UTRECHT UNIVERSITY

MASTER'S THESIS

Exploring Ethics in Requirements Engineering

Student:
Abdel-Jaouad ABERKANE

Student number:
5783909

1st *supervisor:* **dr. F.B. Aydemir**
2nd *supervisor:* **dr. F. Dalpiaz**

August 13, 2018

Abstract

There is a growing interest in the ethical dimension of software engineering, especially concerning privacy and security. However, within the phase of requirements engineering, there is a lack of scientific literature that addresses ethical awareness or that identifies ethical issues that may arise during and from the processes of requirements engineering. This thesis explores ethical issues that emerge from software from a requirements engineering perspective to raise ethical awareness about the ethical implications of software.

Both a systematic literature review (SLR) and grey literature review (GLR) were conducted to capture ethical issues from peer-reviewed literature and grey literature, respectively. Peer-reviewed literature provided us with 98 ethical issues and grey literature with 34 ethical issues. Combined, these ethical issues can function as a ethics pattern library for future research of a Code of Ethics. Finally, machine learning techniques were explored for their usability in identifying ethical issues from requirements in requirements engineering. Textual bigrams proved to be the most useful for identifying ethical issues.

Keywords: ethics, requirements engineering, software engineering, software development, systematic literature review, grey literature review, machine learning.

Acknowledgements

All the praises and thanks be to Allah, the Most Gracious, the Most Merciful.

I would like to express my gratitude to dr. Başak Aydemir for the support, motivation, patience, and valuable feedback throughout the process. I could not have imagined having a better mentor for my master's thesis.

My acknowledgement also goes to dr. Fabiano Dalpiaz for the guidance and supportive comments on this thesis, and for allowing me to participate in the organisation of REFSQ 2018 which proved to be a valuable experience.

And last but not least, I would like to express my deepest gratitude to my mother, father, brother, and sister for their support throughout this process.

Abdel-Jaouad Aberkane

Contents

List of Figures

List of Tables

1 Introduction

Computer ethics developments in academia have grown exponentially since 1985 [2]. This advancement leads to new university courses, conferences, and research centres. Simultaneously, the developments in computing technology intensified due to the ample opportunities to make significant amounts of money. The fast pace of these developments leads to an ethics gap [3].

In 1997, the Association for Computing Machinery and the Institute of Electrical and Electronics Engineers Computer Society combined forces and proposed the Software Engineering Code of Ethics in an attempt to diminish this gap by professionalizing the discipline of software engineering and providing the software professionals with a code of conduct that "emphasizes the professional's obligations to the public at large" [4]. The Software Engineering Code of Ethics remains up until this day the most cited Code in the software paradigm of the last 25 years and strengthened the concept of ethics in software engineering [5, 4, 6, 7, 2].[1] However, critics argue that this code of conduct is highly outdated, its content evident, and too general to guide software professionals in ethical decision making [8, 9, 10].

One of the critics is Neil McBride, who argues that clients should be added into the equation when considering ethics in software engineering [8]. The author states that software engineering should be considered a communicative art where engagement with clients is vital. The same author further argues that understanding the task, goals, and moral concerns of the client, results in the development of software that tailored to the client's need. The engagement with the client takes often place at the outset of software engineering, a process which is commonly recognised as requirements engineering [11, 12, 13]. Requirements engineering can, therefore, be seen as a fertile ground upon which software engineering and ethics can collude, and it is the ground upon which we will focus in this thesis.

Requirements Engineering is the process of discovering the intended objective of a software artefact by identifying its stakeholders, individuals or organisations who hold a stake in a system, and their needs [13]. Stakeholders, however, are not always able to determine the quality of the service, which emphasises the need for professionals that make informed ethical choices in design [10]. By raising ethical awareness in requirements engineering, the stage in software development where the software system is to be defined, engaged parties can consider the ethical implications of their work and wishes.

However, an extensive investigation of literature shows us that there is an absence of theoretical framework concerning ethics in requirements engineering.[2] Furthermore, no moral or ethical code of conduct is present in the field of requirements engineering [14]. This shortcoming may turn out to be problematic, especially in this day and age where the so-called information revolution changed many aspects of life [2]. It is therefore that the potential impact of systems on human values should be considered when a system is still malleable [15]. As Epstein eloquently states: "If we can bring mindfulness and compassion and wisdom into the IT workplace, then maybe we can have a positive impact upon the larger human reality." [16].

With this study, we want to provide an ethical response which can aid, among others, the requirements engineer and the engaged stakeholders in understanding the potential ethical impact of a software artefact. By raising awareness of ethical issues during the processes of requirements engineering, the possible negative impact of software artefacts on society and nature may be averted. We aim to build an ethics pattern library that can be used as a base for a code of conduct for requirements engineering. This ethics pattern library consists of two parts. The first part comprises ethical issues derived from academic peer-reviewed literature; the second part is obtained from grey literature. Furthermore, we use natural language processing as a means to explore and experiment with several textual features of requirements

[1] Search conducted on 8 January 2018 using the following search string on Google Scholar: "*software code of ethics*".

[2] See the systematic literature review in Chapter 5.

to identify features that could be of use in recognising ethical issues in requirements engineering with which we intend to prove the usability of the acquired ethical issues in practice.

This thesis is organised as follows. Chapter 2 presents the problem statement and the corresponding research questions. Chapter 3 elaborates on the methodology that will be used to answer the research questions. Chapter 4 depicts a theoretical background of ethics, software ethics and its link to requirements engineering, this section concludes with a brief explanation of the concept of systematic literature reviews and grey literature reviews. In Chapter 5 a systematic literature review is described with which ethical issues related to requirements engineering were obtained from academic literature. Chapter 6 describes a grey literature review through which ethical issues were extracted from grey literature. This is followed by Chapter 7 wherein feature extraction techniques are presented aimed to identify ethical issues in requirements. Finally, in Chapter 8 the conclusions, discussion and potential future work of this research are described.

2 Problem Statement

2.1 Problem Statement

There is a growing interest in ethical issues in software engineering, especially in subdomains such as privacy and security [17, 18, 19, 20, 21]. However, within the domain of requirements engineering, there is a lack of scientific literature that addresses ethical awareness or that identifies ethical issues that may arise during the processes of requirements engineering, software development or software deployment (see Chapter 4.2). In Chapter 1, we elaborated on the need for such a study. This research intends to fill that gap and explore ethics in requirements engineering by gathering ethical issues related to this domain. Furthermore, it aims to support the traditional process of requirements elicitation by assisting in identifying ethical issues in requirements. By creating ethical awareness, we aim to aid the professionals and stakeholders in understanding the potential ethical impact of their work and product.

The first step to do so is collecting ethical issues related to requirements engineering from both scientific and grey literature, thus creating a complete body of relevant ethical issues. Secondly, we aim to design an artefact that can assist the requirements engineering professionals and stakeholders in capturing ethics requirements to raise consciousness. The research questions that underpin this study are presented in the next subsection. Chapter 3 elaborates on the research methodology that is used to answer these research questions.

2.2 Research Questions

To solve the aforementioned problem, we introduce our research questions (RQs) and sub-research questions (SRQs) as follows:

RQ1 "*What are the ethical issues related to requirements engineering?*"

SRQ1 "*What does current scientific literature state about ethical issues related to and requirements engineering?*"

SRQ2 "*What does grey literature state about ethical issues related to requirements engineering?*"

The purpose of these questions is to investigate ethical issues in requirements engineering. The reason to investigate these issues is to get a clear overview of topical issues in this domain. This is needful due to the little scientific literature on this topic, i.e. we were not able to find a literature review or general study concerning ethical issues in requirements engineering. The answers to these questions will provide us with an ethics pattern library that can be used as a base for a code of conduct for requirements engineers. Furthermore, these ethical issues will form the basis for the subsequent research question.

RQ2 "*How to automatically identify ethical issues in requirements?*"
This question is considered to aid the requirements engineering professionals and stakeholders in identifying requirements that may have an unethical implication. After accumulating ethical issues related to requirements engineering, we want to provide the involved parties with apparatus that can be used to identify these ethical issues from requirements in the processes of requirements engineering. In answer to this question, we will assess several textual features on their usability in identifying ethical issues in requirements.

3 Research Methodology

This chapter describes the research methodology that is adopted for this research. This research project aims to explore ethics in requirements engineering. To achieve this, we have adhered to Wieringa's Design Science Methodology to structure this study [1].

3.1 Design Science

According to Wieringa, design problems and knowledge questions are the two main types of research problems in design science [1]. The same author further explains that design problems focus on designing artefacts that intend to help stakeholders: they call for a change in the real world and require an analysis of actual or hypothetical stakeholder goals. Knowledge questions, on the other hand, ask for knowledge about the world as it is [1]. As a means of guidance, the engineering cycle was proposed. However, because treatment implementation in the real world is out of the scope of this study, related steps were excluded from the engineering cycle that is adopted in this research. The remaining steps are often referred to as the design cycle (see Figure 1) [1].

Design problems are treated by following the design cycle which comprises iteration over *problem investigation*, *treatment design*, and *treatment validation*. Knowledge questions are treated by iterating over the empirical cycle. The empirical cycle is a nested cycle within the treatment validation step of the design cycle. It aims to acquire theoretical knowledge and includes the following steps: *research problem analysis*, *research & inference design*, *design validation*, *research execution* and *analysis*. This project will follow this design science methodology as a direction to answer the aforestated research questions.

Abiding by the definitions as mentioned earlier, we categorise research question 1 as a knowledge question, as it aims to acquire theoretical knowledge about ethical issues related to requirements engineering. The outcome of research question 1 will provide us with a deep understanding of ethical issues related to requirements engineering. This insight may lead to further research. Furthermore, the ethical issues will be used in answering the design problem that is addressed in research question 2.

As mentioned in Chapter 2, research question 2 consists of creating an artefact and can, therefore, be classified as a design problem. To answer this question, we will iterate over the design cycle of which the steps are presented in Table 2. The product of research question 2 is the assessment of several textual features and their usability in identifying ethical issues in requirements to aid requirements engineering professionals and stakeholders in capturing ethics requirements. To recapitulate the structure more succinctly, this research iterates over one knowledge problem and one design problem. This section will elaborate on the methodology that is used to tackle these problems.

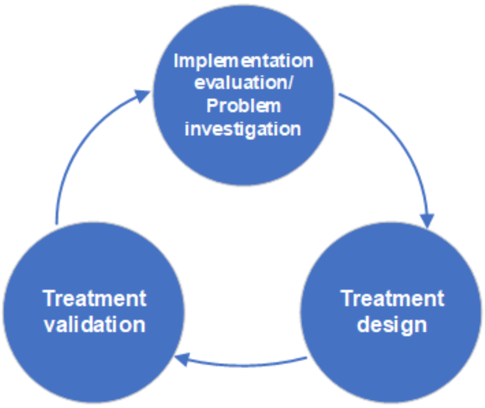

Figure 1: Design cycle of the Design Science Methodology [1].

3.2 Research Question 1

The first research question, "*What are the ethical issues related to requirements engineering?*", aims to explore ethical issues related to requirements engineering. Since it aims to describe *what* the ethical issue rather than exploring *why* they are as they are, we classify this question as a descriptive knowledge question. To answer this question, we need analysable data; this data will be provided by the answers of the underlying sub-research questions as stated in Chapter 2.2. The sub-research questions respectively focus on extracting data from scientific literature and grey literature.

As mentioned before, to tackle this question we will follow the empirical cycle leniently, in other words, if a particular step of the empirical cycle is deemed illogical, it is left out. However, before entering this cycle the research context should be clear [1], we will do so by combining the steps of the research context and research problem analysis. The answer to RQ1 will provide us with a set of ethical issues that can be used to create awareness among the requirements engineering professionals and engaged stakeholders. To answer this question, we will touch upon the domains of ethics, computer ethics, requirements engineering, and software engineering. Stakeholders, i.e. people for whom software is built [22], and the whole of society that can be affected by the artefact are considered as the population in this research context. Therefore, the higher social goal is to create software that is fairer in a sense that it takes its potential ethical implications on society and nature into consideration. In the following subsections, we will present for each sub-research question the remaining stages of the empirical cycle.

3.2.1 Sub-research Question 1

The first sub-research question reads as follows: "*What does current scientific literature state about ethical issues related to requirements engineering?*". Starting with the research & inference design step, we intend to answer this question by conducting a systematic literature review. This review was not designed from scratch; instead we adhered to the guidelines composed by Kitchenham and others [23]. The goal of this systematic review is to outline the ethical issues related to requirements engineering by reviewing the scientific literature.

The design of the systematic literature review was done in several ways. Firstly, a systematic literature review differs from a regular review in a sense that it is considered to be more rigorous which increases its repeatability [23]. Part of this rigour is that systematic reviews start off by defining a review protocol which require a search strategy for the sake of bias elimination, the search strategy that is used for answering this sub-research question can be found in Chapter 5.2.4. The review protocol was validated by conducting several dry-runs and search trials. Before entering the next step of data extraction, a dry-run was conducted by two individuals to validate the data extraction form as presented in Appendix A. After executing the systematic literature review, the extracted ethical issues were analysed both qualitatively and quantitatively, and reported.

Table 1: Steps of the empirical cycle [1].

Step	Description
1. Research problem analysis	*What is the research problem to be solved?*
2. Research design and inference design	*What are you going to do to solve the problem?*
3. Design validation	*How to validate your research design?*
4. Research execution	*Execute research and rapport events related to interpretation of results.*
5. Data analysis	*Analyse data generated by the research.*

3.2.2 Sub-research Question 2

The second sub-research question, *"What does grey literature state about ethical issues related to requirements engineering?"*, aims to identify ethical issues related to requirements engineering by using grey literature as a source. Focusing on grey literature, we aim to outline a more complete and topical map of ethical issues in requirements engineering as the systematic literature review in the first sub-research question focused on peer-reviewed studies only. Peer-reviewed studies are subjected to many formalities, understandably, which is not the case with grey literature. We believe that grey literature, i.e. non-peer-reviewed studies, such as technical reports and magazine articles can provide us with new insights into ethical issues related to requirements engineering. The filter, or hindrance, so to speak, of academic bureaucracy, does not apply here, which lowers the threshold of publishing an article. The grey literature review was designed by following the guidelines as described by Garousi et al. [24].

The design validation of the grey literature review was similar to the validation of the systematic literature review. A predefined review protocol which included the search strategy was defined beforehand to diminish the threat of bias. Furthermore, the form that was used to capture ethical issues (Appendix C) was validated by several dry runs. The captured ethical issues were then analysed and reported.

3.3 Research Question 2

The second research question reads as follows: "*How to automatically identify ethical issues in requirements?*". This question is partly built upon the ethical issues that are the provided by RQ1. The ethical issues and their taxonomy will aid us in investigating the problem of this research question, namely how to capture ethics requirements. During the elicitation process of requirements engineering, the relevant part of the world, i.e. Universe of Discourse, is modelled. This model, preferably an explicit model, should contain all relevant information.

With addressing the second research question, we aim to assist in this process of requirements elicitation by identifying ethical issues from requirements. With this, we want to make the stakeholder and engineer aware of the potential ethical implications of their work and to make them discuss a suitable approach. By making the requirements engineer and stakeholders discuss the ethical issues, we aim to make them actively think and train their conscience. Dialogue is namely essential to better know moral realities, goodness or badness of human actions [25]. This research question will be answered by following the steps of the design cycle presented in Table 2.

The problem investigation step is necessary for getting a deep understanding of the stakeholder goals and the problem. In this step, we have explored several natural language libraries, machine learning techniques, and how they could be of use in the textual analysis of requirements. After obtaining a solid ground of knowledge with regards to machine learning and facilitating libraries, we started with the design of the treatment.

