Please do forward this link to anyone else who, you think, may be willing to participate in this survey.
Kind Regards,
Maryam Al Hinai, Informatics PhD researcher at the University of Leicester, UK

D.2 Survey Form

D.2.1 Page 1: Information sheet

Research title: Social Sustainability and Software: Case of Equality.

Researcher: Maryam Ali Al Hinai

Department: Informatics, University of Leicester, UK

Introduction:

The only pre-requisite for participating in this survey is familiarity (e.g., due to use) with such applications as online shopping, social media applications (e.g. Facebook, Twitter, Instagram, Snapchat, LinkedIn, etc.), online banking, e-learning, or similar.

This survey aims to help requirements engineers to identify Equality requirements for such software systems.

The survey will take 15 minutes to be completed.

Your confidentiality is assured. The provided answers will be used purely for this PhD research only. No personally identifiable data will be recorded.

Please confirm that you volunteer to participate by selecting Agree option in the consent form below.

For further enquiries, please contact:

Maryam Ali Al Hinai, Department of Informatics, University of Leicester, Leicester, LE1 7RH, UK

Email 1: masah1@le.ac.uk

Email 2: rc256@leicester.ac.uk

Date: 01/10/2016

Consent

We appreciate your decision to participate in this questionnaire.

We assure you that the information you contribute to this questionnaire is for research purposes only and will not be shared with 3rd parties.
You can keep a copy of this consent form (print or screen shoot) for your records. By selecting the **Agree** option, you confirm that you have read the information sheet (above) and you voluntarily agree to participate. You also confirm that you are of age 18 years or above. * Required

- Agree
- Disagree

D.2.2 Page 2: Equality and software

Definition

- Equality is related to providing all individuals in the society with equal life opportunities without discrimination because of their origin, beliefs, position, or (dis-) abilities [Equality Human Rights Commission, GB]. To support equality through software, we believe that the goals of all directly or indirectly affected stakeholders of a given software system should be fairly considered when implementing that software. Furthermore, equal access to software services necessary to realise the fairly selected goals needs to be provided.
- Direct stakeholders are stakeholders who will directly be affected (positively or negatively) by the software direct use such as users.
- Indirect stakeholders are those who will indirectly be benefited or harmed by the software.

Q1: In your view, what is the **importance** of the following software features in **supporting equality**?

Statement	Not at all important to software equality	Neither important nor unimportant	Slightly important to software equality	Important to software equality	Very important to software equality
User authentication is important for supporting equality	☐	☐	☐	☐	☐
Usability of software to users from different age ranges is important to support equality	☐	☐	☐	☐	☐
Suitability of software to users from different age ranges is important to support equality	☐	☐	☐	☐	☐
Short response time to user enquiry is important for supporting equality	☐	☐	☐	☐	☐
Short recovery time after system failure is important for supporting equality	☐	☐	☐	☐	☐
Suitability of software to users from different genders is important to support equality	☐	☐	☐	☐	☐
Considering direct stakeholders' goals behind using a software is important to support equality	☐	☐	☐	☐	☐
Fairly selecting which goals will be implemented in the software is important for supporting equality	☐	☐	☐	☐	☐
Ability to accommodate new types of users is important for supporting equality	☐	☐	☐	☐	☐
Multilingual interface is important for supporting equality	☐	☐	☐	☐	☐
Different information presentation formats (e.g., audio, video, text) is important for supporting equality	☐	☐	☐	☐	☐
Compatibility of software application with different operating systems is important for supporting equality	☐	☐	☐	☐	☐
Compatibility of software application with different hardware devices is important for supporting equality.	☐	☐	☐	☐	☐
Availability of software's usage guidance (e.g., help, tutorials, and tips) considering users with no/little prior knowledge of this software is important for supporting equality	☐	☐	☐	☐	☐
Availability of software's shortcuts to accomplish tasks for experts and fast learners is important for supporting equality	☐	☐	☐	☐	☐
Availability of software application on different web and mobile platforms is important for supporting equality	☐	☐	☐	☐	☐
Availability for use 24 hours per day, 365 days per year is important for supporting equality	☐	☐	☐	☐	☐
Allowing stakeholders to equally access software services to achieve their goals is important for supporting equality	☐	☐	☐	☐	☐
Suitability of software for users from different religious beliefs is important for supporting equality	☐	☐	☐	☐	☐
Accepting information from different media (e.g., voice, text, braille) is important for supporting equality	☐	☐	☐	☐	☐
Considering indirect stakeholder goals that are affected by the software is important for supporting equality	☐	☐	☐	☐	☐

Q2: To support equality in a given software application, we should implement the goals that

- Deliver the highest profit
- Provide best usable interface
- Provide same level of functionality to all user groups
- Implement the most prioritized goals for each user group

Q3: An online shopping software has several groups of users. Normal users are regular users who use the software to perform basic shopping through the website. Gold users are those who buy expensive products that generate high income for the business. Special users are those who use the application with special request of adjustment to their disabilities (colour blindness, hearing loss, etc.). Each group of users have different goals to be implemented by the online shopping application. To support equality through the online shopping application, which goals will you select to be implemented in the application.

- The normal users' goals
- The gold users' goals
- The special users' goals
- None of them
- The normal and special users' goals
- The normal and gold users' goals
- The special and gold users' goals
- All of them

D.2.3 Page 3: Background information

Q4: What is your gender?

- Male
- Female

Q5: What is your age?

- 18 to 24 years
- 25 to 34 years
- 35 to 44 years
- 45 to 54 years
- 55 to 64 years
- Age 65 or older

Q6: What is your religion?

- Christianity
- Islam
- Judaism
- Hinduism
- Buddhism
- Other

- Prefer Not to Say

Q7: What is your highest level of education?

- Unschooled
- High school degree or less
- A college degree (diploma and equivalent)
- Undergraduate (BSC and equivalent)
- Masters Degree (MSc or equivalent)
- PhD (or equivalent)

Q8: What is your current employment status?

- Employed
- Student
- Retired
- Unemployed

Q9: How do you rate your level of software use proficiency (e.g., for online shopping, social media use, internet browsing, etc.)?

- Novice (beginner: need frequent help to use most software applications)
- Intermediate (may need help occasionally with some)
- Advanced (need little or no help with most)
- Expert (provide help to others)

D.2.4 Page 4: End

Thank you for your participation.

D.3 Data Coding

To analyse the survey responses, data has been coded as follows:

- Equality statements (Q1):

  Not at all important to software equality (NAI) = 1, Neither important nor unimportant NINU = 2, Slightly important to software equality (SI) = 3, Important to software equality (IE) = 4, Very important to software equality (VIE) = 5.

- Golas (Q2): Deliver the highest profit = 1, Provide best usable interface = 2, Provide same level of functionality to all user groups = 3, Implement the most prioritized goals for each user group = 4.

- Goals (Q3): The normal users goals = 1, The gold users goals = 2, The special users goals = 3, None of them = 4, The normal and special users goals = 5, The normal and gold users goals = 6, The special and gold users goals = 7.

- Gender (Q4): Female =1, Male = 2.

- Age (Q5): 18 to 24 years = 1, 25 to 34 years = 2, 35 to 44 years = 3, 45 to 54 years = 4, 55 to 64 years = 5, Age 65 or older = 6.

- Religion (Q6): Christianity = 1, Islam = 2, Judaism = 3, Hinduism = 4, Buddhism = 5, Other = 6, Prefer Not to Say = 7.

- Education (Q7): Unschooled = 1, High school degree or less = 2, A college degree (diploma and equivalent) = 3, Undergraduate (BSC and equivalent) = 4, Masters Degree (MSc or equivalent) = 5, PhD (or equivalent) = 6.
- Employment status (Q8): Employed = 1, Student = 2, Retired =3, Unemployed = 4.
- Software proficiency (Q9): Novice (beginner: need frequent help to use most software applications) = 1, Intermediate (may need help occasionally with some) = 2, Advanced (need little or no help with most) = 3, Expert (provide help to others) = 4.

Appendix E

Equality and Software Think Aloud

E.1 Think Aloud Materials

E.1.1 Information sheet

Research title: Social Sustainability and Software: Case of Equality.

Researcher: Maryam Ali Al Hinai

Department: Informatics, University of Leicester, UK

Introduction:

This think aloud requirements elicitation aims to evaluate the templates suggested in this research in facilitating identification of relevant equality requirements. The overall aim is to support development of software which has no, or minimises negative impact on its users and the surrounding community, in terms of, for instance, discrimination, cyber bullying, and similar social phenomena.

The task will take 30 minutes of requirements identification followed by an interview for 30 minutes.

Please note that both parts of the activity will be audio recorded.

Your confidentiality is assured. The provided answers will be used purely for this PhD research.

Please confirm that you volunteer to participate by selecting signing the consent form below.

For further enquiries, please contact:

Maryam Ali Al Hinai

Department of Informatics

University of Leicester

Leicester, LE1 7RH, UK

Email 1: masah1@le.ac.uk

Email 2: rc256@leicester.ac.uk

Date: 01/10/2016

E.1.2 Consent Form

Please tick the appropriate boxes	**Yes**	**No**
Taking Part		
I have read and understood the project information sheet dated 1/10/2016.		
I have been given the opportunity to ask questions about the project.		
I agree to take part in the project.		
I understand that my taking part is voluntary; I can withdraw from the study at any time and I do not have to give any reasons for why I no longer want to take part.		
Use of the information I provide for this project only		
I understand my personal details will not be revealed to people outside the project.		
I understand that my words may be quoted in publications, reports, web pages, and other research outputs.		

Name of participant	Signature	Date
Researcher	Signature	Date

Project contact details for further information:

Maryam Ali Al Hinai

Department of Informatics

University of Leicester

Leicester, LE1 7RH, UK

E.1.3 Background Information

A What is your age?

- 18 to 24 years
- 25 to 34 years
- 35 to 44 years
- 45 to 54 years
- 55 to 64 years
- Age 65 or older

B What is your gender?

- Male
- Female

C How do you rate your proficiency in software requirements identification?

- Novice (beginner)
- Intermediate
- Advanced
- Expert

D How extensive is your experience of working with Requirements?

- None
- Under 1 year
- 1-3 years
- 3-5 years
- 5-10 years

- over 10 years

E What is your highest educational degree?

- Undergraduate (Bachelor)
- Postgraduate (Master)
- Postgraduate (Doctoral)

F Optional Email, please provide email if you dont mind us (potentially) contacting you with clarification questions about your responses.

E.1.4 Instructions

Definition:

Equality is related to providing all individuals in the society with equal life opportunities without discrimination because of their origin, beliefs, position, or (dis-) abilities [Equality Human Rights Commission, GB]. To support equality through software, we believe that the goals of all directly or indirectly affected stakeholders of a given software system should be fairly considered when implementing that software. Furthermore, equal access to software services necessary to realise the fairly selected goals need to be provided.

Task: Your **task is to identify software's equality requirements** for a Smart Garden application. You are provided with a list of stakeholders in case you need it.

Smart Garden description:

A smart garden application is designed with the goals of preserving water resources consumed in home gardens. The software is to be connected to growing beds. "During their growth period, vegetables receive much more water than needed and this goes unquestioned because it does not harm the plants. However, using the sprinklers versus manually watering a garden consumes water at a much greater and unnecessarily faster rate. The application is needed to "find the balance between water consumed while keeping the healthy nature of the plant. Smart garden application will use sensors that are connected to a board "to measure the water levels in growing beds and ultimately find a solution to reduce the amount of water consumption while maintaining a nourished garden. Gardeners are expected to control irrigation through using the Smart garden application. Source (Resilient Smart Gardens, URL `http://birgit.penzenstadler.de/teach/493a.html,accessed4thNov.2017`).

A While doing the task, please **verbalize** your thoughts and let us know:

- What you are trying to do
- What are you looking for
- What you are thinking about
- What decisions you are making
- Do you feel stuck, confused, frustrate, surprised or impressed? Why?

We will use the cloud icon  to remind you to express your thoughts. Remember to **keep talking**.

E.1.5 Stakeholders list

Stakeholders list: You can use the following list to help you identify the stakeholders related to the described system.

Type of Stakeholder	Description
Normal Operator	Do routine commands, entering and monitoring output of product. Communicate with the maintenance operator and operational support as well as functional beneficiaries (e.g. providing them with processed information, and receiving instructions from them).
Maintenance Operator	Product maintenance (hardware, product faults)
Operational Support	Help desk, trainers (help and training normal operators)
Functional Beneficiary	This type benefits from the output/result created by the product. They contact the operators.
Interfacing System	This represent other systems that interface with the product.
Purchaser	Product manager on behalf of consumers or procurement.
Product Champion (aka Sponsor)	The product champion is critical from before the start of a development, and remains important throughout. The role does not necessarily or even desirably contribute to product requirements: it functions mainly at a political rather than a technical level.
Negative Stakeholder	Anything/one that can be harmed by the product (financially, physically, etc.) and they can harm the system. E.g. householders living close to the route of a planned railway.
Hostile Agent (type of negative stakeholders)	Any role that actively seeks to hinder or harm the development and operation of the System. 'Actively' means using some degree of intelligence and creativity to oppose the System. Examples include military enemies, political and commercial spies, hackers, spammers, virus writers, thieves, fraudsters. Clearly the degree of harm intended by such agents varies from complete destruction through malicious pleasure to unauthorised acquisition of assets (with essentially unintended harm as a side-effect).
Political Beneficiary	Any role in public office or private business that can benefit in terms of power, influence and prestige through the success of the Product.
Financial Beneficiary	This type gets financial gains from the product success.
Regulator	Governmental or other regulators, e.g. ISO
Developer	Develop the system or undertake maintenance role in maintenance contract.
Consultant	From outside the development organization.
Supplier	Responsible for components of the products.
Source (I. Alexander, "A Taxonomy of Stakeholders. Human Roles in System Development ", 2005)	

Within next 30 minutes, please list as many equality requirements as you find relevant to the Smart Garden system (please speak aloud expressing your thoughts, and considerations throughout this task).
Equality Requirements:

You have come to the end of this research activity. Thank you for your participation and if you need to contact us please do not hesitate.

E.1.6 Equality Templates

Equality templates:

You can use any of the following templates to write the requirements statements.

- “[stakeholder] should [verb] [functions/services/resources] to accomplish [goal]”.

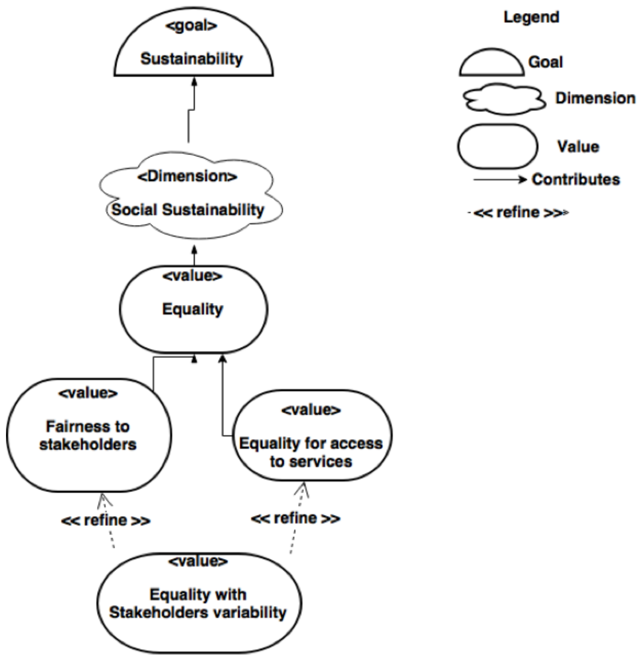
  E.g. [Legal authority] should [be able] to [post legal advice for home gardens] to [share legal best practices]
- “[stakeholder] with [variability] should [verb] [functions/services/resources] to accomplish [goal]”.

  E.g. [Citizen] with [different language] should [be allowed] to [select preferred language] to [customize the health inquiry application]

Stakeholder (1)	Variability (Stakeholders’ gender, age, language, religion, position, income status, knowledge, education, race, technology used, location etc.) (2)	Software functions/services/resources to achieve the goal (3)	Goal(s) (4)
e.g. Legal authority	-	e.g. post legal advice for home gardens	e.g. share legal best practices
e.g. Citizen	e.g. Language	e.g. Select preferred language	e.g. Customize application language

E.1.7 Equality Value Pattern

Equality Value Pattern 



E.1.8 Follow up Interview guide

Follow up Interview guide (Group with No Template)

A Have a look at the two templates (will be provided). Do you think these would have helped you during equality requirements elicitation? Why/Why Not? How?

B How do you think the equality requirements elicitation could be supported? Why are the given templates potentially useful or NOT useful?

Follow up Interview guide (Group with Template)

A Did they help/hinder you in identifying equality requirements? (simple/difficult, easy/difficult to use, time)

B What do you think is the purpose of these templates? Do they fulfil that purpose? Why/Why not?

C What are the advantages/disadvantages of using these templates?

D How can the templates be improved?

Bibliography

[1] Chris Landorf. Evaluating social sustainability in historic urban environments. *International Journal of Heritage Studies*, 17(5):463–477, 2011.

[2] Peter Willis, Stephen McKenzie, and Roger Harris. Introduction: Challenges in adult and vocational education for social sustainability. In *Rethinking Work and Learning*, pages 1–9. Springer, 2009.

[3] Stephen McKenzie. Adult and vocational education for social sustainability: A new concept for tvet for sustainable development. *Work, Learning and Sustainable Development*, pages 177–186, 2009.

[4] Marlies Rybnicek, Rainer Poisel, and Simon Tjoa. Facebook watchdog: a research agenda for detecting online grooming and bullying activities. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*, pages 2854–2859. IEEE, 2013.

[5] Paulo Shakarian, Jana Shakarian, and Andrew Ruef. The dragon and the computer: Why intellectual property theft is compatible with chinese cyberwarfare doctrine. *arXiv preprint arXiv:1309.6450*, 2013.

[6] Ethel Quayle and Max Taylor. Social networking as a nexus for engagement and exploitation of young people. *Information security technical report*, 16(2): 44–50, 2011.

[7] Michael J. Corby. The case for privacy. *Information Systems Security*, 11(2): 9–14, 2002.

[8] Ross J. Anderson, Chris Barton, Rainer Böhme, Richard Clayton, Michel van Eeten, Michael Levi, Tyler Moore, and Stefan Savage. Measuring the cost of cybercrime. In *11th Annual Workshop on the Economics of Information Security, WEIS 2012, Berlin, Germany, 25-26 June, 2012*, 2012.

[9] Christoph Becker, Stefanie Betz, Ruzanna Chitchyan, Leticia Duboc, Steve M. Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. Requirements: The key to sustainability. *IEEE Software*, 33(1):56–65, Jan 2016.

[10] Ruzanna Chitchyan, Christoph Becker, Stefanie Betz, Leticia Duboc, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. Sustainability design in requirements engineering: state of practice. In *Proceedings of the 38th International Conference on Software Engineering, ICSE 2016, Austin, TX, USA, May 14-22, 2016 - Companion Volume*, pages 533–542, 2016.

[11] W3C. Web content accessibility guidelines (wcag) 2.0, 2017. URL `http://www.w3.org/TR/WCAG21/`. Working draft, Accessed: 03-08-2017.

[12] Elizabeth A Dykstra and Robert P Carasik. Structure and support in cooperative environments: The amsterdam conversation environment. *International journal of man-machine studies*, 34(3):419–434, 1991.

[13] Michael J Muller, Daniel M Wildman, and Ellen A White. Taxonomy of participatory design practices: a participatory poster. In *Posters and short talks of the 1992 SIGCHI conference on Human factors in computing systems*, pages 34–34. ACM, 1992.

[14] Michael J Muller and Sarah Kuhn. Participatory design. *Communications of the ACM*, 36(6):24–28, 1993.

[15] Finn Kensing and Jeanette Blomberg. Participatory design: Issues and concerns. *Computer Supported Cooperative Work (CSCW)*, 7(3-4):167–185, 1998.

[16] Jonathan Grudin and John Pruitt. Personas, participatory design and product development: An infrastructure for engagement. In *PDC*, pages 144–152, 2002.

[17] Batya Friedman, Peter Kahn, and Alan Borning. Value sensitive design: Theory and methods. *University of Washington technical report*, pages 02–12, 2002.

[18] Christian Detweiler and Koen V. Hindriks. Value-sensitive design patterns for pervasive health care. In *Tenth Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2012, March 19-23, 2012, Lugano, Switzerland, Workshop Proceedings*, pages 908–913, 2012.

[19] Maaike Harbers and Mark A. Neerincx. Value sensitive design of automated workload distribution support for traffic control teams. In *Engineering Psychology and Cognitive Ergonomics - 11th International Conference, EPCE 2014, Held as Part of HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings*, pages 12–21, 2014.

[20] Kakit Cheong and Alex Mitchell. Human-computer interaction – interact 2015: 15th ifip tc 13 international conference, bamberg, germany, september 14-18, 2015, proceedings, part iii. pages 493–500, Cham, 2015. Springer International Publishing.

[21] B. Longo. R u there? cell phones, participatory design, and intercultural dialogue. *IEEE Transactions on Professional Communication*, 57(3):204–215, 2014.

[22] Usman Mushtaq and Kevin Hall. Power and agency in health information technology: towards a more meaningful participatory design for sustainable

development. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 1036–1041, September 2009.

[23] Christina Hochleitner, Wolfgang Hochleitner, Cornelia Graf, and Manfred Tscheligi. Game user experience evaluation. pages 187–206, Cham, 2015. Springer International Publishing.