Since natural language is commonly used to express requirements [26], we decided to build a Python script that uses natural language processing as a means to assess textual features on their usability for identifying ethical issues in requirements. Python was chosen due to its capability to utilise several machine learning libraries (e.g. scikit-learn). The outcomes of the analysis were validated both manually and by several computed metrics (e.g. precision).

Table 2: Steps of the design cycle [1].

Step	Description
1. Problem investigation	What phenomena must be improved and why?
2. Treatment design	Design one or more artifacts that could treat the problem.
3. Treatment validation	Would these designs treat the problem?

4 Theoretical Background

This section functions as a transition between the higher level methodology section, the systematic literature review of Chapter 5, and the grey literature review of Chapter 6. In this section, we will present the common ground between ethics and requirements engineering. Furthermore, we will define ethical issues related to requirements engineering. We will then present the outline of a systematic literature review according to the guidelines composed by Kitchenham and others [23]. This section is then concluded by describing the steps of a grey literature review as proposed by Garousi et al. [24].

4.1 Human Action and Ethics in Requirements Engineering

To outline ethical issues in requirements engineering, one should first define the meaning of ethical issues. It is worth noting that this research does not aim to raise awareness among people with malicious intent. Immanuel Kant argued that talents of the mind are doubtless in many respects good and desirable; however, they can become evil and harmful if the will, or character, is not good [27]. In line with this position, this research aims to raise awareness among the people that are open to “good”, because raising awareness among those of ill will may need more effort than a literature review.

In this section, we will first discuss morality in itself and then provide the reader with background information about the two most common schools of thought and how they relate to computer ethics. Finally, we will discuss the link between ethics and computers, and more specific, the software side of computer technology.

4.1.1 What is Morality?

According to the Cambridge Dictionary, ethics is the study of what is morally right and wrong [28]. Berdichevsky and Neuenschwander provide us with a more elaborate definition and state that ethics is a rational and consistent system for determining right and wrong [29]. This system of moral principles can govern a person’s behaviour or conduct, and thus perfect human’s ethical character. Moral principles can also be used as a yardstick to distinguish ethical activities from unethical activities. We study ethics to improve our lives, and therefore its principal concern is the nature of human well-being [30]. From a software perspective, we argue that ethics is important because software can either be both harmful and beneficial to society, which raises the need for ethical-decision making [9].

4.1.2 Computer Ethics: A Short History

The concepts of Computer Ethics originate from Second World War when an American scholar named Norbert Wiener helped to develop an antiaircraft cannon capable of shooting down fast warplanes [2, 31, 32]. During the war Wiener predicted that the world would enter an automatic age with a huge potential for both good and evil [2]. In 1950 Wiener presented a foundation for computer ethics in which he discussed several topics of computer ethics [2, 31].

While teaching a course on medical informatics, Walter Maner argued in 1976 that the involvement of computers in human conduct can create entirely new ethical issues that are unique to computing [33, 2, 31]. Unlike Maner, Deborah Johson asserted that technology, in addition to doing things that we could not do before, enabled us to do the same things but in a other way, initiating the foundationalist debate [34, 35]. Johnson further states that it is important to recognize the intimate connection between technology and human action in order to stop the deflection of human responsibility in technology-instrumented activities, especially when something goes wrong [34].

However, in this study, we focus on a particular aspect of technology-instruments, namely the software aspect. In order to do so we adopt the definition of computer ethics provided by Moor in his highly influential “*What is computer ethics?*” wherein he states that computer ethics is the analysis of the nature and social impact of computer technology and the corresponding formulation and justification of policies for the ethical use of such technology [36, 34]. Applying this definition to the software paradigm The analysis of the nature and social impact of computer technology can lead us to new insights about ethical issues.

In this study, we aim to capture these insights and use them as a commodity of ethical awareness that can be used in requirements engineering. Among other things, requirements engineering is concerned with the identification of goals for a proposed system, the operation and conversion of these goals into services and constraints [11]. During the process of identifying stakeholder goals and their development, ethical awareness can play an essential role in providing the stakeholders with a better understanding of the nature and social impact of the software artefact in question. To do so, we will first define what we mean by stakeholders.

4.1.3 Ethical Stakeholders

Nuseibah and Easterbrook define stakeholders in requirements engineering as individuals or organisations who stand to gain or lose from the success or failure of a system [13]. The authors further argue that an essential philosophical element of RE is that it is concerned with interpreting and understanding stakeholder terminology, concepts, viewpoints, and goals. In this study, we aim to raise a sense of awareness among this group and the requirements engineer with regards to the ethical implications of their requirements. To do so, we have to consider those who are affected by the unethical impact of software, the ones aptly described by Gotterbarn as *ethical stakeholders* [37].

4.1.4 Code of Ethics

To guide the various stakeholders in their ethical-decision making, a Code of Ethics was adopted by the prominent computer societies of Association for Computing Machinery (ACM) and Institute of Electrical and Electronic Engineers Computer Society (IEEE-CS) in 1999 [5, 2]. The Code comprises eight guiding principles and has an educational function since it serves as a means to educate both the public and aspiring professionals about the ethical obligations of all software engineers [38]. The Code consist of the following principles: public, client and employer, product, judgment, management, profession, colleagues and self.

In this thesis, we aim to raise awareness among the direct stakeholders to cater to the ethical stakeholders. To do so, we will use the ACM/IEEE-CS Code of Ethics as a guide for framing ethical issues for the reason that it is the most cited Code of Ethics in the software paradigm and because the field of requirements engineering has no such ethical code [14].

4.1.5 Ethics and Requirements Engineering

According to Johnson, ethical issues in the computer ethics paradigm are considered to be issues that persist during the evolution of computer technology and its deployment in new ways, and seemingly new and unique issues that emerge simultaneously [34]. Deciding not to engage in the foundationalist debate on computer ethics, this research discards the notion of uniqueness while talking about ethical issues [35]. Altering the environment in which the definition of ethical issues provided by Johnson resides from computers to a more specific environment of software, leads us to the following definition:

Definition 4.1. Ethical issues related to requirements engineering
Ethical issues in requirements engineering are (complex) ethical concerns that emerge during the design, development or deployment of software artefacts.

Requirements engineering is one of the most crucial stages in software engineering because it addresses the critical problem of designing the right software for the customer. Relevant values should be considered early during design when a system is still malleable [15]. Furthermore, RE is recognised as the bases of system functionality as it determines the quality of software [11]. By raising awareness in the processes of requirements engineering, involved parties are confronted with the ethical dimension of the potential impact of their artefact.

An example of such an impact is the usage of facial recognition software by the police force in the United States. A recent study shows that state-of-the-art face recognition algorithms are more accurate on Asian facial features when it is created by firms in Asian countries, suggesting that the ethnic composition of a training set impacts the robustness of an algorithm [39]. The usage of facial recognition software by the police force stirred much controversy in the United States: the training data is often biased and consists mostly of white people, which leads to software that often falsely recognises African-Americans as it is not trained on their facial features [40]. These false positives can have far-reaching consequences for individuals that are unjustly "recognised" as criminals. The ethical issue is, in this case, is discriminative software that facilitates the racial discrimination of black people in the United States that are subjected to these algorithms, or *racialised code*. This example stresses the need for ethical awareness and more consciousness in creating software.

4.2 Systematic Literature Reviews in Software Engineering

A systematic literature review is a study that uses a rigorous and well-defined methodology to identify, analyse and interpret all available evidence related to a specific research question in a way that is unbiased and (to a degree) repeatable [23]. In this research, we adhere to the guidelines for performing systematic literature reviews in software engineering proposed by Kitchenham and others.

In general, there are three main reasons for conducting a systematic literature review. Firstly, it aids in summarising the existing evidence concerning a treatment or technology. Secondly, it can be used to identify gaps in current research to suggest areas for further investigation. Lastly, a well-conducted systematic literature review can provide a framework that can be used to position further research activities. The goals of the systematic study presented in Chapter 5 align with the second and third reason.

In this study aim to collect ethical issues related to requirements engineering and hence create a framework of these issues. However, if that is not feasible due to gaps in the literature, the gaps in themselves are proof of the need for ethical awareness in requirements engineering. In addition to systematic literature reviews, systematic mapping studies are also commonly used to conduct a literature review methodically. A systematic mapping study, or scoping study, is a broad review of primary studies in a specific topic area that aims to identify what evidence is available on the topic [23]. However, in this study, we adopted the systematic literature review for reasons we will later present in Chapter 5.2.

A systematic literature review has several advantages. First and foremost, it decreases the possibility of a biased review due to its rigorous and well-defined methodology [23]. Another benefit is that if a phenomenon produces the same effect repeatedly in the majority of articles considered along different settings, conclusions can be drawn. On the other hand, if the effects of a phenomenon are inconsistent, it may be so that the different settings of the considered articles need further investigation. In essence, a systematic literature review can act as a guide or as an initiator for further research. However, one substantial disadvantage is the fact that they require significantly more effort and time than so-called regular reviews.

A systematic review differs in various aspects from regular reviews. According to the guidelines, a systematic literature review should start by composing a systematic review protocol. This protocol should include the structure of the intended review including a predefined search strategy. The goal of a search strategy is to identify as much relevant literature as possible, and by documenting the search strategy the repeatability, rigour and completeness of the systematic review are maintained. In addition to defining the search strategy in advance, the protocol should include explicit inclusion and exclusion criteria that will be used to assess whether an article should be considered or not. The review protocol takes place at the outset of the systematic literature review. Regular reviews are often less rigorous and utilise subjective methods to select and interpret studies.

Systematic literature reviews are conducted in three phases: planning the review, conducting the review and reporting the review [23]. Before conducting a systematic literature review, there should be a rational need for such a study. Therefore, the guidelines urge the researchers to identify such a need and subsequently specify the research questions. As mentioned before, the review protocol should be composed beforehand and evaluated by peers. After validating the protocol, the actual review can be initiated.

To review the literature, one should first obtain the literature. Hence, the first step is to select primary studies by composing search queries based on the research question(s) and using them to search through digital libraries, although some researchers prefer conducting a manual search or combining the two. After obtaining the primary studies, each of them is subjected to the previously mentioned inclusion and exclusion criteria and quality appraisal. Furthermore, the relevant data is extracted from the studies by using a data extraction form. Finally, the extracted data is synthesised and presented in a readable and clear format. The guidelines mentioned in this section are worked out it the systematic literature review presented in Chapter 5.

4.3 Grey Literature Reviews in Software Engineering

Systematic literature reviews can sometimes fall short since they only consider academic peer-reviewed literature [41]. Garousi et al. argue that many essential types of knowledge may not be taken into account if only scholarly articles are examined [41]. Therefore, the authors raise the need for more multivocal literature reviews (MLRs) in software engineering. Multivocal literature is comprised of all accessible writings on a familiar topic, which means that it is not restricted to academic writings only [42]. Therefore, multivocal literature reviews consist of both a systematic literature review and a grey literature review which adds, as its name would suggest, grey literature (e.g. blogs, news articles, white papers) to the set of possible sources [24].

According to Garousi et al., a significant benefit of grey literature reviews is that it can be used to close the gap between academic research and professional practice [41]. Furthermore, expert opinions, as expressed in columns for example, or news reports can provide a comprehensive view of ethical issues related to requirements engineering. Another benefit of a GLR is that it provides actual perspectives which may complement gaps in formal literature [24].

Since grey literature reviews are often published as part of a multivocal literature review, we will present the steps of a multivocal literature review. Similar to SLRs, multivocal literature reviews consist of the same three steps: planning the review, conducting the review, and reporting the review [24]. The first step is to identify the need for a multivocal literature review, which comprises deciding whether to include grey literature in the review (i.e. conducting a SLR or MLR). Thereafter, research questions need to be specified based on this need for reviewing literature. After planning the MLR, we enter the stage of conducting the review which consists of a search process, source selection, study quality assessment, data extraction, and data synthesis [24].

First, the relevant “grey” data sources need to be identified. After determining the sources, the actual search can start. According to the guidelines, there are three possible stopping criteria for searching grey literature: theoretical saturation (i.e. a point where no new evidence emerge from the search), effort bounded (i.e. only including the top N search results), and evidence exhaustion (i.e. extract all evidence). The next step consists of subjecting the obtained literature to inclusion and exclusion criteria and assessing the quality of the literature. After identifying all relevant articles, we enter the data extraction process in which data relevant to the research question is extracted from the relevant articles. This process marks the end of the data extraction phase and initiates the data synthesis phase.

In this phase, the data is synthesised, which can mean qualitative synthesis or quantitative synthesis. Finally, the results and conclusion are presented. The guidelines as mentioned above are elaborated upon in the grey literature review presented in Chapter 6.

5 Systematic Literature Review: Ethical Issues Related to Requirements Engineering

5.1 Introduction

This section presents the systematic literature review that is conducted to answer the first sub-research question mentioned in Chapter 2.2. This literature review aims to outline ethical issues related to requirements engineering to raise awareness about the ethical dimension of software artefacts.

A brief investigation of the literature shows us that there is an absence of theoretical framework concerning ethics in requirements engineering. In order to demonstrate the need of this research, we searched through *IEEE Xplore Digital library* using the following string: **ethics** *AND* **"requirements engineering"**.[3] None of the retrieved articles was related to the research question presented in Chapter 5.2.2. This lack of scientific literature and, more specifically, systematic reviews show us that conducting a literature review will be a difficult undertaking. However, we think that it is important to analyse that what is available and present it systematically for further research.

The methodology adopted to review the literature in this knowledge area is a systematic literature review, a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest [23]. The features of the methodology were disclosed in Chapter 4.2. This research aims to locate gaps in the field of ethics in requirements engineering and to provide a theoretical framework for answering the second research question of this research, as stated in Chapter 2.2.

This section is organised as follows. We first present the Research Methodology which comprises the research strategy, the selection process of the studies and the data extraction, and we then describe the Study Results and the drawn conclusions.

5.2 Research Methodology

In general, two types of systematic literature studies are distinguished: systematic mapping studies and systematic literature reviews. The former focuses on building a classification schema for topics studied in a particular field while the latter focuses on identifying, evaluating and interpreting all available research relevant to a particular research question or topic [23, 43]. Mapping studies are often used to provide a big picture of a publication space while systematic literature reviews are often concerned with analysing and integrating the knowledge contained in the reviewed publication [44].

We aimed to extract ethical issues related to requirements engineering, which means that we conducted an in-depth study of the articles in question and therefore we opted for a systematic literature study. The review process consists of three stages: planning the review, conducting the review and reporting the review which is done in the Study Results section.

5.2.1 Planning the Review

Before all else, the need for a review needs to be identified. We have done so throughout this thesis by arguing that requirements engineering is a crucial stage; hence we are of the opinion that raising awareness in this process will turn out to be a fruitful undertaking. This research is premised on the need for ethical awareness in requirements engineering and the lack of scientific literature that discusses or presents ethical awareness or ethical issues related to requirements engineering. This need was demonstrated through a trial search in Chapter 5.1 which showed us that there is no systematic review available that

[3] On 8 January 2018.

reviews ethical issues related to requirements engineering. Moreover, this research forms the groundwork for the answer to RQ2, which is a need in itself. As mentioned before, this review is conducted to answer SRQ1, which forms the foundation of this systematic review. Using SRQ1 as a guideline, a review protocol was drafted. As mentioned in Chapter 4.2, defining the review protocol in advance reduces the possibility of researcher bias and is therefore considered crucial. The research protocol was then improved and refined using feedback from both supervisors, dr. Aydemir and dr. Dalpiaz. The improved protocol was processed further into this very section.