[24] L. T. Yong. User experience evaluation methods for mobile devices. In *Innovative Computing Technology (INTECH), 2013 Third International Conference on*, pages 281–286, Aug 2013.

[25] Gao Changyuan, Wang Shiying, and Zhong Chongran. Research on user experience evaluation system of information platform based on web environment. In *Measurement, Information and Control (ICMIC), 2013 International Conference on*, volume 01, pages 558–562, Aug 2013.

[26] Minna Pikkarainen, Outi Salo, Raija Kuusela, and Pekka Abrahamsson. Strengths and barriers behind the successful agile deployment—insights from the three software intensive companies in finland. *Empirical Software Engineering*, 17(6):675–702, 2011.

[27] European Convention on Human Rights. Technical report, European Court of Human Rights, Strasbourg cedex. URL `http://www.echr.coe.int/Documents/Convention_ENG.pdf`.

[28] The universal declaration of human rights. Technical report, United Nations, 1948. URL `http://www.un.org/en/universal-declaration-human-rights/index.html`.

[29] Social Accountability 8000 International Standard, June 2014. URL `http://sa-intl.org/_data/n_0001/resources/live/SA8000%20Standard%202014.pdf`. Accessed: 04-08-2017.

[30] D. Méndez Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J. L. de la Vara, and R. Wieringa. Naming the pain in requirements engineering - contemporary problems, causes, and effects in practice. *Empirical Software Engineering*, 22 (5):2298–2338, Oct 2017.

[31] Lin Liu, Tong Li, and Fei Peng. Why requirements engineering fails: A survey report from china. In *RE 2010, 18th IEEE International Requirements Engineering Conference, Sydney, New South Wales, Australia, September 27 - October 1, 2010*, pages 317–322, 2010.

[32] Norman L Kerth and Ward Cunningham. Using patterns to improve our architectural vision. *IEEE Software*, 14(1):53–59, 1997.

[33] Stephen Withall. *Software Requirement Patterns.* Pearson Education, 2007.

[34] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design patterns: elements of reusable object-oriented software.* Addison-Wesley Professional, 1994.

[35] Markus Schumacher. *Security Engineering with Patterns: Origins, Theoretical Model, and New Applications.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[36] Suzanne Robertson and James Robertson. *Mastering the Requirements Process: Getting Requirements Right.* Addison-Wesley, 3 edition, 2013.

[37] Maria Riaz and Laurie Williams. Security requirements patterns: understanding the science behind the art of pattern writing. In *Second IEEE International*

Workshop on Requirements Patterns, RePa 2012, Chicago, IL, USA, September 24, 2012, pages 29–34, 2012.

[38] Jill Jesson, Lydia Matheson, and Fiona M Lacey. *Doing your literature review: Traditional and systematic techniques*. Sage, 2011.

[39] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages xiii–xxiii, 2002.

[40] Robert Goodland. Sustainability: Human, social, economic and environmental. *Encyclopedia of Global Environmental Change*, 2002.

[41] Ian Sommerville. *Software Engineering, Global Edition*. Pearson Education Limited, 10 edition, 2016.

[42] ISO/IEC. ISO/IEC 25010:2011(en) systems and software engineering systems and software quality requirements and evaluation (SQuaRE) system and software quality models, 2011. URL `https://www.iso.org/obp/ui/#iso:std:iso-iec:25010:ed-1:v1:en`.

[43] Pamela Zave. Classification of research efforts in requirements engineering. *ACM Comput. Surv.*, 29(4):315–321, December 1997.

[44] Bashar Nuseibeh and Steve Easterbrook. Requirements engineering: A roadmap. In *Proceedings of the Conference on The Future of Software Engineering*, ICSE '00, pages 35–46, New York, NY, USA, 2000. ACM. ISBN 1-58113-253-0.

[45] Hila Shapira, Adela Ketchie, and Meret Nehe. The integration of design thinking and strategic sustainable development. *Journal of Cleaner Production*, 140: 277–287, 2017.

[46] Sarah Thew and Alistair Sutcliffe. Value-based requirements engineering: method and experience. *Requirements Engineering*, Jun 2017.

[47] Sarah Louise Thew. Value based requirements engineering. Phd thesis, University of Manchester, UK, 2014.

[48] Philip Morris, Marcelo Masera, and Marc Wilikens. Requirements engineering and industrial uptake. *Requirements Engineering*, 3(2):79–83, Jun 1998.

[49] Nancy R Mead. Why is it so difficult to introduce requirements engineering research results into mainstream requirements engineering practice? In *Proceedings of the 4th International Conference on Requirements Engineering (ICRE'00)*, page 75. IEEE Computer Society, 2000.

[50] Pete Sawyer, Ian Sommerville, and Stephen Viller. Capturing the benefits of requirements engineering. *IEEE software*, 16(2):78–85, 1999.

[51] Markus Dick, Stefan Naumann, and Norbert Kuhn. A model and selected instances of green and sustainable software. In Jacques Berleur, Magda David Hercheui, and Lorenz M. Hilty, editors, *What Kind of Information Society? Governance, Virtuality, Surveillance, Sustainability, Resilience: 9th IFIP TC 9 International Conference, HCC9 2010 and 1st IFIP TC 11 International Conference, CIP 2010, Held as Part of WCC 2010, Brisbane, Australia, September 20-23, 2010. Proceedings*, pages 248–259, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[52] Norbert Kuhn, Stefan Richter, Michael Schmidt, and Andreas Truar. Improving accessibility to governmental forms. In Peter Forbrig, Fabio Paternò, and Annelise Mark Pejtersen, editors, *Human-Computer Interaction Symposium: IFIP 20th World Computer Congress, Proceedings of the 1st TC 13 Human-Computer Interaction Symposium (HCIS 2008), September 7-10, 2008, Milano, Italy*, pages 119–128, Boston, MA, 2008. Springer US.

[53] Stefan Naumann, Eva Kern, Markus Dick, and Timo Johann. Sustainable software engineering: Process and quality models, life cycle, and social aspects.

In Lorenz M. Hilty and Bernard Aebischer, editors, *ICT Innovations for Sustainability*, pages 191–205, Cham, 2015. Springer International Publishing.

[54] Stefan Naumann, Markus Dick, Eva Kern, and Timo Johann. The GREENSOFT model: A reference model for green and sustainable software and its engineering. *Sustainable Computing: Informatics and Systems*, 1(4):294 – 304, 2011.

[55] Felipe Albertao, Jing Xiao, Chunhua Tian, Yu Lu, Kun Qiu Zhang, and Cheng Liu. Measuring the sustainability performance of software projects. In *IEEE 7th International Conference on e-Business Engineering, ICEBE 2010, Shanghai, China, November 10-12, 2010*, pages 369–373, 2010.

[56] Petter Larsson. Energy-efficient software guidelines. White paper, Intel Software Solutions Group, April 2011. URL `https://software.intel.com/sites/default/files/article/57237/energy-efficient-software-guidelines-v3-4-10-11_1.pdf`. Accessed: 22-09-2017.

[57] Nadine Amsel and Bill Tomlinson. Green tracker: A tool for estimating the energy consumption of software. In *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, pages 3337–3342, New York, NY, USA, 2010. ACM.

[58] Birgit Penzenstadler, Joseph Mehrabi, and Debra J. Richardson. Supporting physicians by RE4S: Evaluating requirements engineering for sustainability in the medical domain. In *4th IEEE/ACM International Workshop on Green and Sustainable Software, GREENS 2015, Florence, Italy, May 18, 2015*, pages 36–42, 2015.

[59] Birgit Penzenstadler. Infusing green: Requirements engineering for green in and through software systems. In *Proceedings of the Third International Workshop on Requirements Engineering for Sustainable Systems, RE4SuSy 2014, co-located with 22nd International Conference on Requirements Engineering (RE 2014), Karlskrona, Sweden, August 26, 2014.*, pages 44–53, 2014.

[60] Joseph Mehrabi, Birgit Penzenstadler, and Debra Richardson. Project Cognatio: Developing a System for Medication Adherence (Evaluation of Requirements Engineering for Sustainability). ISR technical report, UCI-ISR-14-4, Institute for Software Research , 2014. URL `http://isr.uci.edu/sites/isr.uci.edu/files/techreports/UCI-ISR-14-4.pdf`. Accessed: 03-08-2017.

[61] Patricia Lago, Sedef Akinli Koçak, Ivica Crnkovic, and Birgit Penzenstadler. Framing sustainability as a property of software quality. *Commun. ACM*, 58 (10):70–78, 2015.

[62] Birgit Penzenstadler and Henning Femmer. A Generic Model for Sustainability . Technical report, Technical University of Munich, November 2012. URL `http://mediatum.ub.tum.de/attfile/1121449/hd2/incoming/2012-Nov/561736.pdf`. Accessed: 04-08-2017.

[63] Birgit Penzenstadler and Henning Femmer. A Generic Model for Sustainability with Process- and Product-Specific Instances. In *Proceedings of the Workshop on Green in/by Software Engineering (GIBSE'13)*, pages 3–8, Fukuoka, Japan, mar 2013. ACM.

[64] Kristin Roher and Debra J. Richardson. Sustainability requirement patterns. In *Third IEEE International Workshop on Requirements Patterns, RePa 2013, Rio de Janeiro, Brazil, July 16, 2013*, pages 8–11, 2013.

[65] Stephen Withall. Software requirement patterns, August 2008. URL http://www.withallyourequire.com/software_requirements_patterns.pdf. Accessed: 02-10-2017.

[66] Kristin Roher and Debra J. Richardson. A proposed recommender system for eliciting software sustainability requirements. In *2nd International Workshop on User Evaluations for Software Engineering Researchers, USER 2013, San Francisco, CA, USA, May 26, 2013*, pages 16–19, 2013.

[67] Martin Mahaux, Patrick Heymans, and Germain Saval. Discovering sustainability requirements: An experience report. In Daniel Berry and Xavier Franch, editors, *Requirements Engineering: Foundation for Software Quality: 17th International Working Conference, REFSQ 2011, Essen, Germany, March 28-30, 2011. Proceedings*, pages 19–33. Springer, 2011.

[68] Jordi Cabot, Steve Easterbrook, Jennifer Horkoff, Lysanne Lessard, Sotirios Liaskos, and Jose-Norberto Mazón. Integrating sustainability in decision-making processes: A modelling strategy. In *Software Engineering-Companion Volume, 2009. ICSE-Companion 2009. 31st International Conference on*, pages 207–210. IEEE, 2009.

[69] José María Conejero, Isabel Sofia Brito, Ana Moreira, Jacome Cunha, and João Araújo. Modeling the impact of UAVs in sustainability. In *24th IEEE International Requirements Engineering Conference, RE 2016, Beijing, China, September 12-16, 2016*, pages 208–216, 2016.

[70] Balbir Barn, Ravinder Barn, and Franco Raimondi. On the role of value sensitive concerns in software engineering practice. In *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, ICSE '15, pages 497–500, Piscataway, NJ, USA, 2015. IEEE Press.

[71] Daisy Yoo, Alina Huldtgren, Jill Palzkill Woelfer, David G. Hendry, and Batya Friedman. A value sensitive action-reflection model: Evolving a co-design space with stakeholder and designer prompts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 419–428, New York, NY, USA, 2013. ACM.

[72] Maryam Al Hinai and Ruzanna Chitchyan. Social sustainability indicators for software: Initial review. In Birgit Penzenstadler, Martin Mahaux, and Camille Salinesi, editors, *Proceedings of the 3rd International Workshop on Requirements Engineering for Sustainable Systems (RE4SuSy'14)*, pages 21–27, Karlskrona, Sweden, Aug 2014.

[73] Guidelines for performing systematic literature reviews in software engineering. Technical report, ver. 2.3 EBSE Technical Report, Keele University, Software Engineering Group, 2007.

[74] J. B. Guinée. *Handbook on life cycle assessment: operational guide to the ISO standards*, volume 7. Kluwer Academic Publishers, Dordrecht;Boston;, 2002.

[75] SNM Menikpura, Shabbir H Gheewala, Sébastien Bonnet, and Chart Chiemchaisri. Evaluation of the effect of recycling on sustainability of municipal solid waste management in thailand. *Waste and Biomass Valorization*, 4(2): 237–257, 2013.

[76] Tomas Ekvall. Nations in social lca. *The International Journal of Life Cycle Assessment*, 16(1):1–2, Jan 2011.

[77] Yosef Manik, Jessica Leahy, and Anthony Halog. Social life cycle assessment of palm oil biodiesel: a case study in jambi province of indonesia. *The International Journal of Life Cycle Assessment*, 18(7):1386–1392, 2013.

[78] Carin Labuschagne, Alan C Brent, and Schalk J Claasen. Environmental and social impact considerations for sustainable project life cycle management in the process industry. *Corporate Social Responsibility and Environmental Management*, 12(1):38–54, 2005.

[79] Laurence Stamford and Adisa Azapagic. Sustainability indicators for the assessment of nuclear power. *Energy*, 36(10):6037–6057, 2011.

[80] Tourane Corbière-Nicollier, Isabelle Blanc, and Suren Erkman. Towards a global criteria based framework for the sustainability assessment of bioethanol supply chains: Application to the swiss dilemma: Is local produced bioethanol more sustainable than bioethanol imported from brazil? *Ecological Indicators*, 11(5):1447–1458, 2011.

[81] Martin C Heller and Gregory A Keoleian. Assessing the sustainability of the us food system: a life cycle perspective. *Agricultural Systems*, 76(3):1007 – 1041, 2003.

[82] Costantino Menna, Domenico Asprone, Fatemeh Jalayer, Andrea Prota, and Gaetano Manfredi. Assessment of ecological sustainability of a building subjected to potential seismic events during its lifetime. *The International Journal of Life Cycle Assessment*, 18(2):504–515, Feb 2013.

[83] G. Assefa and B. Frostell. Social sustainability and social acceptance in technology assessment: A case study of energy technologies. *Technology in Society*, 29(1):63 – 78, 2007.

[84] Murat Kucukvar and Omer Tatari. Towards a triple bottom-line sustainability assessment of the u.s. construction industry. *The International Journal of Life Cycle Assessment*, 18(5):958–972, Jun 2013.

[85] Alan Brent and Carin Labuschagne. Social indicators for sustainable project and technology life cycle management in the process industry. *The International Journal of Life Cycle Assessment*, 11(1):3–15, Jan 2006.

[86] Hamil Pearsall. From brown to green? assessing social vulnerability to environmental gentrification in new york city. *Environment and Planning C: Government and Policy*, 28(5):872–886, 2010.

[87] Hemanta Doloi. Assessing stakeholders' influence on social performance of infrastructure projects. *Facilities*, 30(11/12):531–550, 2012.

[88] Margot J. Hutchins, John S. Gierke, and John W. Sutherland. Decision making for social sustainability: A life-cycle assessment approach. In *2009 IEEE International Symposium on Technology and Society*, pages 1–5, May 2009.

[89] Omidreza Saadatian, Kamaruzzaman Bin Sopian, and Elias Salleh. Adaptation of sustainability community indicators for malaysian campuses as small cities. *Sustainable Cities and Society*, 6:40–50, 2013.

[90] Jian Liu, Jian Zuo, Zhiyu Sun, George Zillante, and Xianming Chen. Sustainability in hydropower developmenta case study. *Renewable and Sustainable Energy Reviews*, 19:230–237, 2013.

[91] Liam Magee, Andy Scerri, and Paul James. Measuring social sustainability: A community-centred approach. *Applied Research in Quality of Life*, 7(3): 239–261, 2012.

[92] Marc Schlossberg and Adam Zimmerman. Developing statewide indices of environmental, economic, and social sustainability: a look at oregon and the oregon benchmarks. *Local Environment*, 8(6):641–660, 2003.

[93] Basil Manos, Thomas Bournaris, and Parthena Chatzinikolaou. Impact assessment of cap policies on social sustainability in rural areas: an application in Northern Greece. *Operational Research*, 11(1):77–92, 2011.

[94] Chengcheng Fan, John D Carrell, and Hong-Chao Zhang. An investigation of indicators for measuring sustainable manufacturing. In *Sustainable Systems and Technology (ISSST), 2010 IEEE International Symposium on*, pages 1–5. IEEE, 2010.

[95] Edwin Chan and Grace KL Lee. Critical factors for improving social sustainability of urban renewal projects. *Social Indicators Research*, 85(2):243–256, 2008.

[96] B Manos, Th Bournaris, P Chatzinikolaou, J Berbel, and D Nikolov. Effects of cap policy on farm household behaviour and social sustainability. *Land Use Policy*, 31:166–181, 2013.

[97] Glen Bramley and Sinead Power. Urban form and social sustainability: the role of density and housing type. *Environment and Planning B: Planning and Design*, 36(1):30–48, 2009.

[98] Benard Omondi Ochieng, Kenneth FD Hughey, and Hugh Bigsby. Rainforest alliance certification of kenyan tea farms: a contribution to sustainability or tokenism? *Journal of cleaner production*, 39:285–293, 2013.

[99] V.P. Vasileiadis, A.C. Moonen, M. Sattin, S. Otto, X. Pons, P. Kudsk, A. Veres, Z. Dorner, R. van der Weide, E. Marraccini, E. Pelzer, F. Angevin, and J. Kiss. Sustainability of european maize-based cropping systems: Economic, environmental and social assessment of current and proposed innovative ipm-based systems. *European Journal of Agronomy*, 48:1 – 11, 2013.

[100] Justus von Geibler, Christa Liedtke, Holger Wallbaum, and Stephan Schaller. Accounting for the social dimension of sustainability: experiences from the biotechnology industry. *Business Strategy and the Environment*, 15(5):334–346, 2006.

[101] Rodolfo Valdes-Vasquez and Leidy E Klotz. Social sustainability considerations during planning and design: framework of processes for construction projects. *Journal of construction engineering and management*, 139(1):80–89, 2012.

[102] Klaas J. Van Calker, Paul B. M. Berentsen, Gerard W. J. Giesen, and Ruud B. M. Huirne. Identifying and ranking attributes that determine sustainability in Dutch dairy farming. *Agriculture and Human Values*, 22(1):53–63, Mar 2005.

[103] Marijke Meul, Steven Van Passel, Frank Nevens, Joost Dessein, Elke Rogge, Annelies Mulier, and Annelies Van Hauwermeiren. MOTIFS: a monitoring tool for integrated farm sustainability. *Agronomy for Sustainable Development*, 28 (2):321–332, Jun 2008.

[104] Gerardo Marletto and Francesca Mameli. A participative procedure to select indicators of policies for sustainable urban mobility. outcomes of a national test. *European Transport Research Review*, 4(2):79–89, Jun 2012.

[105] Abdallah M Hasna. Embedding sustainability in capstone engineering design projects. In *Education Engineering (EDUCON), 2010 IEEE*, pages 1601–1610. IEEE, 2010.

[106] Mahbubul Alam, Yasushi Furukawa, and Kazuhiro Harada. Agroforestry as a sustainable landuse option in degraded tropical forests: a study from Bangladesh. *Environment, Development and Sustainability*, 12(2):147–158, Apr 2010.

[107] Stacy Blake-Beard, Regina O'Neill, Cynthia Ingols, and Mary Shapiro. Social sustainability, flexible work arrangements, and diverse women. *Gender in Management: An International Journal*, 25(5):408–425, 2010.

[108] Joseph Sarkis, Marilyn Michelle Helms, and Aref A Hervani. Reverse logistics and social sustainability. *Corporate Social Responsibility and Environmental Management*, 17(6):337–354, 2010.

[109] Carin Labuschagne, Alan C Brent, and Schalk J Claasen. Environmental and social impact considerations for sustainable project life cycle management in the process industry. *Corporate Social Responsibility and Environmental Management*, 12(1):38–54, 2005.

[110] A J McMichael and J W Powles. Human numbers, environment, sustainability, and health. *BMJ*, 319(7215):977–980, 1999. ISSN 0959-8138.

[111] Tourane Corbiére-Nicollier, Isabelle Blanc, and Suren Erkman. Towards a global criteria based framework for the sustainability assessment of bioethanol supply chains: Application to the swiss dilemma: Is local produced bioethanol more sustainable than bioethanol imported from brazil? *Ecological Indicators*, 11(5):1447 – 1458, 2011.

[112] Kjell Andersson, Per Angelstam, Robert Axelsson, Marine Elbakidze, and Johan Törnblom. Connecting municipal and regional level planning: Analysis and visualization of sustainability indicators in bergslagen, sweden. *European Planning Studies*, 21(8):1210–1234, 2013.

[113] V Ceyhan. Assessing the agricultural sustainability of conventional farming systems in Samsun province of Turkey. *African Journal of Agricultural Research*, 5(13):1572–1583, 2010.

[114] Minna Halme, Christine Jasch, and Michael Scharp. Sustainable homeservices? toward household services that enhance ecological, social and economic sustainability. *Ecological economics*, 51(1):125–138, 2004.

[115] Shannon H Rogers, Kevin H Gardner, and Cynthia H Carlson. Social capital and walkability as social aspects of sustainability. *Sustainability*, 5(8):3473–3483, 2013.

[116] Carina Weingaertner and Åsa Moberg. Exploring social sustainability: learning from perspectives on urban development and companies and products. *Sustainable Development*, 22(2):122–133, 2014.