5.2.2 Research Question

The goal of this review is to outline the ethical issues in requirements engineering. Initial research shows that the body of scientific knowledge concerning this topic is little. Therefore, this review aims to capture that which is at hand in a systematic manner. Ethical issues related to requirements engineering, as defined in Definition 4.1.5, will be explored to assess the state of ethics in requirements engineering. Sub-research question 1 of this research (Chapter 3) forms the foundation of this systematic review:

RQ "*What are the ethical issues related to requirements engineering?*"

Rationale The purpose of this question is to explore ethical issues in or related to requirements engineering. As aforesaid, requirements engineering involves interaction with several stakeholders; this research aims to outline the common ethical issues and how they relate to each other. Subsequently, this can lead us to direct future research, as is the case in this research (Chapter 7). Furthermore, the set of collected ethical issues can provide the involved parties in the requirements engineering process with a better understanding of the possible ethical impact of their work.

5.2.3 Conducting the Review

After the evaluation and refinement of the protocol, the reviewing process was initiated. The first step in conducting the review is identifying the primary studies that need to be reviewed. These studies are searched for using the search strategy that is described in Chapter 5.2.4. This search often results in an abundance of literature of which several cannot be scoped within the SLR. Therefore a selection process for these primary studies is necessary.

5.2.4 Search Strategy

The search strategy of the literature review should be carefully designed since it influences the number and relevancy of the retrieved studies. The approach that was taken in this research is an automated search of digital libraries. Furthermore, only studies that were published between January 1998 and January 2018 were taken into consideration, focusing on contemporary ethical issues. Initially, the following digital libraries were searched: Scopus, IEEE Xplore Digital Library, ScienceDirect, ACM Digital Library and Web of Science. However, after conducting several trial searches, it was decided to use Scopus and IEEE Xplore Digital Library singly due to their high number of results and full coverage.

The search strategy involved identifying the appropriate search terms that could be used to explore the electronic databases. The relevant search terms were acquired by conducting several trial queries that aimed to scope the search query. The keywords were derived from SRQ1. This scoping process resulted in the following search string, which is a database dependant string (i.e. the string might have been adapted to satisfy the syntax of the database in question), that was used for the automated search process:

(**ethics** *OR* **ethical** *OR* **moral** *OR* **"code of conduct"**) *AND* (**"software engineering"** *OR* **"software development"** *OR* **"requirements engineering"** *OR* **"requirement engineering"** *OR* **"requirements analysis"** *OR* **"requirement analysis"** *OR* **"requirements specification"** *OR* **"requirement specifi-**

cation" *OR* **"requirements elicitation"** *OR* **"requirement elicitation"**). [4]

The search query consists of two parts separated by the AND operator. The first part consists of words that aim to identify articles related to ethics. The second part consists of words related to the discipline in which we aim to explore ethics. In addition to requirements engineering, its processes (e.g. requirements analysis), and its synonyms (e.g. requirements analysis), we have included the fields of software engineering and software development as well, since the trial queries including requirements engineering only did not prove to be conducive.

5.2.5 Selection Process

The study selection process a multistage procedure. Figure 2 shows the involved steps and presents the number of studies that survived the three shifts. After obtaining 680 studies from the selected databases according to the described search strategy in Chapter 5.2.4 and exporting the results of the two databases to BibTeX format for further processing, we entered the first step of the selection process. Within this step, the primary studies were subjected to a first shift in which they were examined for duplicates and outliers (e.g. documents that existed of a title page only). This process was facilitated by Jabref, a reference manager system that enabled us to import the two BibTeX files and manually mark outliers from each set of results. After that, we merged the two sets and manually removed the duplicates.

Hereafter in step 2, the resulting studies were held against the inclusion and exclusion criteria as defined in Chapter 5.2.6. This process involved scanning the title, introduction, keywords and abstract of the selected studies and only considering those that did not contravene the inclusion criteria. If it was not possible to exclude a paper without question, the study was not rejected; instead, the complete study was scanned and, again, held against the inclusion and exclusion criteria. The studies that survived this process were then subjected to the data extraction process of Chapter 5.2.7 which comprises of the acquisition of the ethical issues from the selected studies.

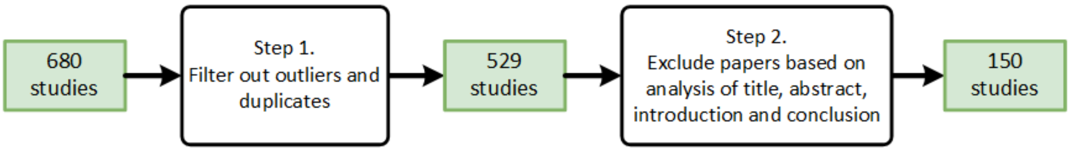
680 studies
Step 1. Filter out outliers and duplicates
529 studies
Step 2. Exclude papers based on analysis of title, abstract, introduction and conclusion
150 studies

Figure 2: The study selection process of the peer-reviewed literature.

5.2.6 Inclusion and Exclusion Criteria

After obtaining the potentially relevant publications, i.e. primary studies, the publications had to be assessed for their actual relevance by holding them against the study selection criteria [45]. The inclusion criteria were used to determine whether an article provides evidence about the research question, and the exclusion criteria were used to determine whether an article should be discarded. Furthermore, the criteria served as quality appraisal as well, diverting from the previously mentioned guidelines by Kitchenham et al. The inclusion and exclusion criteria that were used in the study selection are presented in Table 3.

The first inclusion criterion is necessary to include only peer-reviewed studies. Peer-reviewed studies guarantee a level of quality that is not ensured by non-peer reviewed studies. Furthermore, we aim to include only studies that *focus* on one or more ethical issues or studies that are motivated by one or more ethical issues, in requirements engineering, software engineering, or software development. In this case, *focus* can be explained in two ways. It can either mean that a study discusses one ethical issue or ethical issues as a whole throughout the document. Alternatively, it can mean that a study proposes a solution for an ethical issue, i.e. the ethical issue is (part of) the motivation of the study. If this is not the case,

[4] Search was conducted on 27 February 2018.

Table 3: Inclusion and exclusion criteria of the study selection process.

Inclusion criteria	Exclusion criteria
I1. Peer-reviewed studies (i.e. studies published in a scholarly journal)	**E1.** Non peer-reviewed studies (e.g. magazine articles, technical reports)
I2. Studies that focus on, or are motivated by, one or more, ethical issues related to requirements engineering, software engineering or software development	**E2.** Studies that do not focus on ethics in requirements engineering, software engineering or software development
	E3. Studies published in other language than English
	E4. Studies with 3 pages or less

the study is excluded, hence E2. We do not consider studies that aim to solve general ethical issues by using requirements engineering, software engineering or software development apparatus. Furthermore, if a study is not published in the English language, it is excluded as well. The last exclusion criterion excludes studies that consist of less than 3 pages, intending to select studies with a significant substance.

It is worth explaining how we handled the synthesis of ethical issues since value judgement can sometimes be subjective. Ethical issues were not judged by the authors of this study on their ethicality, i.e. no moral judgement was given. The context of the issue should explicitly or implicitly state that the issue results in ethical concerns. To concretise the assessment process, let us consider the following example. *Professional issues in software engineering curricula: case studies on ethical decision making* is an article written by Georgiadou and Oriogun and presents the way in which Professional Issues (including ethical issues) are introduced and assessed in a software engineering module [46]. In this case, we have classified this study as an experience paper that focuses on ethical dilemmas in software engineering. Therefore it is included, assuming that the other criteria are not breached. On the contrary, if a study discusses ethical issues that did not emerge from software, the study was discarded. An example is a study by Srivastava where the author aims to tackle the issue of corruption in the judicial system of India by creating a tool [47]. This article was discarded for the reason that the ethical issue in question did not emerge from requirements engineering, software engineering, or software development.

Table 4: Results of the Selection Process.

Included or Excluded	Number of Studies
Included	150
Excluded based on E1	78
Excluded based on E2	204
Excluded based on E3	1
Excluded based on E4	84
Unavailable	9

The results of the study selection process are presented in Table 4. 150 Studies survived the inclusion and exclusion criteria. Furthermore, 78 studies were excluded based on exclusion criteria 1, i.e. they were not peer-reviewed. The majority of articles were excluded based on the second exclusion criteria; these studies did not focus on ethics in requirements engineering, software engineering or software development. One study was excluded based on the language in which it was written (i.e. exclusion criteria 3) and 84 studies were deemed insubstantial in size and were excluded based on the last exclusion criteria. Lastly, 11 studies were unfindable or not available through digital means. As a result, the authors of the studies at issue were contacted of which two replied and sent their respective article. These articles were then considered as well.

5.2.7 Data Extraction

To address SRQ1 two types of data extraction were accomplished: extraction of the meta-data of the considered study and the ethical issue(s) mentioned, and an in-depth extraction of the context of each ethical issue. These syntheses were accomplished to allow comprehension about the intrinsic and extrinsic features of the studies that embodied the ethical issues.

Quantitative Synthesis
To facilitate quantitative analysis, an extraction form (see Appendix A) was designed while drafting the research protocol.[5] The form aimed to extract both metadata and intrinsic data from the relevant studies. The data extraction form is structured into four parts. The first part focuses on metadata, which will mostly be extracted automatically. The second part focuses on capturing ethical issues and ethical awareness. The third part focuses on the research methodology of the study at issue; this part will allow us to analyse and draw conclusions about the extracted ethical issues. The last part focuses on data that is unique to empirical research projects, also for analysis purposes. Before starting with the extraction process, a dry run was conducting upon five randomly selected studies to validate the data extraction form. The extracted data was eventually stored in a Microsoft Excel table using the items of the data extraction form as column headers.

The most significant item which is ought to be captured through this form is the ethical issue. The conducted dry-run proved that similar ethical issues are sometimes written or externalised in a slightly different way, which may have negative implications to the quantitative data analysis. To overcome this problem, we took the liberty to capture ethical issues in such a way that they could be categorised. Let us take for example the ethical issue of “Teaching Ethics”. The education of ethics within the software paradigm is discussed in a number of articles, few of them used the exact same terminology. In order to be able to count the frequency of this issue in the selected studies, we decided to use one label, on an atomic level, for all issues with regards to the teaching of ethics.

Narrative Synthesis
In addition to Quantitative Synthesis, Narrative Synthesis was conducted for a more comprehensive understanding of the ethical issues. To address SRQ1, we have attempted to categorise the different types of extracted ethical issues by making use of the principles ACM/IEEE-CS Code of Ethics [48]. This Code of Ethics was used as a guidance because it is the most prevalent one in its number of times cited (see Chapter 1). Even though the Code was designed for software professionals, most of the principles of which the Code is comprised are, in general, applicable to the field of requirements engineering as well. However, some cases may occur where the ethical issues are not classifiable into this framework. For this reason, we have added four principles to the original list of 8, see Table 5 [48].

[5] The filled-in data extraction form for all relevant studies is accessible here: `https://github.com/Aberkane/ethics-in-re/tree/master/RQ1/SRQ1`.

Table 5: 8 Principles of the ACM/IEEE-CS Code of Ethics.

Principle	***Description***
Public	*Software engineers shall act consistently with the public interest*
Client and Employer	*Software engineers shall act in a manner that is in the best interests of their client and employer consistent with the public interest*
Product	*Software engineers shall ensure that their products and related modifications meet the highest professional standards possible*
Judgment	*Software engineers shall maintain integrity and independence in their professional judgment*
Management	*Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance*
Profession	*Software engineers shall advance the integrity and reputation of the profession consistent with the public interest*
Colleagues	*Software engineers shall be fair to and supportive of their colleagues*
Self	*Software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession*

As discussed in the previous paragraph, the 8 principles of the ACM/IEEE Code of Ethics proved to be inadequate to categorise all obtained ethical issues. Therefore 4 principles were added, namely: *Education*, *Research*, *User Behavior*, and *Meta-Ethics* as presented in Table 6. The principle of Education was created to categorise all the ethical issues that focused on educating the future software professionals concerning ethics and replaces the principle of Self. The Study Results section will show us that the numbers of issues that can be labelled as such are plentiful, so much so that we deemed it necessary to categorise them independently from other issues. The Principle of Research was added to categorise all ethical issues that were related to ethical empirical software engineering which focuses mainly on the ethical treatment of human participants. The third principle, the Principle of User-Behavior, is created to classify the issues that are related to the unethical behaviour of users of software artefacts. The last principle, the Principle of Meta-Ethics was added to classify ethical issues or ethical discourse about ethics in the software paradigm.

Table 6: 4 Added principles to the ACM/IEEE-CS Code of Ethics.

Principle	***Description***
Education	*Educating future software professionals in ethical decision-making*
Research	*Taking ethical considerations into account while conducting empirical software engineering*
User-Behaviour	*Unethical behaviour committed by the user of software artefacts*
Meta-Ethics	*Discussion about the nature of ethics*

5.3 Study results

This section presents the results of the systematic literature review following the predefined protocol. We first present the results of the quantitative analysis which comprises the ethical issues that answer the RQ. We then describe the ethical issues and the corresponding context framed into a taxonomical classification.

5.3.1 Quantitative Analysis

Figure 3 shows us that 93% of the selected studies was authored by academics, less than 3% of the selected studies was authored by people from the industry, and 4% of the selected studies was authored by both academics and professionals.

Table 7 shows the distribution of the selected studies concerning Publication type and Research type. 3 Different types of publications were distinguished: conference papers, journal papers, and magazine articles. The distribution shows that the vast majority of the selected studies was published as a conference paper. 47 Of the selected studies were published as a journal paper, and 8 of the selected studies were published as magazine articles.

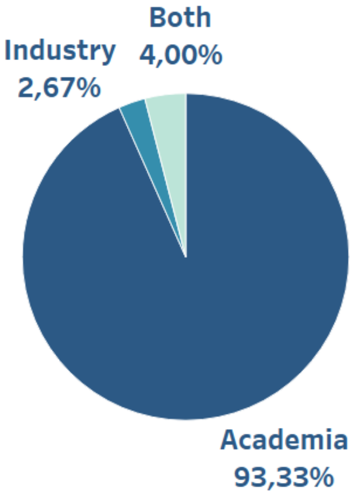

Figure 3: Distribution of authors' affiliation.

As to the Research type, 5 different types were identified: empirical studies, experience papers, meta-studies, position papers, and solution proposals. In contrary to the Publication type, the Research type was not mutually exclusive (e.g. a study could be tagged as both an Empirical study and a Solution proposal). The majority of the selected studies were categorised as solution proposals (e.g. studies presented a software engineering curriculum). Empirical studies were encountered the second most, especially in conference papers (26 studies). Position papers were well-represented in conference and journal publications. Experience papers were most present in conference publications (15 studies). The least published research type were meta-studies: 2 journal papers were categorised as meta-studies against 1 conference paper.

Table 7: Distribution of Publication type and Research type.

	Research type					
Publication Type	Empirical study	Experience paper	Meta-study	Position paper	Solution proposal	*Total*
Conference paper	23	15	1	14	43	94
Journal paper	12	3	2	13	17	46
Magazine article	3	1	0	4	2	10

Concerning the geographical distribution of the selected studies, 29 different countries were identified. As shown in Figure 4, the bulk of the selected studies (61) were affiliated with the United States. Second in place comes the United Kingdom with 21 of the selected studies. Australia comes in third with 8 selected studies. The remainder of the selected studies was affiliated with Canada (7), Finland (7), Germany (7), Sweden (7), the Netherlands (6), China (5), Malaysia (4), India (3), Nigeria (3), Poland (3), Spain (3), Austria (2), Brazil (2), Israel (2), Japan (2), Norway (2), Pakistan (2), Switzerland (2), Egypt (1), Greece (1), Hungary (1), New Zealand (1), Romania (1), Saudi Arabia (1), South Africa (1), and Turkey (1).