[117] Zachary A Gennett, Jacqueline A Isaacs, and Thomas P Seager. Developing a social capital metric for use in an educational computer game. In *Sustainable Systems and Technology (ISSST), 2010 IEEE International Symposium on*, pages 1–6. IEEE, 2010.

[118] Francisco das Chagas Oliveira, Ángel Calle Collado, and Luiz Fernando Carvalho Leite. Autonomy and sustainability: An integrated analysis of the development of new approaches to agrosystem management in family-based farming in carnaubais territory, piauí, brazil. *Agricultural systems*, 115:1–9, 2013.

[119] Niels R Faber and René J Jorna. The use of multi-actor systems for studying social sustainability: Theoretical backgrounds and pseudo-specifications. In *Computer Supported Cooperative Work in Design (CSCWD), 2011 15th International Conference on*, pages 842–849. IEEE, 2011.

[120] Grace KL Lee and Edwin HW Chan. Evaluation of the urban renewal projects in social dimensions. *Property Management*, 28(4):257–269, 2010.

[121] Thérésa Lebacq, Philippe V. Baret, and Didier Stilmant. Sustainability indicators for livestock farming. a review. *Agronomy for Sustainable Development*, 33(2):311–327, Apr 2013.

[122] Giacomo Fiumara, Dario Maggiorini, Alessandro Provetti, and Laura Ripamonti. Knowledge representation in virtual teams: a perspective approach for synthetic worlds. *Collaborative Networks for a Sustainable World*, pages 619–625, 2010.

[123] Chi 2014: Toronto, on, canada. URL `http://dblp.uni-trier.de/db/conf/chi/chi2014.html`. Accessed: 21/01/2018.

[124] Andi H Widok, Lars Schiemann, Paul Jahr, and Volker Wohlgemuth. Achieving sustainability through a combination of lca and des integrated in a simulation software for production processes. In *Simulation Conference (WSC), Proceedings of the 2012 Winter*, pages 1–12. IEEE, 2012.

[125] Xin Zhou, Yuqin Jin, He Zhang, Shanshan Li, and Xin Huang. A map of threats to validity of systematic literature reviews in software engineering. In *Software Engineering Conference (APSEC), 2016 23rd Asia-Pacific*, pages 153–160. IEEE, 2016.

[126] Mary L McHugh. Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282, 2012.

[127] Barbara Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University, Software Engineering Group, 2004.

[128] Edwin A Locke. The motivation sequence, the motivation hub, and the motivation core. *Organizational Behavior and Human Decision Processes*, 50(2): 288 – 299, 1991. Theories of Cognitive Self-Regulation.

[129] An-Shou Cheng and Kenneth R Fleischmann. Developing a meta-inventory of human values. *Proceedings of the Association for Information Science and Technology*, 47(1):1–10, 2010.

[130] Milton Rokeach. *The nature of human values.* Free press, 1973.

[131] Shalom H Schwartz. Are there universal aspects in the structure and contents of human values? *Journal of social issues*, 50(4):19–45, 1994.

[132] Clyde Kluckhohn. Values and value-orientations in the theory of action: An exploration in definition and classification. In Talcott Parsons and Edward Shils, editors, *Toward a General Theory of Action*, pages 388–433. Harvard University Press, 1951.

[133] William D. Guth and Renato Tagiuri. Personal values and corporate strategy. Harvard business review, 1965. URL `https://hbr.org/1965/09/personal-values-and-corporate-strategy`. Accessed: 08/01/2018.

[134] Pat Duffy Hutcheon. Value theory: Towards conceptual clarification. *The British Journal of Sociology*, 23(2):172–187, 1972.

[135] Valerie Braithwaite and Russell Blamey. Consensus, stability and meaning in abstract social values. *Australian Journal of Political Science*, 33(3):363–380, 1998.

[136] Batya Friedman, Peter H Kahn Jr, and Alan Borning. Value sensitive design and information systems. In P. Zhang and D. Galletta, editors, *Human-Computer Interaction in Management Information Systems: Foundations*. M.E. Sharpe, Inc., NY, 2006.

[137] Carole L. Jurkiewicz and Robert A. Giacalone. A values framework for measuring the impact of workplace spirituality on organizational performance. *Journal of Business Ethics*, 49(2):129–142, Jan 2004.

[138] Paul McDonald and Jeffrey Gandz. Identification of values relevant to business research. *Human Resource Management*, 30(2):217–236, 1 1991.

[139] Frederick Bird and James A. Waters. The nature of managerial moral standards. *Journal of Business Ethics*, 6(1):1–13, 1987.

[140] Elizabeth C Ravlin and Bruce M Meglino. Effect of values on perception and decision making: A study of alternative work values measures. *Journal of Applied psychology*, 72(4):666–673, 1987.

[141] Lynn R. Kahle, Basil Poulos, and Ajay Sukhdial. Changes in social values in the united states during the past decade. *Journal of Advertising Research*, 28: 35–41, 1988.

[142] George W. England. Personal value systems of american managers. *The Academy of Management Journal*, 10(1):53–68, 1967.

[143] Shalom H. Schwartz and Wolfgang Bilsky. Toward a universal psychological structure of human values. *Journal of Personality and Social Psychology*, 53 (3):550 – 562, 1987.

[144] Shalom H. Schwartz and Wolfgang Bilsky. Toward a theory of the universal content and structure of values: Extensions and cross-cultural replications. *Journal of Personality and Social Psychology*, 58(5):878 – 891, 1990.

[145] World Values Survey Association. World values survey. URL `http://www.worldvaluessurvey.org/`. Accessed: 26-09-2017.

[146] Tony Vinson and Matthew Ericson. The social dimensions of happiness and life satisfaction of australians: Evidence from the world values survey. *International Journal of Social Welfare*, 23(3):240–253, 2014.

[147] William Tov and Evelyn W. M. AU. Comparing well-being across nations: Conceptual and empirical issues. In Susan A David, Ilona Boniwell, and Amanda Conley Ayers, editors, *The Oxford handbook of happiness*. Oxford University Press, 2014.

[148] *The Common Cause Handbook. A Guide to Values and Frames for Campaigners, Community Organisers, Civil Servants, Fundraisers, Educators, Social Entrepreneurs, Activists, Funders, Politicians, and everyone in between*. Public Interest Research Centre PIRC, 2011.

[149] Setha M. Low. Managing change: Sustainable approaches to the conservation of the built environment. In Jeanne Marie Teutonico and Frank Matero, editors, *4th Annual US/ICOMOS International Symposium Organized by US/ICOMOS, the Graduate Program in Historic Preservation of the University of Pennsylvania, and the Getty Conservation Institute, Philadelphia, Pennsylvania April 2001*. Getty Conservation Institute Proceedings Series, 2003. URL `http://www.getty.edu/conservation/publications_resources/books/managing_change_built_enviro.html`. Accessed: 22-09-2017.

[150] Barry Boehm. Value-based software engineering. *SIGSOFT Software Engineering Notes*, 28(2):3–, March 2003. ISSN 0163-5948.

[151] Christian Detweiler and Maaike Harbers. Value stories: Putting human values into requirements engineering. In *Joint Proceedings of REFSQ-2014 Workshops, Doctoral Symposium, Empirical Track, and Posters, co-located with the 20th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2014), Essen, Germany, April 7-10, 2014.*, pages 2–11, 2014.

[152] P. Waddell, A. Borning, M. Noth, N. Freier, M. Becke, and G. Ulfarsson. Microsimulation of urban development and location choices: Design and implementation of urbansim. *Networks and Spatial Economics*, 3(1):43–67, Jan 2003.

[153] Maria Angela Ferrario, Will Simm, Stephen Forshaw, Adrian Gradinar, Marcia Tavares Smith, and Ian Smith. Values-first se: Research principles in practice. In *Proceedings of the 38th International Conference on Software Engineering Companion*, ICSE '16, pages 553–562, New York, NY, USA, 2016. ACM.

[154] Tjerk de Greef, Arvind Mohabir, Ibo van der Poel, and Mark A. Neerincx. sCEthics: Embedding ethical values in cognitive engineering. In *European Conference on Cognitive Ergonomics 2013, ECCE '13, Toulouse, France, August 26 - 28, 2013*, pages 4:1–4:8, 2013.

[155] David Benyon and Catriona Macaulay. Scenarios and the HCI-SE design problem. *Interacting with Computers*, 14(4):397–405, 2002.

[156] Martin Kurdve and Heleen de Goey. Can social sustainability values be incorporated in a product service system for temporary public building modules? *Procedia CIRP*, 64:193 – 198, 2017. 9th CIRP IPSS Conference: Circular Perspectives on PSS.

[157] Ana Paula Barquet, Johannes Seidel, Günther Seliger, and Holger Kohl. Sustainability factors for pss business models. *Procedia CIRP*, 47:436–441, 2016.

[158] Anthony A Leiserowitz, Robert W Kates, and Thomas M Parris. Sustainability values, attitudes, and behaviors: A review of multinational and global trends. *Annu. Rev. Environ. Resour.*, 31:413–444, 2006.

[159] UN General Assembly. United nations millennium declaration. Technical report, 2000. URL http://www.un.org/millennium/declaration/ares552e.pdf. Accessed: 25/01/2018.

[160] Michael OB́rien. Equality and fairness: Linking social justice and social work practice. *Journal of Social Work*, 11(2):143–158, 2011.

[161] Torstein Eckhoff. *Justice: Its determinants in social interaction.* Rotterdam University Press, 1974.

[162] J. Stacy Adams. Inequity in social exchange. volume 2 of *Advances in Experimental Social Psychology*, pages 267 – 299. Academic Press, 1965.

[163] Karen S. Cook and Karen A. Hegtvedt. Distributive justice, equity, and equality. *Annual Review of Sociology*, 9:217–241, 1983.

[164] Understanding equality. URL http://www.equalityhumanrights.com/private-and-public-sector-guidance/education-providers/secondary-education-resources/useful-information/understanding-equality. Accessed: 06-08-2017.

[165] Equality Act 2010 c.15, 2010. URL http://www.legislation.gov.uk/ukpga/2010/15/pdfs/ukpga_20100015_en.pdf. Accessed: 04-08-2017.

[166] Maryam Al Hinai and Ruzanna Chitchyan. Engineering requirements for social sustainability. In *ICT for Sustainability 2016*. Atlantis Press, 2016.

[167] Guidance on social responsibility ISO 26000, 2010. URL http://www.cnis.gov.cn/wzgg/201405/P020140512224950899020.pdf. Accessed: 04-08-2017.

[168] Jane Webster and Richard T Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS quarterly*, pages xiii–xxiii, 2002.

[169] Matthew B. Miles, A. Michael Huberman, and Johnny Saldaña. *Qualitative data analysis: A methods sourcebook*. SAGE, 3 edition, 2014.

[170] Cambridge Dictionaries Online. Pattern, 2016. URL `http://dictionary.cambridge.org/dictionary/english/pattern`. Accessed: 23-08-2017.