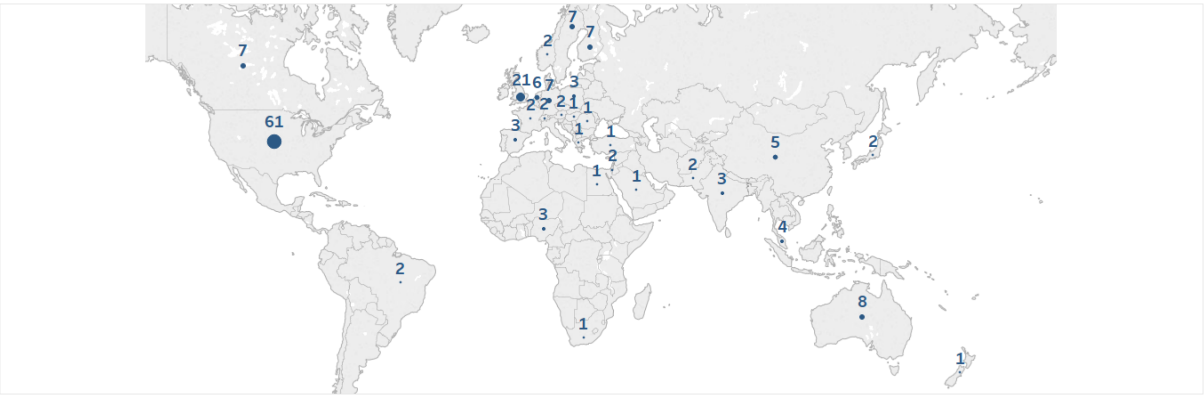

Figure 4: Geographical distribution of the selected studies.

Figure 5 presents the distribution of the selected studies over time. Since the beginning of the observations (1998), at least 3 studies were published per year. The first optimum is visible in 2001 with 11 selected studies. From the beginning of the observations up and until the first optimum, 26 studies were published. Of these 26 studies, 10 studies referred to the ACM/IEEE-CS Code of Ethics or to its draft that was published in 1997 [4]. Between 2001 and 2013 the line becomes erratic and the number of studies fluctuates between 11 and 3 per year. However, a significant increase in related studies is apparent from 2013 onward which may indicate the rise in popularity of ethics within the software paradigm. Another interesting fact that supports this explanation is that 2017 is the most diverse year regarding the country of affiliation of the selected studies: 19 different countries were identified.

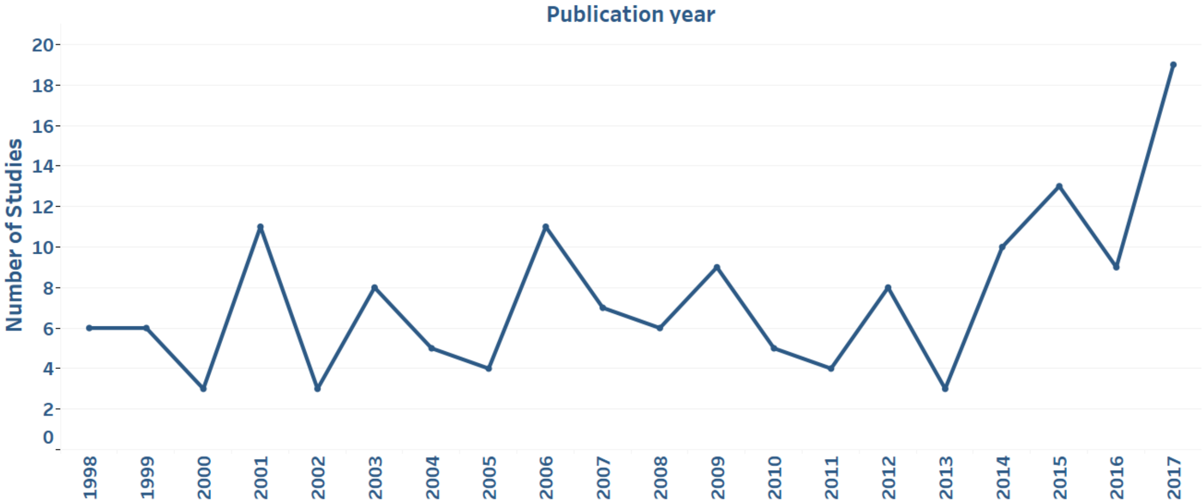

Figure 5: Temporal distribution of selected studies.

Figure 6 shows the 25 most occurring ethical issues as encountered in the selected studies. Responsibility is the ethical issue that was echoed the most in the studies, 35 studies mentioned this ethical issue. In addition to Responsibility, Code of Ethics was an oft-recurring theme as well, counting 27 studies. The third most recurring issue is Privacy which was discussed in 26 of the selected studies. The issue of Teaching Ethics was captured 21 times, Intellectual Property 15 times, Ethical Decision-Making 13 times, Security 11 times, and Informed Consent 7 times. The issues of Autonomy, Computer Abuse, Confidentiality, Digital Divide, Embedded Values, Ethical Decision-Making for AI, and Workplace Ethics occurred 5 times each. Data Collection, Plagiarism, Risks, Software Quality, Virtual Harm, Vulnerability, and Whistle-Blowing occurred 4 times each. The complete list of encountered ethical issues is presented in the table of Appendix B; this table comprises all ethical issues related to requirements engineering, their description, and references. This list can, therefore, be seen as the answer to SRQ1.

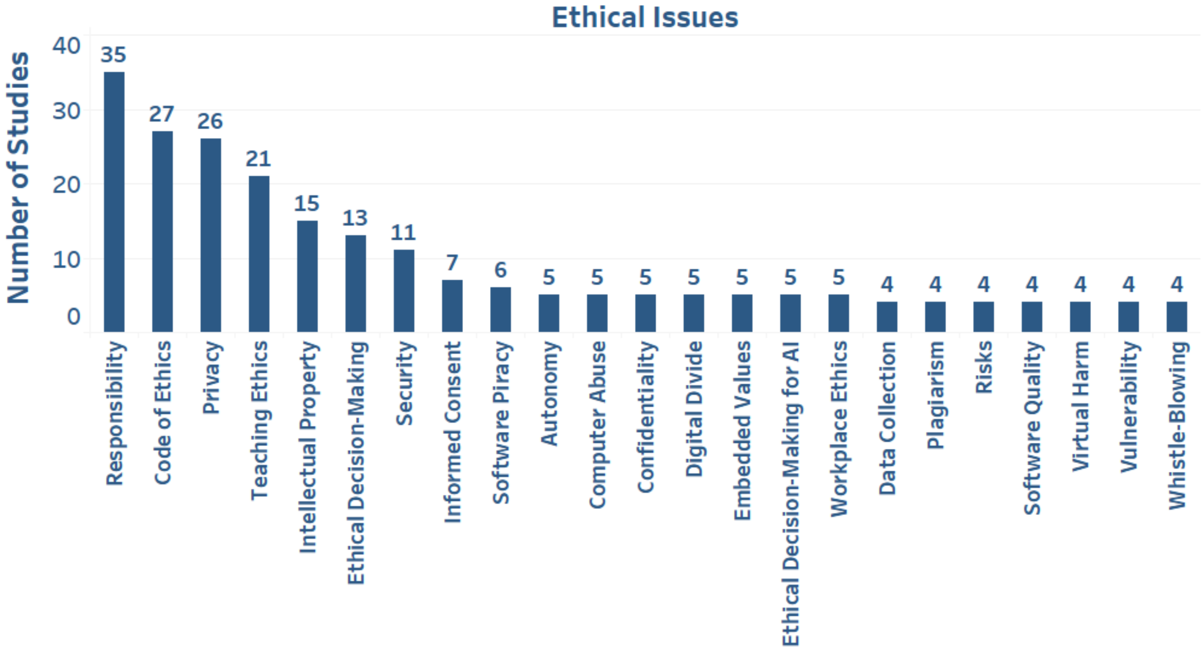

Figure 6: Top-23 most frequent ethical issues in the selected studies.

In Figure 7 the three most frequent issues per year are presented, thus combining the previous two figures. The frequency of the ethical issues is fairly low: no ethical issue was discussed more than four times per year. As commented before, the issue of Code of Ethics is prevalent in the first observed years. Following the example of Code of Ethics, the issue of Responsibility manifests itself as a recurring issue as well, with its first optimum in 2001. In 2002 and 2003 no ethical issue stands out. The optima of 2004 and 2006 are formed by respectively Intellectual Property and Privacy. Between the two, in 2005, no ethical issue occurred more than 1 time. In 2008, the issue of Responsibility was, again, the most occurring issue. The issue of Code of Ethics returns as the most occurring issue in 2009, followed up by Responsibility. Similar to 2005, 2011 was a year of few publications as became apparent from Figure 5. In 2012 Responsibility, Code of Ethics and Ethical Decision-Making reoccurred 3 times each.

The issue of Privacy leaps in 2014 and 2015, which is an interesting development since the last year wherein Privacy was present in the top three was in 2004. This sudden rise of interest in this issue may be explained by the extensively discussed and documented whistle-blowing of Edward Snowden in 2013 when he uncovered governmental spy agencies' mass surveillance behaviour. In 2017, the year wherein most of the selected studies were published, the top three of most occurring ethical issues consists of Responsibility, Privacy, and Ethical Decision-Making. The presence of Responsibility and Ethical Decision-Making may be correlated to the Volkswagen emissions scandal that occurred at the end of 2015 when Volkswagen admitted to the use of falsification software [49].

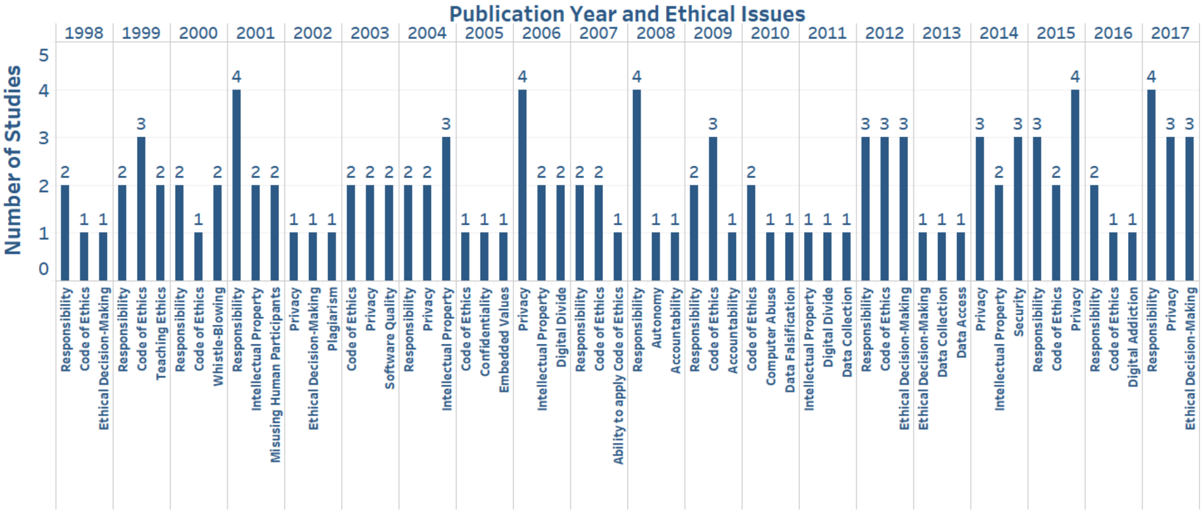

Figure 7: Top-3 most frequent ethical issues per year.

Another interesting metric is the accumulated number of citations of the articles that cover the particular ethical issues. The bar chart in Figure 8 shows the top-25 issues that were cited the most. The issue of Software Piracy comes out on top, a large part of its citations originate from an article by Moores and Chang that was cited 295 times [50]. Second in place comes the issue of Privacy cited 420 times. The issues of Code of Ethics, Responsibility, and Intellectual Property complete the top-5 which were cited 345, 323, and 186 times, respectively.

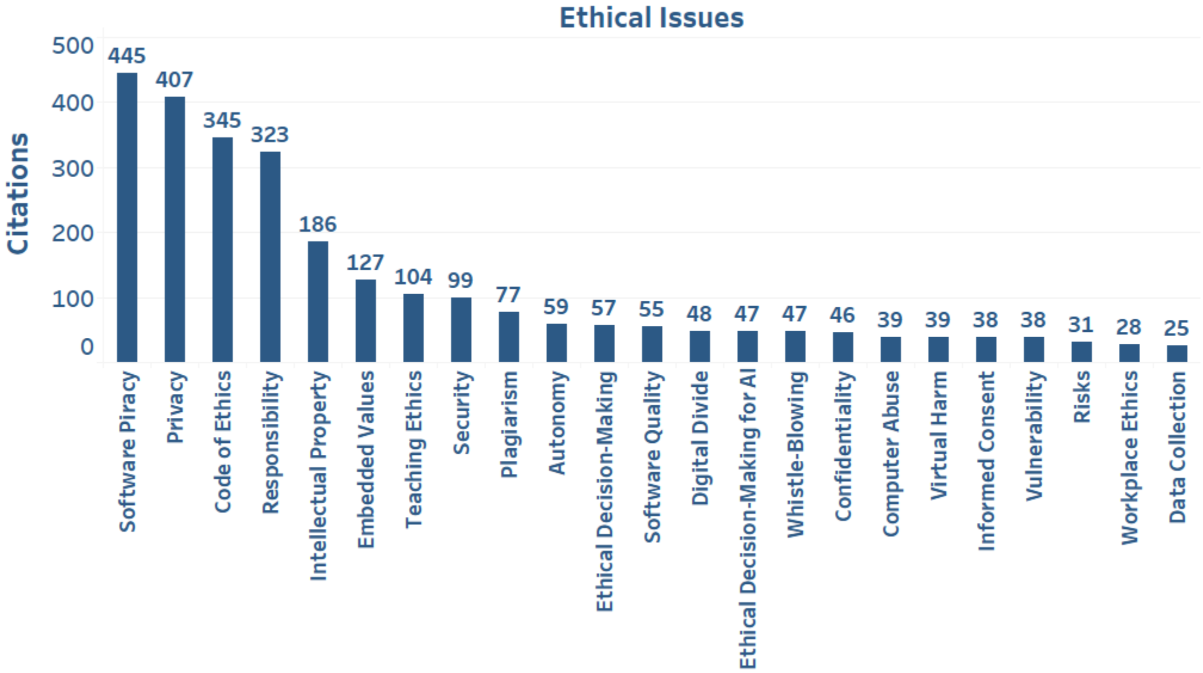

Figure 8: Top-23 most cited ethical issues.

5.3.2 Descriptive Analysis

In this section, we present the results of the descriptive analysis. Each ethical issue that was identified in the literature is presented along with its context. Moreover, the ethical issues are grouped within the principles of the ACM/IEEE-CS Code of Ethics and underlying themes. The ACM/IEEE-CS Code of Ethics is used as a guide and is not leading; therefore several principles have been added, as explained in Chapter 5.2.7. Furthermore, for the sake of readability, the ACM/IEEE-CS Code of ethics is referred to as "*the Code*" or is presented in its full form. Moreover, since the Code is referred to regularly in this section, we have refrained from citing the Code every time it is mentioned.

PUBLIC

The ACM/IEEE-CS Code of Ethics emphasises the centrality of the public interest in software engineering. Furthermore, the Code also states that software engineers need to progress the integrity and reputation of the software engineering profession consistent with the public interest. Similar to the software engineer, the requirements engineer has to be aware of the possible implications of one's work. In this section, we present the encountered ethical issues that are related to the professional's responsibility. We will start off with ethical implications on society followed up by a set of more atomic ethical issues.

Ethical Implications on Society

Software can have far-reaching implications on the society, human values, and quality of life. These concerns could be subjugated by involving users in the software process to build high-quality software, as advocated by Begier [51]. Furthermore, the author conducted a survey to "learn the degree to which the potential threats are known among software authors and users." The results show that today's students and future programmers are conscious of some of the potential threats of global informatisation such as health risks and lack of privacy.

Some studies focused on the ethical implications of software from a requirements engineering perspective. Eden et al. examine how methods and techniques in requirements engineering might contribute to developing an appreciation of the broader contexts in which research and development outcomes are embedded, such as the potential societal impact of ICT [52]. Detweiler and Harbers focus on the same field of requirements engineering and present a method that extracts values and related elements acquired through existing value elicitation techniques [15].

From a higher level of abstraction, Thomson and Schmoldt examine ethical issues of software system design and development concerning privacy, accuracy, property, accessibility, and effects on quality of life [53]. The authors further argue that professional codes of ethics are not very useful because they "hold individuals at fault, not whole organisations, and do not make people more ethical."

Other studies focused more on the values themselves. Ferrario et al. explore the interrelationship between values and software engineering choices by drawing on findings from evidence-based value research [54]. The authors conclude by stating that "we need to speak more openly and less self-consciously about human values, and use tools, languages and techniques that prevent their dismissal." Several authors follow a similar tone and presented methods that aim to aid the engineers in overcoming ethical issues.

Rashid et al. propose an initial framework for managing new ethical concerns during software engineering in society projects [55]. The authors state that dealing with ethical concerns during software engineering in society goes beyond managing the ethical use of information and communication technologies. In fact, questioning core notions of ethics will result in an ethics-aware design that is "critical, situated and reflexive, and, hence, sits well with contemporary digital life."

Ema et al. take a different approach by proposing a system that facilitates rational dialogue between ICT, social sciences and humanities researchers to discuss ethical, legal and social implications of Human Computer Interaction [56].

Embedded Values

Another factor that may have implications for the public are embedded values. Gotterbarn et al. report on their experience in applying the Software Development Impact Statement (SoDIS) process to designing software for Maori culture, outlining the compatibility between a Maori worldview and values embedded in the SoDIS process [57]. Although the process was able to identify some of the "different" unique issues, it was not capable of identifying them all. However, the careful use of SoDIS inspections "may ameliorate the worst forms of recolonisation by Western values through software and may enable the development of software in respectful partnerships".

Barn argues that designers and developers can use values accompanied by an appropriate framework derived from non-functional requirements as a means of discourse of ethical concerns of the design of software [58]. The author, therefore, proposes an extension of an existing classification of requirements to include these value concerns.

Digital Addiction and Digital Divide

The ethical issue of digital addiction was also raised many times in the literature. Alrobai et al. focused on understanding this phenomenon by conducting empirical research to understand users' perception of persuasive intervention technology that aims to combat digital addiction [59]. To raise awareness of the potential consequences of the use of addictive software, Ali et al. propose the implementation of digital addiction labels [60]. The authors conducted an empirical study to investigate the design of these labels from a user's perspective.

With regards to the ethical issue of "digital divide", Busch discusses in his conceptual article the strategies of corporations in the information and communication technologies sector and their role in the conflict over access to knowledge in the digital environment [61]. He hypothesises that "ICT corporations are very capable actors when it comes to bridging digital divides in both developed and developing countries maybe even the most capable actors."

Ethics of Artificial Intelligence

Another issue that has implications on the society is the ethics of artificial intelligence (AI), especially in their decision making. Yilmaz et al. present an overview of machine ethics which results in an understanding of the specific limitations of existing methods in resolving ethical dilemmas for autonomous systems [62]. The authors, therefore, propose a domain architecture "toward developing an Ethical Advisor for training purposes as well as for designing cognitive decision-making models."

A more fundamental question with regards to ethical robots is whether they are possible, or even desirable. Dodig-Crnkovic and Çürüklü answer this in the affirmative and conclude by stating that "artifactual (functional) morality should be built into future robots/softbots, with the aim of ensuring their ethically adequate behaviour" [63]. Somewhat in line with this stance, Kornai argues that artificial general intelligences (AGIs) do not pose an existential threat [64]. The core of the author's proposal is the concept of ethical rationalism which is a means to achieve "guarantees of friendliness in a purely deductive fashion."

Another concern that emerges from ethical decision-making in AI is whether they should be equipped with the capacity to kill as discussed by Fleischman [65]. The author argues that these autonomous robotic weapons should never be developed nor deployed, furthermore, engineers and scientists should "condemn and refuse to participate in their development."

Internet

The internet proved to be an oft-recurring theme in this literature review. Several studies discussed the phenomenon of virtual harm that emerges from it. Watson et al. attempt to address the ethical considerations around internet monitoring technologies that aim to mitigate the problems of internet child abuse [66]. Another issue of virtual harm is discussed by Huff et al. who focus on the ethical issues that emerge from virtual reality environments [67].

Another concern that emerges from the internet is the exposure to particular websites, which might be disturbing for some users. To get more insight in this phenomenon, Usmani et al. propose Internet Age, a software tool that keeps record of the amount of time a user spends on the internet and gives us an insight in the time required to be exposed to certain websites [68].

Another ethical issue that is discussed in the literature is the misuse of Search Engine Optimization methods as discussed by Iqbal et al. [69]. The authors attempt to overcome this problem by presenting an Artificial Immune System for Spam Calculation method that memorises the discriminating attributes of web pages to detect web spam, which is the product of the use of such unethical methods.

Healthcare

A substantial number of articles discussed ethical issues that emerge from software artefacts within the healthcare domain. Kuchinke discusses the ethical concerns that are raised as a product of the Patient Empowerment Tool, a highly integrated tool that enables patient autonomy by giving patients access to their data [70]. An ethical requirements analysis was conducted which resulted in the following five ethical clusters: informed consent, access to own data, display of information and data collection.

Another study that focused on ethical requirements was conducted by Craven et al., where the issue of non-functional requirements with regards to health-care applications is discussed [71]. The authors state that ethical concern for end-user burden dominated the requirements of research institutions ethics committees, which limited the implementation options. Furthermore, user feedback shows that users were concerned with the reliability of data collection.

Sutcliffe et al. explore "affect-laden requirements", factors that determine user acceptance, to achieve system acceptance in healthcare, where ethical and emotional factors can be crucial [72]. Interview results show that seniors regard technology as risky and that they have privacy and security concerns. An example of a privacy concern is geographic data sharing in public health informatics as discussed by Ölvingson et al. [73].

A handful of studies focused on how to surmount ethical issues in the healthcare domain. Kammüller et al. for example provide an overview of their application of formal techniques to improve security and privacy of human-centric Internet of Things systems in the healthcare domain [17]. The authors combine a framework for ethical requirement elicitation (eFRIEND) with automated reasoning. Alike, Aumayr et al. explore how to overcome barriers (e.g. data protection) and enhance valuable and widespread user integration in Ambient Assisted Living (AAL) systems [74]. Furthermore, one study proposes a Multi-Agent System that abides by ethical regulations for healthcare data and enables secure communication, uniform home site authentication and customized resource access authorization [21].

Privacy

A significant number of articles discussed the ethical issue of privacy. Begier presents an ethical approach to introduce a smart metering system, which comprises the consulting of inhabitants [75]. Citizens were interviewed, and questionnaires were conducted to capture citizens' concerns, of which some were related to fear of privacy intrusion. The same was true for a study that Khalid et al. conducted in which user-reviews for IOS apps were considered to understand user-reported issues [76]. The results show that users are most displeased by "issues related to the invasion of their privacy and unethical actions of the app developer."

Other studies took privacy into account while developing software artefacts. Thain et al. explore how to measure the spread of one open-source software package deployed around the world, taking the ethical issue of automatic data collection and data publication into consideration [77]. The authors conclude by saying that "that users are far more amenable to the data collection when they have some understanding of its content and connection to its use." Cary et al. explore the ethical concerns and issues raised by the mining of consumer data and propose a set of system development practices for incorporating an ethical risk management strategy into the process of developing applications that aim to collect or utilise personal information for “data mining related activities” [18].

Contrarily, there was also a study that indicated users’ acceptance of their data being collected. Zhang et al. conducted empirical research to explore the use of social robots in public spaces, in order to understand the social acceptability of robots [78]. Results show that the surveyed people were comfortable with flying drones as long as they have robust safety mechanisms and any personal data captured is “protected by an official government based agency.”

Several studies presented tools in an attempt to overcome the issue of privacy. Kung et al. propose an approach to integrate privacy issues into a privacy-by-design process, based on the use of the concept of socio-contextual, ethical, legal and technical (SALT) frameworks and an associated SALT Framework Management Tool (SFMT) [19]. Kaasinen et al. present an approach in identifying user requirements for a computing architecture that facilitates mobile applications sensing their environment [79]. In addition to privacy, the authors discuss the ethical issues of security and trust as well.

Hiring

Natural language processing is nowadays applied in several fields, including the field of Human Resources which may give rise to ethical concerns that impact the public as pointed out by Chwastek [80]. The author discusses the possibility of discrimination as a result of a filter bubble and destabilisation of the market due to a supply bubble (produced by prediction algorithms).

Environment

Besides impacting human lives, software can also impact the environment of human beings. Tee et al. conducted an empirical study on the motivating factors that influence the Green Software Development (Green SD) [81]. Results show that the intention of software practitioners to develop nature-friendly software products is driven by their environmental concerns, sense of care and responsibility to our natural environment. Furthermore, Moraga et al. present a green code of ethics (GreCo) for software engineering and professional practices as contemporary codes of ethics “do not pay sufficient attention to one of the most important trends to have appeared in the last years’ environmental issues” [82].

CLIENT AND EMPLOYER

The relationship between client and employer is fundamental in software engineering; therefore one of the principles of the ACM/IEEE-CS Code of Ethics urges the software engineers to act in a way that is in the best interest of their employer or client. The engagement of stakeholders is considered an essential factor in the process of requirements engineering as well as highlighted several times in this research. In particular, the reviewed literature focused on dishonesty during software projects.

Lying on Software Projects

All the encountered ethical issues concerning the relationship between client and employers are related to lying. Glass et al. investigated the phenomenon of lying in the software field by surveying active software practitioners and “those closely connected with software practice” [83]. The results show that the most dominant form of lying on software projects was with regards to providing cost and schedule estimates in an early phase of a project. To explain this phenomenon of lying, Smith and Keil proposed a theoretical model, that can be used by both researchers and practitioners, to explain the reluctance to transmit negative project status information within a software project context [84].

PRODUCT

The higher goal of requirements engineering is developing a software artefact or a product. With regards to this product, the ACM/IEEE-CS Code of Ethics states that “Software engineers shall ensure that their products and related modifications meet the highest professional standards possible.” Evidently, this principle is highly related to the field of requirements engineering where the product demands are defined and externalised. First, we will present the studies that discussed issues connected to software quality. Next, studies that discuss issues related to accountability and vulnerabilities will be presented.

Quality

One of the studies that focused on high standard software is the study by Hepting where the SOLE (Sustainable Organic Local Ethical) acronym is introduced into the software paradigm [85]. The author examines whether "SOLE can be meaningfully used to signify software of the highest standard", by taking the dimensions that make up the full form of SOLE into account.

Staples et al. focus on validating data analytics systems and put forward a research agenda, which includes validating their ethical and legal use [86]. The authors stress the importance of continuous validating the usage of a system as the needs of a system, i.e. functionality, nonfunctional qualities, and broader ethical/legal constraints, change over time.

Larsson and Crnkovic analyse the ethical grounds on which decisions, which might have unforeseen implications, are made by focusing on the prediction of quality attributes [87]. Prediction-enabled technology, the author say, aims to determine the component behaviour and predict the system behaviour with a certain degree of accuracy. However, the risk is that these predictions may be perceived as "absolutely correct" and that the system will work accordingly.

Kutar et al. discuss the ethical dilemma's that occur in the packaged software industry [88]. More specifically, the authors argue that vendors that sell culturally loaded products across multicultural contexts may encounter ethical dilemma's.

Accountability

Pieters and van Cleeff discuss the phenomenon of deperimeterization, i.e. the disappearing of boundaries between systems and organisations [89]. More specific, they argue that "deperimeterization implies not only that the border of the organisation's IT infrastructure becomes blurred, but also that the border of the organisation's accountability fades."

Turili analyses several aspects of the relationship between the design of computational artefacts and their ethical implications [90]. The result shows that the ethical implications of the degree of autonomy of computational artefacts imply an increase in the responsibility and accountability of software designers.

Vulnerabilities

Another significant item is the issue of software vulnerability which is highly related to the reliability of software artefacts. Lundestad and Hommels outline vulnerabilities associated with software systems and identify several social and organisational factors (e.g. job satisfaction) that affect software developers and contribute to these vulnerabilities [91]. The authors conclude their article by stressing the need for "a greater awareness of the implications of software flaws".

Miller, for example, advocates the use of informed consent, a concept borrowed from medical ethics, in software [92]. Miller argues that both the seller and consumer of software will benefit from consent, the former will gain trust from the latter, and the latter will obtain "detailed information about the safety and reliability of software."

Black advocates for the use of static analysis as part of ethical software development and reviews several tools using the Static Analysis Tool Exposition, which aims to "create a better understanding of static-analysis tools, encourage their improvement, and increase their use" [93].

Gotterbarn proposes a Software Development Impact Statement which expands on the concept of software risks (including ethical risks) that lead to software failure [94].

JUDGMENT

The judgment of software engineers, or requirements engineers, can have far-reaching consequences.

Therefore the Code urges software engineers to maintain integrity and independence in the process of decision making. In this literature review, a large number of articles discussed ethical discussion making, however, a lot of them were allocated to the Public principle since the emphasis lied on the societal impact of these judgments. In this section 2 studies that are related to software engineers' judgment are briefly discussed.

Decision-Making
Van den Hoven sketches in his paper the historical background and future program of value sensitive design in IT [95]. Value sensitive design studies "the ways in which our accepted moral values can be operationalised and incorporated in IT design."

Järvinen discusses in his paper the problem of unethical code and the problems raised by increased intelligence [96]. The author states with regards to recognising whether a decision has ethical consequences or not that "particularly in software, where the same piece of code can be reused again and again, and the software is produced as components, the decisions should be identified on the early phases of development." A statement that falls in line with the credo of this thesis.

MANAGEMENT

The ACM/IEEE-CS Code of Ethics states that "software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance", which brings us to the principle of Management. The encountered ethical issues concerning this principle focus on managing the practices of requirements engineering, software engineering or software development.

Manager's behaviour
Sutling et al. conducted a systematic literature review to develop an understanding of the agile project manager's behaviour [97]. One of the so-called "behaviors" that the manager should adopt is ethics. The agile manager should have or strive for the following characteristics: honesty, respect for the stakeholder and fairness.

Focusing on the organisational point of view, Carlise argues that the consequences of software failure to the organisation can be far-reaching [98]. The author states that current practices and policies regarding software are not ethical. This unethical use of software is derived from the firm's unfounded trust and acceptance of software-based systems and that this trust emerges from "management's lack of knowledge about software and desire to maintain a relationship with software similar to relationships it has with other complex technologies." Another study that researched the organisational point of view is the study by Athavale and Singh wherein propose a Human Behavior Change model that aims to spread ethics that are part of the organisational values throughout the organisation [99].