[171] Hsiu-Fang Hsieh and Sarah E Shannon. Three approaches to qualitative content analysis. *Qualitative health research*, 15(9):1277–1288, 2005.

[172] Seyed Abbas Hosseinijou, Saeed Mansour, and Mohsen Akbarpour Shirazi. Social life cycle assessment for material selection: a case study of building materials. *The International Journal of Life Cycle Assessment*, 19(3):620–645, 2013.

[173] M.G. Bos, D.H. Murray-Rust, D.J. Merrey, H.G. Johnson, and W.B. Snellen. Methodologies for assessing performance of irrigation and drainage management. *Irrigation and Drainage Systems*, 7(4):231–261, 1993.

[174] Arul Chib and A.L.E. Komathi. Extending the technology-community-management model to disaster recovery: Assessing vulnerability in rural asia. In *Information and Communication Technologies and Development (ICTD), 2009 International Conference on*, pages 328–336, April 2009.

[175] Philip R. Walsh. Creating a "values" chain for sustainable development in developing nations: where maslow meets porter. *Environment, Development and Sustainability*, 13(4):789–805, 2011.

[176] Lin Zhen, Shuyan Cao, Yunjie Wei, Oliver Dilly, Xuelin Liu, Fen Li, Hannes Koenig, Karen Tscherning, and Katharina Helming. Comparison of sustainability issues in two sensitive areas of china. *Environmental Science and Policy*, 12(8):1153 – 1167, 2009.

[177] Kevin Murphy. The social pillar of sustainable development: a literature review and framework for policy analysis. *Sustainability: Science, Practice, & Policy*, 8(1), 2012.

[178] Rungrawee Jitpakdee and Gopal Bahadur Thapa. Sustainability analysis of ecotourism on yao noi island, thailand. *Asia Pacific Journal of Tourism Research*, 17(3):301–325, 2012.

[179] Kevin R. Jones and Lauren Mucha. Sustainability assessment and reporting for nonprofit organizations: Accountability "for the public good". *VOLUNTAS: International Journal of Voluntary and Nonprofit Organizations*, 25(6):1465–1482, Dec 2014.

[180] Rob Bijl. Never waste a good crisis: Towards social sustainable development. *Social Indicators Research*, 102(1):157–168, 2010.

[181] Deepti Sharma and Shardendu Shardendu. Assessing farm-level agricultural sustainability over a 60-year period in rural eastern india. *The Environmentalist*, 31(3):325–337, 2011.

[182] Erik Blair. A reflexive exploration of two qualitative data coding techniques. *Journal of Methods and Measurement in the Social Sciences*, 6(1):14–29, 2015.

[183] Steve Stemler. An overview of content analysis. *Practical assessment, research & evaluation*, 7(17):137–146, 2001.

[184] Helena Sjölin, Veronica Lindström, Håkan Hult, Charlotte Ringsted, and Lisa Kurland. What an ambulance nurse needs to know: A content analysis of curricula in the specialist nursing programme in prehospital emergency care. *International emergency nursing*, 23(2):127–132, 2015.

[185] Anne Wiese, Julian Kellner, Britta Lietke, Waldemar Toporowski, and Stephan Zielke. Sustainability in retailing–a summative content analysis.

International Journal of Retail & Distribution Management, 40(4):318–335, 2012.

[186] Margot J. Hutchins and John W. Sutherland. An exploration of measures of social sustainability and their application to supply chain decisions. *Journal of Cleaner Production*, 16(15):1688 – 1698, 2008.

[187] Joseph Sarkis, Marilyn Michelle Helms, and Aref A Hervani. Reverse logistics and social sustainability. *Corporate Social Responsibility and Environmental Management*, 17(6):337–354, 2010.

[188] Mikel Gurrutxaga. Changes in rural–urban sex ratio differences in the young professional age group as an indicator of social sustainability in rural areas: a case study of continental spain, 2000–2010. *Area*, 45(3):337–347, 2013.

[189] Edward T Jackson. Participatory impact assessment for poverty alleviation: Opportunities for communities and development agencies. *Knowledge, Technology & Policy*, 10(1):6–23, 1997.

[190] Andrew D Basiago. Economic, social, and environmental sustainability in development theory and urban planning practice. *The Environmentalist*, 19 (2):145–161, 1998.

[191] Laura T Raynolds, Douglas Murray, and Andrew Heller. Regulating sustainability in the coffee sector: A comparative analysis of third-party environmental and social certification initiatives. *Agriculture and human values*, 24(2): 147–163, 2007.

[192] William R Sheate, Maria Rosário Do Partidário, Helen Byron, Olivia Bina, and Suzan Dagg. Sustainability assessment of future scenarios: methodology and application to mountain areas of europe. *Environmental management*, 41 (2):282–299, 2008.

[193] Rebecca Spence. Education in post-conflict environments: Pathways to sustainable peace. In Peter Willis, Stephen Mckenzie, and Roger Harris, editors, *Rethinking Work and Learning: Adult and Vocational Education for Social Sustainability*, pages 155–162, Dordrecht, 2009. Springer Netherlands.

[194] A Chel and G Kaushik. Renewable energy for sustainable agriculture. *Agronomy for Sustainable Development*, 2010.

[195] Riccardo Rossi, Massimiliano Gastaldi, and Gregorio Gecchele. Comparison of fuzzy-based and ahp methods in sustainability evaluation: a case of traffic pollution-reducing policies. *European Transport Research Review*, 5(1):11–26, 2013.

[196] Robert Axelsson, Per Angelstam, Erik Degerman, Sara Teitelbaum, Kjell Andersson, Marine Elbakidze, and Marcus K Drotz. Social and cultural sustainability: Criteria, indicators, verifier variables for measurement and maps for visualization to support planning. *Ambio*, 42(2):215–228, 2013.

[197] N-H Schmidt, Koray Erek, Lutz M Kolbe, and Rüdiger Zarnekow. Towards a procedural model for sustainable information systems management. In *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, pages 1–10. IEEE, 2009.

[198] Sandra Buchmüller, Gesche Joost, Nina Bessing, and Stephanie Stein. Bridging the gender and generation gap by ict applying a participatory design process. *Personal and Ubiquitous Computing*, 15(7):743, May 2011.

[199] Ashtarout Nu'man. A framework for adopting e-voting in jordan. *Electronic Journal of e-Government*, 10(2):133–146, 2012.

[200] Mario Toboso. Rethinking disability in amartya sens approach: Ict and equality of opportunity. *Ethics and Information Technology*, 13(2):107–118, 2011.

[201] William N. Robinson and Slav Volkov. A meta-model for restructuring stakeholder requirements. In *Proceedings of the 19th International Conference on Software Engineering*, ICSE '97, pages 140–149, New York, NY, USA, 1997. ACM.

[202] William N Robinson and Vecheslav Volkov. Requirement conflict restructuring. In *Conflict-Oriented Requirements Restructuring GSU CIS Working Paper*, 1999.

[203] Wasi Haider Butt, Sameera Amjad, and Farooque Azam. Requirement conflicts resolution: Using requirement filtering and analysis. In Beniamino Murgante, Osvaldo Gervasi, Andrés Iglesias, David Taniar, and Bernady O. Apduhan, editors, *Computational Science and Its Applications - ICCSA 2011: International Conference, Santander, Spain, June 20-23, 2011. Proceedings, Part V*, pages 383–397, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[204] Barry W. Boehm, Paul Grünbacher, and Robert O. Briggs. Developing groupware for requirements negotiation: Lessons learned. *IEEE Software*, 18(3): 46–55, 2001.

[205] Barry Boehm and Alexander Egyed. Software requirements negotiation: Some lessons learned. In *Proceedings of the 20th International Conference on Software Engineering*, ICSE '98, pages 503–506, Washington, DC, USA, 1998. IEEE Computer Society.

[206] Julian Le Grand. Equality and choice in public services. *Social Research*, 73 (2):695–710, 2006.

[207] Birgit Penzenstadler, Bill Tomlinson, Eric Baumer, Marcel Pufal, Ankita Raturi, Debra Richardson, Baki Cakici, Ruzanna Chitchyan, Georges Da Costa, Lynn Dombrowski, et al. Ict4s 2029: What will be the systems supporting

sustainability in 15 years. In *Proceedings of the 2014 conference ICT for Sustainability*, volume 2, pages 30–39. Atlantis Press, 2014.

[208] Kristen Magis and Craig Shinn. Emergent themes of social sustainability. In Jesse Dillard, Veronica Dujon, and Mary C King, editors, *Understanding the social dimension of sustainability*. Routledge, 2008.

[209] Lawrence Chung and Julio Cesar Sampaio do Prado Leite. On non-functional requirements in software engineering. In Alexander T. Borgida, Vinay K. Chaudhri, Paolo Giorgini, and Eric S. Yu, editors, *Conceptual Modeling: Foundations and Applications: Essays in Honor of John Mylopoulos*, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[210] Lawrence Chung. Non-functional requirements, practices and recommendations @ONLINE. URL `https://www.utdallas.edu/~chung/SYSM6309/NFR-18-4-on-1.pdf`.

[211] Martina Z Huber, Lorenz M Hilty, and Martin Glinz. Uncovering sustainability requirements: An exploratory case study in canteens. In *RE4SuSy@ RE*, pages 35–44, 2015.

[212] Fabiano Dalpiaz, Vítor E. Silva Souza, and John Mylopoulos. The many faces of operationalization in goal-oriented requirements engineering. In *Proceedings of the Tenth Asia-Pacific Conference on Conceptual Modelling - Volume 154*, APCCM '14, pages 3–7. Australian Computer Society, Inc., 2014.

[213] Feng-Lin Li, Jennifer Horkoff, John Mylopoulos, Renata SS Guizzardi, Giancarlo Guizzardi, Alexander Borgida, and Lin Liu. Non-functional requirements as qualities, with a spice of ontology. In *Requirements Engineering Conference (RE), 2014 IEEE 22nd International*, pages 293–302. IEEE, 2014.

[214] Batya Friedman and David Hendry. The envisioning cards: A toolkit for catalyzing humanistic and technical imaginations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 1145–1148, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1015-4.

[215] Lisa P Nathan, Batya Friedman, Predrag Klasnja, Shaun K Kane, and Jessica K Miller. Envisioning systemic effects on persons and society throughout interactive system design. In *Proceedings of the 7th ACM conference on Designing interactive systems*, pages 1–10. ACM, 2008.

[216] Lisa P. Nathan, Predrag V. Klasnja, and Batya Friedman. Value scenarios: A technique for envisioning systemic effects of new technologies. In *CHI '07 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '07, pages 2585–2590, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-642-4.

[217] Richard W Woolridge, Denise J McManus, and Joanne E Hale. Stakeholder risk assessment: An outcome-based approach. *IEEE software*, 24(2), 2007.