(Virtual) Workspace
In contrary to the management of the physical workspace, current developments resulted in the presence of virtual workspaces. Gheni and others conducted a literature review to identify the main challenges of organisations in managing so-called "virtual teams", global teams that interact with each user using technologies [100]. One of the challenges is ethical leadership, for example by responding quickly to unethical workspace behaviour that may arise from the lack of face-to-face interactions.

Another study that touches the issue of ethical issues in virtual teams is the study conducted by Cafer en Misra where they advocate simplicity and also demonstrate how cognitive and ethical issues affect leadership qualities, especially in virtual team leadership [101]. Furthermore, the authors argue against the prevailing notion among software specialists that the more complex a project manager thinks, the better his leadership is.

Ethical issues with regards to management can also emerge from the application of tools. Radevski et al. propose Emendo, a conceptual system for continuous monitoring of developers' neural state that aims to improve software developers' productivity and discuss its ethical implications as well [102]. The authors advocate an utterly transparent integration of this system by stating that the employees "should be the ones to decide whether to use the system."

Ownership

Another ethical issue that emerges from software is the issue of ownership. The Code states that one should ensure that there is a fair agreement concerning ownership of any software. Focusing on software ownership, Van der Graaf gives us an insight into the boundaries of digital ethics in the context of the logic of participation and sharing within and across institutional and platform boundaries [103].

PROFESSION

One of the goals of the ACM/IEEE-CS Code of Ethics was to establish the profession of software engineering, which makes the principle that focuses on the profession itself a significant one. The Code states with regards to this principle that software engineers need to progress the integrity and reputation of the software engineering profession consistent with the public interest. Similar to the software engineer, the requirements engineer has to be aware of the possible implications of one's work. In this section, we present ethical issues that are related to the professional's responsibility.

Responsibility

A considerable amount of encountered articles discussed the topic of responsibility. Responsibility, as put by the ACM/IEEE-CS Code of Ethics, is taking responsibility for "detecting, correcting, and reporting errors in software and associated documents on which they work." Spior and Ward discuss the ethical responsibility of the software developer [104]. They wonder out loud whether we should accept an imperfect software world and whether users will "accept the inevitability of flawed software without seeking recourse, even in cases of devastating consequences?"

Becker-Kornstaedt focuses on the responsibility of process engineers and describes some cases in which the engineers have encountered ethical dilemmas while collecting available process information [105]. Additionally, the author provides techniques and guidelines that may assist the engineer in managing concrete situations where sensitive information needs to be handled. Responsibility is also an item within the paradigm of medical informatics as discussed by Gell [106]. The author, therefore, proposes an adapted version of the ACM/IEEE-CS Code of Ethics from the perspective of a medical informatician.

In addition to Code of Ethics, several studies focused on other ways to establish the discipline of software professionals. Laurie discusses the certification and licensing for software professionals and organisations; the author outlines the current status of efforts to introduce these concepts into the computing field [107]. Thompson shares the goal of establishing a discipline and evaluates the developments in Software Engineering that attempt to design a Body of Knowledge for the discipline (e.g. the ACM/IEEE-CS Code of Ethics) [108].

Lurie and Mark take a different route and propose an ethical framework, Ethical-Driven Software Development (EDSD), for software engineers that connects the ethical responsibility of software developers to their professional standards [109]. This connection can overcome the division between professional skills and ethical skills and provides an alternative to the stand-alone engineering code of ethics.

In addition to the working professional, organisations have a responsibility as well. Vartiainen et al. explored so-called grey area phenomena in Information Systems and Software Development (ISSD), i.e. immoral behaviours of individuals and organisations [110]. Based on their findings (e.g. unethical data collection and lying), the authors put forward a research agenda for studying this phenomenon.

Code of Ethics

In 1999 Gotterbarn et al. present the Software Engineering Code of Ethics, or ACM/IEEE-CS Code of Ethics, as a product of a joint taskforce of the IEEE Computer Society and the ACM [6]. This Code of Ethics is intended as a guide for members of the evolving software engineering profession. In the same year, Gotterbarn published an article in which he outlines how the "new" Code of Ethics affects the software engineering professional [7]. Furthermore, in the same year, the same author reported on how the Code of Ethics came into existence [111]. In 2009 Gotterbarn and Miller published an article in which they demonstrate how to use the Code of Ethics by going through several examples [112].

Aiken et al. discuss the usage of Codes of Conduct to resolve legal disputes by going through a fictional case study using the ACM/IEEE-CS Code of Conduct as a reference [113]. The authors conclude the article by stating that "an organisation's ability to evaluate conduct and, when appropriate, consider potential legal matters more knowledgeably is paramount to imposing accountability on all participants." Another fictional process is discussed by Takanen et al. [114]. The authors discuss a fictional software vulnerability process focusing on the acute handling of discovered vulnerabilities, reporting, correcting and disclosing these vulnerabilities which are all related to the responsibility of the working professional

Karim et al. review the existing work in the field of software engineering and the practice of the ACM/IEEE-CS Code of Ethics [115]. The authors argue that the Code is difficult to implement and is in need of "a proper mechanism and framework." Therefore, authors propose a framework that maps the principles of the ethical that are included in the Code to the phases of the Software Engineering Development Life Cycle.

Thompson provides a summary of the developments in the field of software engineering education and professionalism [116]. One of the focusing points of the study is the project concerned with the ACM/IEEE-CS Code of Ethics and Professional Practice.

The ACM/IEEE-CS Code of Ethics is not the only Code of Ethics that is published or used. Some studies focused on other codes or propose one themselves. Payne and Landry did the latter and proposed a single code of ethics that covers principles applicable to both business and IT professionals [117]. According to the authors, the following seven values comprise any code of ethics: consistency, respect for individuals, autonomy for all, integrity, justice, utility, and competence. Simonette et al. focus on the IEEE Code of Ethics and propose an extension of the SEMAT kernel to deal with this code [118].

Lack or Limitation of Codes of Ethics
However, there were also studies that focused on the lack of ethical codes. Ochara et al. argue that, among others, there is a lack of a moral and ethical code in requirements engineering, which results in a decision-making deficiency [14]. This lack of moral and ethical code enables the concept of "Belief in Inherent Morality" which translates to the belief of organisational members in "the rightness of their cause and therefore ignore the ethical or moral consequences of their decisions."

Burmeister and Weckert discuss the lack of a professional code of usability engineers; therefore the authors apply the ACM/IEEE-CS Code of Ethics to usability cases to determine its usability for usability engineering [119]. Münzel evaluates existing codes from professional institutions regarding their relevance to sustainable development to advocates the need for green software engineering [120].

Mason and Gallagher argue that there are four factors that stress the need for an ethical dimension in managing technology, namely the pace of technical change, the malleability of technology, the invisibility of technical operations, and the potential irreversibility of technical impacts [121]. the authors further argue that current codes of ethics are limited, due to the "dynamism of the professional roles and situations" and the "increasing ambiguity, uncertainty, and complexity of decisions."

Oriogun et al. argue that Codes of Ethics, including the ACM/IEEE-CS Code of Ethics, are too general and therefore press for a more specific approach of guiding software engineers in ethical decision making [9]. The authors explain that standards should be created for specific software categories (e.g. social networking), similar to the International Standard Organization, except guided by ethics.

Other studies concentrate on ethical guidelines that focus on specific methodologies or processes within software engineering. Judy attempts, from an Agile practitioner's perspective, to compare and contrast Agile Principles with other approaches to Software Ethics (e.g. ACM/IEEE-CS Code of Ethics) [122]. The author concludes by stating that "the conversation on ethical dilemmas is largely absent from an Agile context where they do not directly affect business value or teams."

Rahanu et al. propose an adapted version of a theoretical framework developed by the US Content Subcommittee of the Impact CS Steering Committee that consists of moral and ethical concepts which can be used to identify moral issues with regards to Software Process Improvement (SPI), in order to raise awareness about the ethical considerations of SPI [123].

COLLEAGUES

Colleagues play a significant role in the daily work practices of requirements engineers and software engineers. Therefore, the Code urges software engineers to be fair and supportive of their colleagues. Unethical behaviour in the workplace of software organisations, discussed in the Management principle as well, can be seen as an ethical issue that emerges from the fields of requirements engineering, software engineering or software development. In contrary to the Management principle, in this section, we will present the ethical issues that occur on a horizontal dimension rather than a vertical one. More specifically, we will focus on the phenomenon of Intellectual Property. Concerning this issue, the Code states that software engineers should credit the work of others and refrain from taking undue credit.

Intellectual Property

Sojer et al. propose a theoretical framework that attempts to capture individuals' decision-making process to this unethical reuse of internet-accessible code [124]. The framework was tested with a survey among professional software developers, among other things, following conclusions were made: programmers are weighing the consequences of potential actions (potential punishment outweighs potential benefits), climate influences intention indirectly (peers' ethical standard). Reuse of code and plagiarism were also common on StackOverflow as discussed by Ahmed and Srivastava [125].[6]

Another study was conducted by Siponen and Vartiainen which comprised a literature analysis of approaches for and against the unauthorised copying of software in view of a theory of moral development [126]. The author's make four conclusions: the use of punishments and psychological means of manipulating people have been overvalued, present approaches that use a psychological way of control may violate the autonomy of the individual if used excessively, there is a need for approaches that focus on the higher stages of moral development, and no single approach covers all stages of moral development.

Focusing more on the ethical standards, Duda and Peters conducted a descriptive study intended to investigate whether software engineers comply with the existing ethical standards with regards to copying computer software code [127]. The authors conclude their study by stating that "it appears that software engineers inherently respect the work product and intellectual property rights of others". An et al. conducted a similar study where they investigated whether developers respect license terms when reusing code from Stack Overflow posts [128].

On a macro level, Kimppa examines the Intellectual Property Right (IPR) behaviour of countries, including IPRs with regards to software [129]. The author states that copying software has become much harder since the source code, in contrary to the object code, is being held as a trade secret.

EDUCATION

A significant part of the reviewed literature focused on the education of ethics to software engineering students. The ACM/IEEE-CS Code of Ethics states that "software engineers shall participate in lifelong learning regarding the practice of their profession and shall promote an ethical approach to the practice of the profession" as a description of the principle of Self. We took the liberty to rename this principle to

[6] An online community for developers to ask and answer (technical) questions, see: `www.stackoverflow.com` for more information.

Education of Ethics, which we deemed a more striking term for this literature review since an substantial number of studies discussed the incorporation of ethics in university curricula which has to do with educating oneself.

Teaching ethics
Sedlet presents an overview of a computer science course in ethics, social and legal issues [130]. The author argues that computer science students need to "to understand and appreciate the enormous potential impact of computers relative to society before they begin their professional careers." Towell focuses solely on ethics and describes a pilot study conducted to investigate the teaching of ethics in software engineering curricula by surveying 127 educators [131]. Amidst the topics that were considered most critical by the respondent were quality and testing, liability and risks in health and safety-critical environments, and unauthorised access and computer security. Furthermore, Towell and Thompson stressed the importance of teaching topics related to ethics within Software Engineering programmes by surveying software engineering professionals [132]. Among other things, 41% of the respondents said that the teaching of ethics in their curriculum was "largely ignored."

According to Oriogun and Ogunleye-Johnson there is a necessity of reviewing the inclusion of ethics teaching and assessment in the "educational training of computing students to guard against the perceived negative tendencies of computing" and to better prepare them for the ethical complexities that await them after graduation [9]. In addition to ethics, Wilk discusses the necessity of teaching a law course that encompasses legal aspects, ethical aspects and professional responsibility to raise students' awareness of ethics [133].

Concentrating on security, Vaughn Jr. et al. propose the incorporation of computer security into the software engineering and computer science programs [20]. The authors advocate that "it is not difficult to argue that we should be preparing our graduates to defend themselves and their employers from common threats that are known today and to help them understand the problem sufficiently".

However, modelling curricula for software engineering bring about challenges as well as argued by Bagert [134]. The author stated that one of the critical issues faced by software engineering in its establishment as a discipline is the formulation of a code of ethics and professional practice. This call was later answered by Gotterbarn and others as described under the *Profession* principle.

Reviewing the literature shows that there are several ways to meet this need for the teaching of ethics. Lutz et al. discuss a software engineering seminar that, among others, attempts to address software engineering ethics by making the students familiar with professional responsibilities and ethical demands of the discipline [135]. One of the intended learning outcomes is that a student will be able to identify the principles of the IEEE-CS Code of Ethics. In a similar attempt, Epstein describes an undergraduate course that serves as a basic introduction to software engineering, the course includes issues related to professional responsibilities, ethics and work culture [136].

Duley et al. present ideas and issues related to the design and implementation of a software engineering curriculum [137]. One of the curriculum objectives is to "describe the importance of ethical behaviour and professional practice by a Software Engineer." Liu et al. focus on the application of software namely, the Internet of Things (IoT). The authors propose a course dedicated to the IoT in which its ethical and environmental impact will be taught "in order to increase students' awareness of the potential complication of adopting IoT technology" [138].

Teaching apparatus
Several other studies proposed frameworks that attempt to support the education of ethics in the software paradigm. De Melo and De Sousa for example, propose a conceptual framework for analysing cyberethics education as "millennial Software Engineers must think and act ethically to deal with critical issues of the new century" [139]. Furthermore, the authors suggest how to integrate this framework

into the software engineering undergraduate curriculum. Another framework that focuses on the assessment of teaching professional ethics to IT engineering students is proposed by Minaño et al. [140]. The authors present a model of assessment that focuses on four dimensions of ethical competences, namely ethical sensitivity, ethical reasoning, ethical decision-making, and knowledge and adequate use of deontological codes. Magenheim and Schulte advocate the integration of social and ethical issues of Information Systems with technical subjects in computer science courses [141]. The authors, therefore, propose a learner-centred educational approach that considers these social and ethical issues as essential parts of the software development process.

Gehringer describes a website on ethics in computing and the methodology for creating it [142]. The author deems such a website necessary as he argues that "ethics in Computing is a fast-changing field" which makes it difficult for textbooks to keep up. Therefore, the author suggests that it would be valuable to have a website that keeps track of ethical issues in computing, as there are a "plethora" of ethical articles available on the web and to give students an in-depth look at several ethical issues.

In addition to supportive frameworks, some studies focused on developing apparatus that enable the teaching of ethics. Ali, for example, aims to increase the absorption of concepts by software engineering students using playful activities, for example by including pictures about software engineering ethics to teach the fundamental ethical concepts in the development and management of organisational software engineering [143].

Wang et al. took a different approach by proposing a "performance-oriented design model of the content & style of Software Engineering textbook for undergraduate software related majors" [144]. This textbook includes the software engineering Code of Ethics at the beginning, along with other universal principles of professional ethics. In addition to these two studies, Caltagirone et al. propose a Reconfigurable Attack-Defend Instructional Computing Laboratory (RADICL) for computer security education based on the need for additional instruction on the apparatus for systems attack, in addition to the more familiar concept of defence in computer security courses [145].

One of the aftereffects of teaching software engineering is plagiarism. To tackle this issue, Gibson proposes a policy for software reuse to overcome the problems that emerge in preventing, detecting, assessing and punishing project work that contains plagiarised material of students [146]. Moreover plagiarism in education, Niezgoda and Way propose a software tool entitled SNITCH that enables automated plagiarism detection in computer science and other science and engineering courses [147].

Lessons Learned

In addition to the proposals and framework, several other studies described experiences from teaching ethics in software curricula. Fleischman reports his experiences on teaching a course on ethical issues in computer science and computer engineering [148]. The central themes of the course are risks and liabilities associated with engineering and certifying safety-critical systems, information privacy, encryption and individual rights, computer abuse, hacking and cracking, intellectual property issues, the Internet, and issues of equity.