[218] Ian F Alexander. A taxonomy of stakeholders. *International Journal of Technology and Human Interaction*, 1(1):23–59, 2005. URL `http://www.scenarioplus.org.uk/papers/stakeholder_taxonomy/stakeholder_taxonomy.htm`. Accessed: 06-08-2017.

[219] W3C. Web accessibility initiative (wai), 2017. URL `http://www.w3.org/WAI/`. Accessed: 25-08-2017.

[220] Rilla Khaled. Equality = inequality: Probing equality-centric design and development methodologies. In Pedro Campos, Nicholas Graham, Joaquim Jorge, Nuno Nunes, Philippe Palanque, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2011: 13th IFIP TC 13 International Conference, Lisbon, Portugal, September 5-9, 2011, Proceedings, Part II*, pages 405–421, 2011.

[221] The National Association of Citizens Advice Bureaux. Discrimination at work - positive action, 2017. URL https://www.citizensadvice.org.uk/work/discrimination-at-work/what-doesn-t-count-as-discrimination-at-work/discrimination-at-work-positive-action/. Accessed: 25-08-2017.

[222] Oxford University Press. English Oxford living dictionaries, 2017. URL https://en.oxforddictionaries.com/definition/bame. Accessed: 05-09-2017.

[223] Chelsea Football Club. Chelsea support for BAME initiative, 2017. URL http://www.chelseafc.com/news/latest-news/2017/01/chelsea-support-for-bame-initiative.html. Accessed: 05-09-2017.

[224] BBC. BBC launches expanded Expert Voices: BAME talent days, 2014. URL http://www.bbc.co.uk/mediacentre/latestnews/2014/bame-talent-days. Accessed: 05-09-2017.

[225] Equality Challenge Unit ECU. ECU's Athena SWAN Charter, 2017. URL http://www.ecu.ac.uk/equality-charters/athena-swan/. Accessed: 05-09-2017.

[226] Chantal M Davies and Muriel Robison. Bridging the gap: An exploration of the use and impact of positive action in the united kingdom. *International Journal of Discrimination and the Law*, 16(2-3):83–101, 2016.

[227] Lizzie Barmes. Equality law and experimentation: The positive action challenge. *Cambridge Law Journal*, 68:623, 2009.

[228] English Oxford living Dictionaries. Positive discrimination, 2017. URL https://en.oxforddictionaries.com/definition/positive_discrimination. Accessed: 05-09-2017.

[229] English Oxford living Dictionaries. Reverse discrimination, 2017. URL https://en.oxforddictionaries.com/definition/reverse_discrimination. Accessed: 05-09-2017.

[230] Legal Dictionary. Reverse discrimination. URL https://legaldictionary.net/reverse-discrimination/. Accessed: 05-09-2017.

[231] Anthony J Onwuegbuzie and Nancy L Leech. Validity and qualitative research: An oxymoron? *Quality & Quantity*, 41(2):233–249, 2007.

[232] Satu Elo, Maria Kääriäinen, Outi Kanste, Tarja Pölkki, Kati Utriainen, and Helvi Kyngäs. Qualitative content analysis: A focus on trustworthiness. *SAGE open*, 4(1):2158244014522633, 2014.

[233] SK Eric and K Yu. Towards modeling and reasoning support for early-phase requirements engineering. In *RE*, pages 226–235, 1997.

[234] Monica Bahl. Requirements management framework & tool, integrating systems thinking into agile requirements engineering to support sustainability. Unpublished Dissertation, Department of Informatics, University of Leicester, 2016.

[235] Sergio Soares, Eduardo Laureano, and Paulo Borba. Implementing distribution and persistence aspects with aspectj. *SIGPLAN Not.*, 37(11):174–190, November 2002. ISSN 0362-1340.

[236] Requirements document for the virtualartviewer. URL https://www.ics.uci.edu/~taylor/ICS_52_FQ02/ics52_Fall02_Req_Ver3.doc. Accessed: 31-10-2016.

[237] Taco van Dijk. Requirements specification for the social travel app, 2012. URL http://www.citysdk.eu/wp-content/uploads/2013/09/DELIVERABLE_WP4_TA_SRS_0.21.pdf. Accessed: 03-11-2016.

[238] Campbell prediction system, 2009. URL `https://kenai.com/projects/cps/pages/SoftwareDevelopmentPlan`. Accessed: 03-11-2016.

[239] Arcade game maker pedagogical product line, 2009. URL `http://www.sei.cmu.edu/productlines/ppl/requirements_model.html`. Accessed: 03-11-2016.

[240] Personal investment management system, 2009. URL `https://www.iiitd.edu.in/~jalote/jalotesebook/JaloteSEbook/CaseStudies/CaseStudy2/SRS.pdf`. Accessed: 27-04-2016.

[241] Collin, Kaitryn, Nancy, Nathan, and Ryan. Resilient smart gardens. Unpublished Course Project, Department of Computer Engineering and Computer Science, California State University, Long BeachAccessed: October 2016.

[242] Sergio Soares, Eduardo Laureano, and Paulo Borba. Implementing distribution and persistence aspects with aspectj. *SIGPLAN Not.*, 37(11):174–190, nov 2002.

[243] Barbara A Kitchenham and Shari L Pfleeger. Personal opinion surveys. In *Guide to Advanced Empirical Software Engineering*, pages 63–92. Springer, 2008.

[244] Charles Teddlie and Fen Yu. Mixed methods sampling. *Journal of Mixed Methods Research*, 1(1):77–100, 2007.

[245] Richard M Heiberger, Naomi B Robbins, et al. Design of diverging stacked bar charts for likert scales and other applications. *Journal of Statistical Software*, 57(5):1–32, 2014.

[246] UNESCO. Digital empowerment: access to information and knowledge using icts for persons with disabilities. UNESCO Document, 2016. URL `http:`

//unesdoc.unesco.org/images/0024/002445/244543e.pdf. Accessed: 06-11-2017.

[247] Centre for Studies on Inclusive Education. Index for inclusion: developing learning and participation in schools, 2017. URL http://www.csie.org.uk/resources/inclusion-index-explained.shtml. Accessed: 06-11-2017.

[248] James Robertson and Suzanne Robertson. Volere requirements specification template. Copyright © the Atlantic Systems Guild Limited, 1995-2007. URL http://www11.informatik.uni-erlangen.de/Lehre/SS2015/PR-SWE/Material/volere-template.pdf. Accessed: 14-08-2017.

[249] Allan Kelly. The agile 10 steps model. *ACCU Overload Journal*, 101, Feb 2011.

[250] Maryam Al Hinai and Ruzanna Chitchyan. Equality requirements for software systems: A survey. In *ICT for Sustainability 2019*, . (Forthcoming).

[251] Martine Van Selm and Nicholas W. Jankowski. Conducting online surveys. *Quality and Quantity*, 40(3):435–456, 2006.

[252] Ian Brace. *Questionnaire design: How to plan, structure and write survey material for effective market research.* Kogan Page Publishers, 2008.

[253] Oluwafemi Osho, Victor Legbo Yisa, and Olawale Joshua Jebutu. E-voting in nigeria: A survey of voters' perception of security and other trust factors. In *Cyberspace (CYBER-Abuja), 2015 International Conference on*, pages 202–211. IEEE, 2015.

[254] Robert A. Hanneman, Augustine J. Kposowa, and Mark D. Riddle. *Basic Statistics for Social Research.* John Wiley and Sons, 2012.

[255] Andy Field. *Discovering statistics using IBM SPSS statistics.* Sage, 2009.

[256] Cyrus R Mehta and Nitin R Patel. Ibm spss exact tests. *IBM Corporation, Cambridge, MA*, 2012.

[257] Jakob Nielsen. 10 usability heuristics for user interface design, 1995. URL `https://www.nngroup.com/articles/ten-usability-heuristics/`. Accessed: 28-10-2017.

[258] W3C. Accessibility, usability, and inclusion: Related aspects of a web for all, 2016. URL `https://www.w3.org/WAI/intro/usable`. Accessed: 29-10-2017.

[259] Keiko Ishii, Charis Eisen, and Hidefumi Hitokoto. The effects of social status and culture on delay discounting. *Japanese Psychological Research*, 59(3), 2017.

[260] Wanjiang Du, Leonard Green, and Joel Myerson. Cross-cultural comparisons of discounting delayed and probabilistic rewards. *The Psychological Record*, 52(4):479–492, 2002.

[261] Michael E Addis and James R Mahalik. Men, masculinity, and the contexts of help seeking. *American psychologist*, 58(1):5, 2003.

[262] Hazwani Mohd Mohadis and Nazlena Mohamad Ali. Designing persuasive application to encourage physical activity at workplace among older workers. In *Digital Information and Communication Technology and its Applications (DICTAP), 2016 Sixth International Conference on*, pages 126–130. IEEE, 2016.

[263] G. H. Freeman and J. H. Halton. Note on an exact treatment of contingency, goodness of fit and other problems of significance. *Biometrika*, 38(1/2):141–149, 1951.

[264] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering.* Springer Science & Business Media, 2012.

[265] K Anders Ericsson and Herbert A Simon. Verbal reports as data. *Psychological review*, 87(3):215, 1980.

[266] Neville A. Stanton, Paul M. Salmon, and Guy H. Walker. *Human Factors Methods: A Practical Guide for Engineering and Design.* Ashgate Publishing Ltd, 2007.

[267] Cynthia J Atman, Justin R Chimka, Karen M Bursic, and Heather L Nachtmann. A comparison of freshman and senior engineering design processes. *Design Studies*, 20(2):131 – 152, 1999.

[268] S. A. Abdulhak, K. H. Cha, and D. K. Kang. Evaluation of online social network games usability using verbal protocol analysis. In *2011 6th International Conference on Computer Sciences and Convergence Information Technology (ICCIT)*, pages 7–11, November 2011.

[269] Tingting Zhao and Sharon McDonald. Keep talking: An analysis of participant utterances gathered using two concurrent think-aloud methods. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, pages 581–590, New York, NY, USA, 2010. ACM.

[270] Po-Yin Yen and Suzanne Bakken. A comparison of usability evaluation methods: heuristic evaluation versus end-user think-aloud protocol-an example from a web-based communication tool for nurse scheduling. In *AMIA*, 2009.

[271] K. D. Schenk, Nicholas P. Vitalari, and K. Shannon Davis. Differences between novice and expert systems analysts: What do we know and what do we do? *Journal of Management Information Systems*, 15(1):9–50, 1998.

[272] Maryam Al Hinai and Ruzanna Chitchyan. Evaluating equality requirements for software systems. In *ICT for Sustainability 2018*, .

[273] Donald T Campbell. Factors relevant to the validity of experiments in social settings. *Psychological bulletin*, 54(4):297, 1957.

[274] Elekes Katalin. “please keep talking”: The ‘think-aloud’ method in second language reading research. *Novelty*, 7(3):48–59, 2000.

[275] Phil Carter. Liberating usability testing. *interactions*, 14(2):18–22, March 2007.

[276] Jacob Buur and Kirsten Bagger. Replacing usability testing with user dialogue. *Commun. ACM*, 42(5):63–66, May 1999.

[277] Safiya Richardson, Rebecca Mishuris, Alexander O’Connell, David Feldstein, Rachel Hess, Paul Smith, Lauren McCullagh, Thomas McGinn, and Devin Mann. “Think aloud” and “near live” usability testing of two complex clinical decision support tools. *International journal of medical informatics*, 106:1–8, 2017.

[278] A Naghiyev, S Sharples, B Ryan, A Coplestone, and M Carey. Real world verbal protocol data analysis of european rail traffic management system train driving and conventional train driving. In *Intelligent Rail Transportation (ICIRT), 2016 IEEE International Conference on*, pages 191–196. IEEE, 2016.

[279] Robert C Wu, M Scott Orr, Mark Chignell, and Sharon E Straus. Usability of a mobile electronic medical record prototype: a verbal protocol analysis. *Informatics for Health and Social Care*, 33(2):139–149, 2008.

[280] Philipp Mayring. Qualitative content analysis, 2000. URL `http://nbn-resolving.de/urn:nbn:de:0114-fqs0002204`. Forum Qualitative Sozialforschung / Forum: Qualitative Social Research. 1(2). Accessed: 15-12-2017.

[281] Suzanne Robertson and James Robertson. *Mastering the Requirements Process: Getting Requirements Right.* Addison-wesley, 2012.

[282] David R. Thomas. A general inductive approach for analyzing qualitative evaluation data. *American Journal of Evaluation*, 27(2):237–246, 2006.

[283] Sedef Akinli Koçak, Gülfem Isiklar Alptekin, and Ayse Basar Bener. Integrating environmental sustainability in software product quality. In *Proceedings of the Fourth International Workshop on Requirements Engineering for Sustainable Systems, RE4SuSy 2015, co-located with the 23rd IEEE International Requirements Engineering Conference (RE 2015), Ottawa, Canada, August 24, 2015.*, pages 17–24, 2015.

[284] Francis T. Edum-Fotwe and Andrew D.F. Price. A social ontology for appraising sustainability of construction projects and developments. *International Journal of Project Management*, 27(4):313 – 322, 2009.

[285] Sandra Buchmüller, Gesche Joost, Nina Bessing, and Stephanie Stein. Bridging the gender and generation gap by ict applying a participatory design process. *Personal and Ubiquitous Computing*, 15(7):743–758, October 2011.

[286] Alan Dennis. *Systems Analysis and Design.* John Wiley & Sons, Inc., 5th edition, 2012.

[287] Richard Berntsson-Svensson and Maryam Taghavianfar. Selecting creativity techniques for creative requirements: An evaluation of four techniques using creativity workshops. In *23rd IEEE International Requirements Engineering Conference, RE 2015, Ottawa, ON, Canada, August 24-28, 2015*, pages 66–75, 2015.

[288] Usman Rafiq, Sohaib Shahid Bajwa, Xiaofeng Wang, and Ilaria Lunesu. Requirements elicitation techniques applied in software startups. In *43rd Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2017, Vienna, Austria, August 30 - Sept. 1, 2017*, pages 141–144, 2017.

[289] Ruzanna Chitchyan, Iris Groher, and Joost Noppen. Uncovering sustainability concerns in software product lines. *Journal of Software: Evolution and Process*, 29(2), 2 2017. ISSN 2047-7481. doi: 10.1002/smr.1853.

[290] Iris Groher and Rainer Weinreich. An interview study on sustainability concerns in software development projects. In *Software Engineering and Advanced Applications (SEAA), 2017 43rd Euromicro Conference on*, pages 350–358. IEEE, 2017.

[291] Jakob Nielsen. Estimating the number of subjects needed for a thinking aloud test. *International journal of human-computer studies*, 41(3):385–397, 1994.

[292] Abraham Nieva de la Hidalga, María Auxilio Medina Nieto, María Leticia Ramírez Castillo, Antonio Benitez Ruiz, and Jorge de la Calleja Mora. Adaptation of biovel portal: a usability case study. In *Software Engineering Research and Innovation (CONISOFT), 2016 4th International Conference in*, pages 185–194. IEEE, 2016.

[293] Steve Easterbrook. Handling conflict between domain descriptions with computer-supported negotiation. *Knowledge acquisition*, 3(3):255–289, 1991.

[294] Anatol Rapoport. Game theory as a theory of conflict resolution. Springer, 1974.

[295] Maulahikmah Galinium, Irvan Prama Defindal, and Ivonne Melissa. E-learning system introduction: equality in education for teachers in rural area

of indonesia. In *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)*, pages 201–206, May 2012.

[296] Minna Isomursu, Mari Ervasti, Marianne Kinnula, and Pekka Isomursu. Understanding human values in adopting new technology-a case study and methodological discussion. *International Journal of Human-Computer Studies*, 69(4):183 – 200, 2011.

[297] Evangelia Kavakli. Goal-oriented requirements engineering: A unifying framework. *Requirements Engineering*, 6(4):237–251, Jan 2002.

[298] Emmanuel Letier and Axel van Lamsweerde. Deriving operational software specifications from system goals. In *Proceedings of the 10th ACM SIGSOFT Symposium on Foundations of Software Engineering*, SIGSOFT '02/FSE-10, pages 119–128, New York, NY, USA, 2002. ACM. ISBN 1-58113-514-9.

[299] Anne Dardenne, Axel Van Lamsweerde, and Stephen Fickas. Goal-directed requirements acquisition. *Science of computer programming*, 20(1-2):3–50, 1993.

[300] Alistair Cockburn. Structuring use cases with goals, 1995. URL `http://alistair.cockburn.us/Structuring+use+cases+with+goals`. Accessed: 20-11-2017.

[301] Jonathan Lee and Nien-Lin Xue. Analyzing user requirements by use cases: A goal-driven approach. *IEEE software*, 16(4):92–101, 1999.

[302] Camille Ben Achour, Colette Rolland, and Carine Souveyet. A proposal for improving the quality of the organisation of scenarios collections. In *4th International Workshop on Requirements Engineering Foundation for Software Quality*, pages 33–45, 1998.

[303] Helen Sharp, Anthony Finkelstein, and Galal Galal. Stakeholder identification in the requirements engineering process. In *Database and Expert Systems*

Applications, 1999. Proceedings. Tenth International Workshop on, pages 387–391. IEEE, 1999.

[304] Athanasia Pouloudi and Edgar A Whitley. Stakeholder identification in inter-organizational systems: gaining insights for drug use management systems. *European journal of information systems*, 6(1):1–14, 1997.

[305] Robert Darimont, Wei Zhao, Christophe Ponsard, and Arnaud Michot. Deploying a template and pattern library for improved reuse of requirements across projects. In *25th IEEE International Requirements Engineering Conference, RE*, page 2, 2017.

[306] W Spencer Smith and Lei Lai. A new requirements template for scientific computing. In *Proceedings of the First International Workshop on Situational Requirements Engineering Processes–Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP*, volume 5, pages 107–121. Citeseer, 2005.

[307] Cristina Palomares, Xavier Franch, and Carme Quer. Requirements reuse and patterns: A survey. In *Requirements Engineering: Foundation for Software Quality - 20th International Working Conference, REFSQ 2014, Essen, Germany, April 7-10, 2014. Proceedings*, pages 301–308. Springer, 2014.

[308] Paul Grünbacher and Norbert Seyff. Requirements negotiation. In Aybüke Aurum and Claes Wohlin, editors, *Engineering and Managing Software Requirements*, pages 143–162, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[309] Ian F Alexander and Ljerka Beus-Dukic. *Discovering requirements: how to specify products and services*. John Wiley & Sons, 2009.

[310] United Nations. Report of the world commission on environment and development: Our common future, 1987. URL `http://www.un-documents.net/ocf-02.htm#I`. Accessed: 16-08-2017.

[311] Birgit Penzenstadler, Henning Femmer, and Debra Richardson. Who is the advocate? stakeholders for sustainability. In *Green and Sustainable Software (GREENS), 2013 2nd International Workshop on*, pages 70–77. IEEE, 2013.

[312] Birgit Penzenstadler, Ankita Raturi, Debra J. Richardson, Coral Calero, Henning Femmer, and Xavier Franch. Systematic mapping study on software engineering for sustainability (SE4S). In *18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, London, England, United Kingdom, May 13-14, 2014*, pages 14:1–14:14, 2014.

[313] Coral Calero and Mario Piattini. Introduction to green in software engineering. In *Green in Software Engineering*, pages 3–27. Springer, 2015.

[314] Mario Rosado de Souza, Robert Haines, and Caroline Jay. Defining sustainability through developers eyes: Recommendations from an interview study. In *Proceedings of the 2nd Workshop on Sustainable Software for Science: Practice and Experiences (WSSSPE 2014)*, 2014.

[315] Forum for the Future. The impact of ict on sustainable development, 2002. URL `http://homepage.cs.latrobe.edu.au/sloke/greenIT/eito_forum_2002.pdf`. Accessed: 03-10-2017.

[316] Peter Madden and Ilka Weißbrod. Connected: ICT and sustainable development, 2008. URL `https://www.forumforthefuture.org/sites/default/files/project/downloads/connected.pdf`. Accessed: 03-10-2017.

[317] Christoph Becker, Ruzanna Chitchyan, Leticia Duboc, Steve Easterbrook, Birgit Penzenstadler, Norbert Seyff, and Colin C. Venters. Sustainability design

and software: The karlskrona manifesto. In *37th IEEE/ACM International Conference on Software Engineering, ICSE 2015, Florence, Italy, May 16-24, 2015, Volume 2*, pages 467–476, 2015.

[318] Birgit Penzenstadler, Veronika Bauer, Coral Calero, and Xavier Franch. Sustainability in software engineering: A systematic literature review. In *16th International Conference on Evaluation Assessment in Software Engineering (EASE 2012)*, pages 32–41, May 2012.

[319] Nelly Condori-Fernandez and Patricia Lago. Characterizing the contribution of quality requirements to software sustainability. *Journal of Systems and Software*, 2017.

[320] Nelly Condori-Fernandez and Patricia Lago. Software sustainability survey, 2017. URL `http://www.s2group.cs.vu.nl/wp-content/uploads/2017/02/surveyS.pdf`. Accessed: 13-12-2017.

[321] Timo Johann and Walid Maalej. Position paper: The social dimension of sustainability in requirements engineering. In *Proceedings of the 2nd International Workshop on Requirements Engineering for Sustainable Systems*. Citeseer, 2013.

[322] Martin Mahaux. Could participation support sustainability in requirements engineering ? In *Proceedings of the 2nd International Workshop on Requirements Engineering for Sustainable Systems, RE4SuSy 2013, Rio, Brasil, July 15, 2013*, 2013.