Maxim et al. describe their experiences teaching a computer science major course in which student-led discussions on the responsibilities of the computing professional, risk and liabilities and intellectual property took place [149]. Similar to Maxim and others, Zeid describes his experiences and states that ethical issues unique to the region (e.g. religious ethics) should be considered while developing a curriculum for Computer Science [150].

Trippe reports on experiences associated with the development of C++ programming courses and outlines the design, encouraging student learning of ethics, especially with regards to plagiarism [151]. Drake reports on a survey course designed to cover application programming topics suggested in ACM/IEEE Computing Curricula 1999 [152]. The course covers, among other things, ethics, and touches on the topic of ethical dilemma recognition. Furthermore, computer ethics are discussed and also methods to make a logical decision.

Carroll et al. present their experiences in a new standalone course that covers ethics, contemporary issues, life-long learning, and communications [153]. The students studied several topics (e.g. privacy, intellectual property) with regards to the ethical and social impact of computing technology as well as the responsibilities of engineers and computer scientists.

Dodig-Crnkovic present experiences from two ethics courses given at different universities, furthermore the author discusses the benefits of teaching professional ethics and sustainable development to engineering students [154]. Suri focuses on the field of requirements engineering and summarises lessons learned from developing and teaching a requirements engineering course in a software engineering degree program [155]. The author argues that students should be provided with opportunities to practice ethical reasoning skills to prepare them to face dilemmas in their future career.

Other studies focused on measuring the level of ethical awareness of students. Berry and Berenbach tested students in a computers-and-society course on their ethicality before and after learning ethics [156]. Results show that learning ethics had no effect; hence the subtitle of the article: "A Disturbing Experience." Furthermore, Von Konsky et al. conducted empirical research aiming to assess "self-perceived and actual ability" to apply the Australian Computer Society (ACS) Code of Ethics [157].

Sajeev and Crnkovic investigated how graduates from different cultural backgrounds would handle negative information that is critical to a project by capturing their approach to several scenario's [158]. The results show that the average graduate is highly likely to report bad news to the supervisor in one of the experimented cases, irrespective of their background or work experience. However, the graduates were uncommitted with regards to taking the matter further up the hierarchy if their supervisor discarded their report.

Chinn and Vandegrift conducted an empirical study to analyse student responses to an exercise used in computer ethics and a software engineering course to raise awareness about issues related to hiring (including the issues of responsibility and diversity) [159]. The results show that students consider values that they deem decisive to obtain a degree in computer science, also crucial in hiring in the software industry.

Using Case Studies to Teach Ethics

Many studies describe the use of case studies as a means to teach ethics or assess its state. Fledderman presents four ethics case studies covering topics (e.g. responsibility, software quality) that have been shown to interest electrical and computer engineering students [160]. The author concludes by stating that students "are more likely to be engaged when the case studies are related to the subject matter that they are studying in their other engineering classes." Georgiadou and Oriogun used a similar approach and report on how professional issues are introduced and assessed in a software engineering module at university level [46]. Students were posed with ethical dilemma through case studies, which they demonstrated to be able to resolve.

Bowyer discusses a real whistle-blowing case that can be used for teaching in, among other things, computer science majors, especially since "the codes of ethics of our professional societies require the professional to blow the whistle in certain circumstances" [161]. Clancy et al. conducted a case study on a course in electrical and computer engineering to "increase students' awareness of ethical issues in the workplace" [162]. A few statistically significant improvements in ethical awareness were found, for instance, exposure to the ACM/IEEE-CS Code of Ethics was useful to most students.

Origun et al. conducted a case study to review the state of teaching software engineering ethics [163]. The authors state that "it is pertinent to note the necessity to strengthen the teaching methods and approaches applied in teaching and assessing software engineering ethics in higher education." Furthermore, the authors conclude that using real-life scenarios of ethical implications encountered by software professionals would equip students with the needed skills to counter them.

RESEARCH

Quantitative analysis has shown us that a significant portion of the encountered ethical issues was related to the conduct of research in the software engineering field. The studies reviewed in this section will show us that the majority of the reviewed articles that focus on research, concentrate on the treatment of research participants. Since the Code does not provide us with a principle in which these studies can be framed, we deemed it necessary to create a different principle, hence this section of Research issues.

Treatment of Research Participants

As shown in the quantitative analysis, ethical issues can emerge from the conduct of research as well with most of them focusing on the treatment of the research participants. Yaman et al. examine by empirical means the ethical attitudes of software development practitioners concerning notifying users when involving them in experimentation [164]. Furthermore, Hall et al. surveyed current practice with ethical issues in empirical software engineering [165]. The outcome shows that considering ethics while researching with human participants is "accorded a fairly low priority" by the majority of the Heads of computing departments in the UK.

Storey et al. report on several ethical issues that they encountered while conducting an empirical experiment concerning web-based learning tools [166]. The authors discuss ethical issues concerning relationships among the research participants and experimenters, the research process itself (informed consent, minimisation of harm concerning merit, competence, and confidentiality) and unevaluated tool deployment.

Badampudi reviews research articles in Empirical Software Engineering - An International Journal for reporting of ethical considerations to improve accountability and trust [167]. The results show that two out of seven reviewed studies reported some ethical information in an implicit manner. However, aspects such as competence, comprehensibility, and vulnerability of the subjects were not discussed. Fotrousi et al. discuss ethical challenges that researchers may experience while collecting and using user feedback as a method to support requirements engineering by going over an example case [168]. The authors discuss informed consent, anonymity, engaging users to provide feedback, using data in research, informing the users about the results and storing user data.

Baltes et al. report on their experience with different employed sampling techniques in empirical software engineering and their ethical implications [169]. Furthermore, the authors present an existing ethical guideline and point to works from other research communities that may initiate a discussion in the software engineering research community. Focusing on increasing the realism of controlled experiments in industry, Moløkken-østvold explores the (ethical) challenges experienced by researchers in their attempt to achieve this [170]. After reviewing the literature, the author concludes that "there will not necessarily be ethical problems with increased realism, provided that the researchers respect the principles of informed consent, benefice, and confidentiality."

Lack of Ethics Consideration

Andrews et al. argue that this lack of ethics consideration is due to the limitation of ethical policies in empirical software engineering and can therefore not guarantee ethical behaviour. The authors conclude that "credible results and a strong discipline of empirical software engineering are based on mutual trust that everyone will behave ethically" [171]. Sieber draws the similar conclusion in his research where she introduces some of the types of risks and vulnerabilities involved in empirical software engineering using human participants [172]. The author argues that human participants who are treated ethically are more likely to be cooperative and to produce valid data.

USER-BEHAVIOUR

As mentioned before, one of the disadvantages of the Code in this research is that the Code focuses on solely software professionals, which raises limitations concerning mapping the encountered studies. This principle of User-Behaviour was added to map ethical issues that arise from the behaviour of the user, rather than the behaviour of the software professional. In this section, we will discuss ethical issues that are related to software piracy and open source software ethics.

Piracy

Several studies focused on analysing the decision of using pirated software. Yi et al. for example examine the normative and informational social influences that affect an individual's decision to buy or use pirated software [173]. In a similar study Giri et al. conducted an empirical study to explore the different factors that consumers consider before free downloading, pirating or buying software [174]. The results show that, among others, the price of software plays a significant role. The authors, therefore, advice organisations to "keep the price of the software low".

Seale et al. propose a model that aims to predict and explain incidents of software piracy by integrating previous research on software piracy [175]. The model indicates that that social norms, expertise required, gender, and computer usage (both home and at work), all have direct effects on self-reported piracy. Moores and Chang propose a model of ethical decision making to explain ethical decision making in software piracy [50]. The authors conclude by stating that the results suggest that software piracy, and perhaps other ethical problems in the field of Information Systems, could be addressed by developing an ethics program.

Moores et al. discuss whether legislative and educational outreach programs can reduce the level of software piracy [176]. The authors assume that is people are aware of the societal and economic implications of piracy, and that it is a crime, people will stop engaging in software piracy. Results show that this is indeed the case and that the fear of legal consequences impacts the most on attitude regarding software piracy.

Open Source Software Ethics

Grodzinsky et al. examine ethical issues that have been raised by open source software and its challenge to commercial software models [177]. The authors address autonomy, quality, and accountability in an attempt to explain the motivations of the OSS movement.

Spinello argues with regards to Open Source Software that governments should not tilt the playing field to open source programs, even if its proponents consider it as morally superior compared to proprietary code [178]. The author states that the market should decide the fate of open source software as it is "the best forum for determining how the complex software industry evolves."

Tourani et al. conducted an empirical study aiming to investigate the role, scope and influence of codes of conducts within the field of open source software projects [179]. The results show that the top codes of conducts were adopted by "hundreds to thousands of project" and all of them included the following five dimensions in one way or another: purpose, honourable behaviour, unacceptable behaviour, enforcement, and scope.

Free Software

Freedom Zero, as defined by the Free Software Definition (FSD), intends to protect the right of the user to deploy software in whatever fashion, towards whatever end, he or she sees fit. Chopra and Dexter argue that Freedom Zero needs to be upheld as it "serves to facilitate a broader debate about software's larger social significance" [180].

META-ETHICS

Similar to the previous section, the limitation of the Code forces us to introduce a new principle, the principle of Meta-ethics. Various studies discuss ethics in itself and how to approach ethical issues. This section focuses on these studies and presents the meta-ethical discourses.

Meta-ethics
Aycock et al. examine the ethics of computer security defences by considering the problem of malicious software [181]. The authors propose the concept of a "cosecure system", i.e. a system that by design enables both legitimate and malicious software to coexist on the same machine.

Stieb reacts to the claims that computer professionals should be held responsible for so-called undesirable events and that many computer disasters can be avoided by genuinely understanding responsibility [182]. The author argues that this position is overkill: "programmers and others who are only human ought to be defended against the overzealous." In fact, advocating this impossible ideal may be counterproductive as it may cause cynicism among the professionals.

Génova et al. examine and contrast two different schools of ethical thinking, namely deontologism and consequentialism, the authors argue that a moderate version of the former that takes both rules and consequences into consideration while assessing actions, should be addressed in the ethical education of software engineering [183].

Moor discusses the fundamental question of machine ethics "what is machine ethics, and how good can it be?" and presents the varieties of machine ethics [184]. The author urges to refrain from questioning whether machines can become full ethical agents, instead we should focus on developing limited explicit ethical agents.

Gotterbarn presents an outline of various types of responsibility avoidance by software developers [185]. Furthermore, the author introduces the concept of positive responsibility which, in contrary to negative responsibility, emphasises on "the virtue of having or being obliged to have regard for the consequences of his or her actions on others."

Stahl proposes the concept of reflective responsibility to address the ethical and moral problems of risk in software and information system development [186]. The author further explains that reflective responsibility is based on the idea that "responsibility ascriptions should be an expression of responsibility."

5.4 Threats to Validity

The results of this systematic literature review may have been affected by several factors. In this section the potential threats to validity are discussed, these threats should be kept in mind while considering the results of this review. Furthermore, we also discuss the steps taken to mitigate these threats.

The selection process of the relevant studies was a rigorous process based on the Kitchenham guidelines [23]. Concerning the completeness of the search process, there might have been studies that were not retrieved from the electronic databases even though they should have been considered. One reason is that only two databases were considered as sources for the relevant articles. Trial searches proved that the selected databases were the ones that returned the most relevant literature. However, discarding other databases may result in missing several relevant articles.

Another possible limitation is that we have applied the search strings only to the title, keywords and abstract which means that relevant studies that do not include our search terms in the title, keywords or abstract may have been ignored. The reason why we decided to search only in title, abstract and keywords is that we deemed it unlikely that a relevant article would not have included our specified search terms in these places. Moreover, we did not include the terms "software" and "computer ethics" in our search string as it would have resulted in a high number of false-positive results, i.e. an abundance of irrelevant literature. We aimed to overcome incompleteness of the obtained studies by conducting several trial searches, conducted by two persons. Furthermore, it is also possible that we have missed ethical issues by excluding non-peer-reviewed literature; however, we aim to mitigate this by conducting a review of the grey literature.

Respectively 1, and 2 supervisors validated the review protocol and data extraction form. However, the data extraction was performed by one researcher only, where the Guidelines state that data extraction should, whenever possible, be executed independently by two or more researchers [23]. Furthermore, the actual assembly of ethical issues through the data extraction form may have been subject to interpretation. The ethical issue of the teaching of ethics, for example, is based on the implicit notion of the need for ethics education. If an author argues that computing curricula should comprise an ethical dimension, ipso facto, there exists a need for such an ethical dimension, which we defined as an ethical issue in itself, i.e. teaching of ethics. The same applies to the classification of the ethical issues following the ACM/IEEE-CS Code of Ethics.

No taxonomy was found with regards to ethical issues in software-related fields; therefore the authors decided to make use of the ACM/IEEE-CS Code of Ethics, which is limited in a sense that it only covers ethics from a software professional point of view. We aimed to overcome this problem by modifying the Code to the needs of the acquired ethical issues. The actual process of categorisation of the studies and the ethical issues discussed was also subject to the interpretation of the authors of this study, especially since the categorisation was carried out by a single researcher.

Furthermore, this review did not include the strict evaluation of the quality of the relevant literature as described in the previously mentioned guidelines. The reason why we refrained from appraising the selected studies on their quality is due to the nature of ethical issues. Ethical issues are often perceptions or normalised principles that are not always accompanied by foundations in the relevant studies. Therefore, quality appraisal is challenging to achieve, and we deemed it out of scope for this review.

5.5 Conclusion

We have conducted a systematic review of relevant literature that provided evidence to answer sub-research question 1. The purpose of this study is to provide an overview of the literature in the field, the ethical issues in this domain, and to identify uncovered areas in research. Furthermore, exploring the context of the ethical issues was also a research focus of this study.

We identified 680 studies from searching through the databases, of which 150 studies proved to be relevant to SRQ1. The relevant articles were analysed for evidence to answer the research question. These answers were interpreted using a predefined data extraction form that enabled us to extract the evidence rigorously and systematically. Related to the research question we have identified 98 different ethical issues.We have presented our findings in two parts. The first part consists of a quantitative analysis of the relevant studies and the ethical issues culminating in the comprehensive list of ethical issues as presented in Appendix B. The second part of our findings is presented in a narrative way intending to provide context about the relevant studies wherein the ethical issues were discussed.

One of the conclusions is that there is an increase in the number of publications visible in the research with regards to ethical issues in the domain of requirements engineering, software engineering, and software development from 2013 to 2017, with 2017 as the year with the most frequent published relevant articles. This increase, as shown in Figure 5 suggests that there is an increasing interest in this field of work.

Our results show that the most discussed ethical issue is the issue of responsibility of the (future) software professional. Taking the ethical implications of their work into account is deemed necessary by the authors of the majority of the studies. Looking at the ethical issue that was cited the most we can see that software piracy comes out on top, which is notable since it does not occur in the top-5 most common ethical issues (Figure 6), nor does it appear in the top-3 of most frequent ethical issues in all considered years (Figure 7).

Another conclusion is that our review has revealed that contextual factors may characterise ethical issues. The issue of autonomy in ethical decision-making, for example, can be related to autonomous systems or to software professionals (or both, since the latter defines the former).

As argued throughout this thesis, requirements engineering is an essential domain of software engineering and software development since much of the decisions of the involved parties can be traced back to that process. Therefore, raising ethical awareness in this process is essential. However, the domain of requirements engineering was not well-represented among the relevant studies. Only 7 out of 150 relevant studies were directly related to requirements engineering.

Another interesting observation is that none of the relevant studies discussed how to, or provided tooling to, teach ethics to the software professional. A large part of the studies focuses on teaching ethics to future software engineers (i.e. students) while neglecting the software professionals that weren't taught the ethical dimension of their profession.

We believe that the results of this systematic literature review can be useful for requirements engineering and software engineering professionals and researchers. The exhaustive list of ethical issues as presented in Appendix B can be used as a basis for future research or act as a pool for codes of ethics.

6 Grey Literature Review: Ethical Issues in the News

6.1 Introduction

This section presents the grey literature review (GLR) that will be conducted to answer the second sub-research question as stated in Chapter 2.2. This review aims to outline ethical issues that are related to requirements engineering to raise awareness about the ethical side of software artefacts.

The methodology that will be used to explore ethical issues in grey literature is the methodology proposed by Garousi et al. and is in general terms similar to the protocol used for the systematic literature review in the previous section [24]. Detailed features of this methodology were disclosed in Chapter 4. This review aims to outline all ethical issues related to requirements by influential news outlets to capture the public opinion.

This section is organised as follows. We first present the Research Methodology which comprises the Search Strategy, the selection process of the articles and the Data Extraction. We then conclude with the Study Results and the Conclusions.

6.2 Research Methodology

Grey literature reviews are often seen as a part of a so-called multivocal literature review. A multivocal literature review, or MLR, is a literature review that comprises both a SLR and a GLR [41]. Grey literature uses non-academic sources (e.g. news articles, blogs, videos, and white papers) as well. Capturing people's opinions and topical reports will provide us with a comprehensive view of the public perception concerning ethical issues related to requirements engineering. In this review, we focus on the so-called second-tier grey literature (e.g. news articles) for the reasons stated above [24]. We aim to extract ethical issues related to requirements engineering as reported by news outlets. This review process consists of 5 stages: search process, source selection, quality assessment of sources, data extraction, and data synthesis.

6.2.1 Planning the Review

Similar to the systematic literature review, we need to identify the need for the grey literature review. In fact, the rationale behind this review is relatively similar to the motivation of the SLR. The motivation was discussed throughout this research by arguing that requirements engineering is a crucial stage in creating software; hence we are of the opinion that raising awareness in this process will turn out to be a profitable undertaking.

This research is premised on the need for ethical awareness in requirements engineering and the lack of scientific literature that discusses or presents ethical awareness or ethical issues related to requirements engineering. In addition to the SLR, GLR can broaden our perspective of the ethical issues in the domain of requirements engineering and identify a possible gap between ethical issues discussed in science and those who are reported upon in the news.

6.2.2 Research Question

The goal of this review is to outline the ethical issues in requirements engineering. Ethical issues related to requirements engineering, as defined in Definition 4.1.5, will be explored to assess the state of ethics in requirements engineering. Sub-research question 2 of this research proposal (Chapter 3) forms the core of this grey literature review:

RQ "*What does grey literature state about ethical issues related to requirements engineering?*"

Rationale The purpose of this question is to explore ethical issues in or related to requirements engineering. As aforesaid, requirements engineering involves interaction with several stakeholders. This research aims to outline the common ethical issues and how they relate to each other. Subsequently, this can lead us to direct future research. Furthermore, the set of collected ethical issues can provide the involved parties in the requirements engineering process with a better understanding of the possible ethical impact of their process.

6.2.3 Conducting the Review

After the evaluation and refinement of the protocol, the reviewing process was initiated. The first step in conducting the review is acquiring the articles that need to be reviewed. These articles are searched for using the search strategy that is defined in Chapter 6.2.4. The search process often results in a high number of irrelevant articles; therefore a selection procedure is deemed necessary.

6.2.4 Data Sources and Search Strategy

The search strategy is one of the most influential parts of literature reviews since it defines the way articles will be searched for. Similar to the SLR, we focused on digital sources for this review. The Google search engine was used to identify relevant articles. The SLR provided us with a set of keywords that formed the basis for the search in this review. However, it had to be somewhat adapted since trial searches proved that the returned results had mainly to do with peer-reviewed studies and university curricula, while we focus on opinion pieces and news reports.

Since this GLR aims to identify ethical issues from news reports and opinion pieces, we determined to restrict our search to a predefined set of online news outlets. The selection of the news outlets was based on their ranking concerning popularity (4IMN, Alexa Ranking) and geographical coverage, intending to cover large parts of the world [187, 188]. The following all-English news outlets were explored: Al Jazeera English, China Daily, Independent Online [7], New York Daily News, The Guardian, The Sydney Morning Herald, and USA Today.

Al Jazeera English is a Doha based news outlet covering news from all over the world but was selected due to its intensive coverage of the Middle East, Asia Pacific, and Africa [189]. China Daily is an English daily newspaper published in the People's Republic of China [190]. Independent Online is a news and information website based in South Africa and was, according to the 4IMN ranking, the most popular newspaper on the African continent in 2016 [187, 191]. New York Daily News, a New York-based newspaper, and USA Today, were selected for their coverage of the United States [192, 193]. The Guardian, a British daily newspaper, was selected for both its international as its UK coverage. Lastly, The Sydney Morning Herald, an Australia based daily newspaper, was selected for its coverage of Australia and New Zealand [194]. The selected outlets publish, among others, feature articles, opinion pieces, interviews, and regular news reports. We aim to use these articles to capture ethical issues from public opinion. To identify these articles, news outlet-dependent search queries were formed using the SLR search query as a basis. The following query was the result of several trial searches:

site:<homepage-of-newsoutlet> (((**ethics** *OR* **ethical** *OR* **moral** *OR* **good** *OR* **bad** *OR* **harmful**) *AND* (**software** *OR* **computing**))

The search query consists of two parts separated by the logical AND operator. The first part consists of words that aim to capture articles related to ethics. The second part aims to identify the domain in which we want to explore ethics. Since news articles are less formal than academic literature, we have refrained from the use of technical terms and used the more general terms of "software" and "computing" instead.

In our search strategy we adopt the so-called effort bounded stopping criteria technique, i.e. we include

[7] Based in South Africa, not to be confused with the UK based online newspaper The Independent.

the top N search engine hits [24]. The reason why we preferred this option to theoretical saturation (i.e. stopping when no new evidence emerges) and evidence exhaustion (i.e. extracting all available evidence) has to do with time restrictions. Moreover, this review includes the top 50 results per newspaper as decided by the Google PageRank. For each time the search query was executed, the first 50 hits were opened in new tabs, and the search was saved using Session Manager [8], a Google Chrome plugin that facilitates saving the state of the web-browser future use. We deemed this necessary to preserve the same set of returned articles of the executed search query.

6.2.5 Selection Process

After defining the search strategy, the relevant articles need to be selected. The article selection consisted of one shift, see Figure 9. The query and the selected sources provided us with 350 articles. All the articles were read in full and held against the inclusion and exclusion criteria as presented in Chapter 6.2.6. The relevant articles withstood this shift (126) were then subjected to the data extraction process of Chapter 6.2.7 which consists of the extraction of the ethical issues from the selected articles.

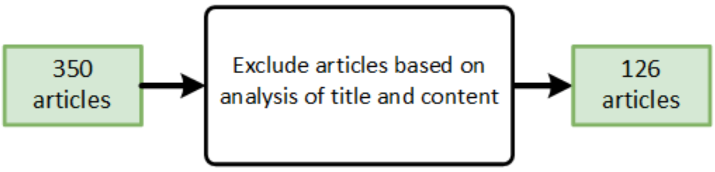

Figure 9: The article selection process of the grey literature.

6.2.6 Inclusion and Exclusion Criteria

After obtaining the potentially relevant articles, the content of these selected articles needs to be assessed for their actual relevance to the Research Question. To assess this relevance, the articles were held against the Inclusion and Exclusion Criteria of Table 8. These criteria were designed to determine whether the obtained articles are relevant or not. If the article in question was considered relevant, it was processed further in the data extraction phase. However, if an article was deemed irrelevant, it was discarded.

Table 8: Inclusion and exclusion criteria of the grey literature selection process.

Inclusion criteria	Exclusion criteria
I1. Articles in the form of opinion pieces, columns, news reports, feature articles, or interviews	**E1.** Articles that are not opinion pieces, columns, news reports, feature articles, or interviews
I2. Articles that focus on one or more ethical issues that emerge from software	**E2.** Articles that do not focus on ethics in software

The first inclusion criterion was designed to only consider articles in the form of opinion pieces, columns, news reports, feature articles, or interviews to be selected. Trial searches showed that some websites, for example, published articles that compared software products or answered garden-variety user questions about software, hence Exclusion Criterion 1. The second Inclusion Criterion aims to select only the relevant studies, i.e. the studies that discuss ethical issues related to software. It is worth noting that "software" is used instead of the more technical terms of requirements engineering, software engineering, and software development. As explained along with the search query in Chapter 6.2.4, the trial searches proved that the news outlets did not use such technical terms. The second Exclusion Criterion filters out the articles that do not focus on ethics in software.

[8] For more information about Session Manager see: `https://chrome.google.com/webstore/detail/session-manager/bbcnbpafconjjigibnhbfmmgdbbkcjfi?hl=en`

Table 9: Selected articles per news outlet.

Included	**Number of Articles**
Al Jazeera	26
China Daily	11
Independent Online	15
NY Daily News	15
The Guardian	24
The Sydney Morning Herald	20
USA Today	15
Excluded	**218**

Table 9 shows the outcome of the article selection process. 126 Articles were considered to be relevant according to the inclusion and exclusion criteria. Al Jazeera and The Guardian supplied the most number of relevant articles, respectively 41 and 40 articles. 29 Articles were selected from The Sydney Morning Herald. USA Today, NY Daily News, and Independent Online contributed with 15 articles each. The news outlet that provided us with the least amount of relevant articles was China Daily with 11 articles.

6.2.7 Data Extraction

To answer SRQ2, evidence needs to be gathered. To do so, a data extraction form was designed to capture ethical issues related to software from the relevant articles. The data extraction form is presented in Appendix C.

The Data Extraction Form for this review is similar to the Data Extraction Form of the SLR, except that it does not include the research features. The Data Extraction Form consists of 2 main parts. The first part focuses on the extraction of meta-data and captures among other things the title, publication date, the news outlet, and the URL of the article. The second part focuses on capturing the ethical issues and awareness. Before starting with the extraction process, a dry run was conducting upon ten randomly selected articles to validate the data extraction form. The extracted data was eventually stored in a Microsoft Excel table using the items of the data extraction form as column headers.

With regards to capturing the type of the article, three distinctions were made: news reports, feature articles, and opinion pieces [195]. News reports are articles that aim to inform the reader about topical happenings in the world. Feature articles are often longer than news articles and explore news stories in more depth. Such an article often aims to explore and analyse the phenomenon in question. Opinion pieces (e.g. columns) are articles written by professional writers who are often experts on the topic and aim to inform the reader about a topic.

Akin to the Data Extraction form of the SLR, the most valuable item that ought to be captured through this form in the GLR are the ethical issues. Since the ethical issues might be externalised by the different authors of the selected articles in different ways, we took the liberty to name them in a way that facilitates categorisation and therefore analysis.[9]

[9] The filled-in data extraction form for all relevant studies is accessible here: `https://github.com/Aberkane/ethics-in-re/tree/master/RQ1/SRQ2`.

6.3 Study Results

This section presents the results of the grey literature review following the guidelines of Garousi and others [24]. The results are entirely quantitative.

In Figure 10 the distribution of the article types is presented. As explained in Chapter 6.2.7 3 distinctions were made: news reports, feature articles, and opinion pieces. The results show that the majority of the selected articles were categorised as feature articles. Second in place come regular news reports comprising exactly one-third of the selected articles. Lastly, opinion pieces made up slightly more than 10% of the selected articles.

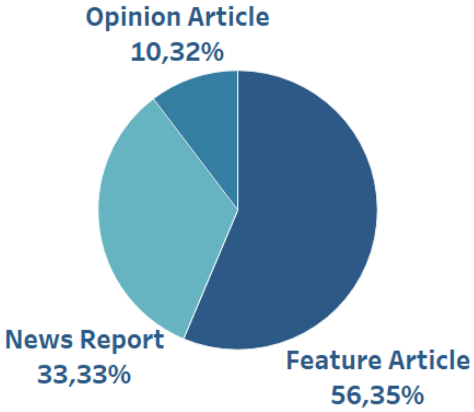

Figure 10: Distribution of article type.

Figure 11 presents the distribution of the selected studies from 1999 up to and including 27 April 2018. The graph shows that news outlets began to publish on the topic of ethical issues related to software in 2004. From 2013 on, two large spikes are apparent in 2015 and 2017. In the first two months of 2018, 11 articles were published. The graph shows a significant increase in recurrence of publishing over years which suggests a rise in interest or the rise of actualities concerning ethical issues related to software

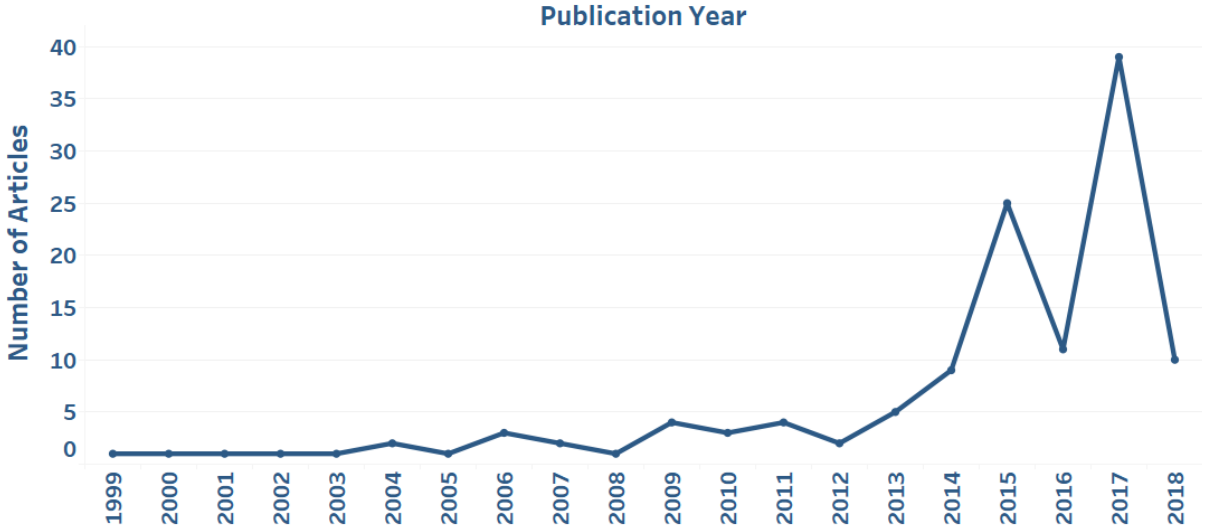

Figure 11: Distribution of selected articles over time period.

Figure 12 presents the 25 most frequent occurring ethical issues. Security was the ethical issue that occurred the most in the selected articles, appearing 47 times. Second in place comes the ethical issue of Privacy which occurred 32 times. The ethical issue of Piracy completes the top-three most common ethical issues counting eight occurrences in the articles. Interestingly, the difference in frequency between the top-two most common ethical issues and the rest of the issues is significant. Piracy is followed up by the issues of Responsibility and Virtual Harm, both counting seven occurrences. The ethical issue of Cybercrime and Ethical Decision-Making occurred six times each. The issues of Data Collection, Flawed Software, Green Software, Job Loss and Transparency all count four occurrences each. The issues of Diversity, Hacking, Lying, Net Neutrality were captured three times each. Accountability, Code of Ethics, Data Monitoring, Disappearance of Tradition, Discrimination, Economic Inequality, Lack of

Awareness, Software Quality, and Vulnerability occurred two times each.

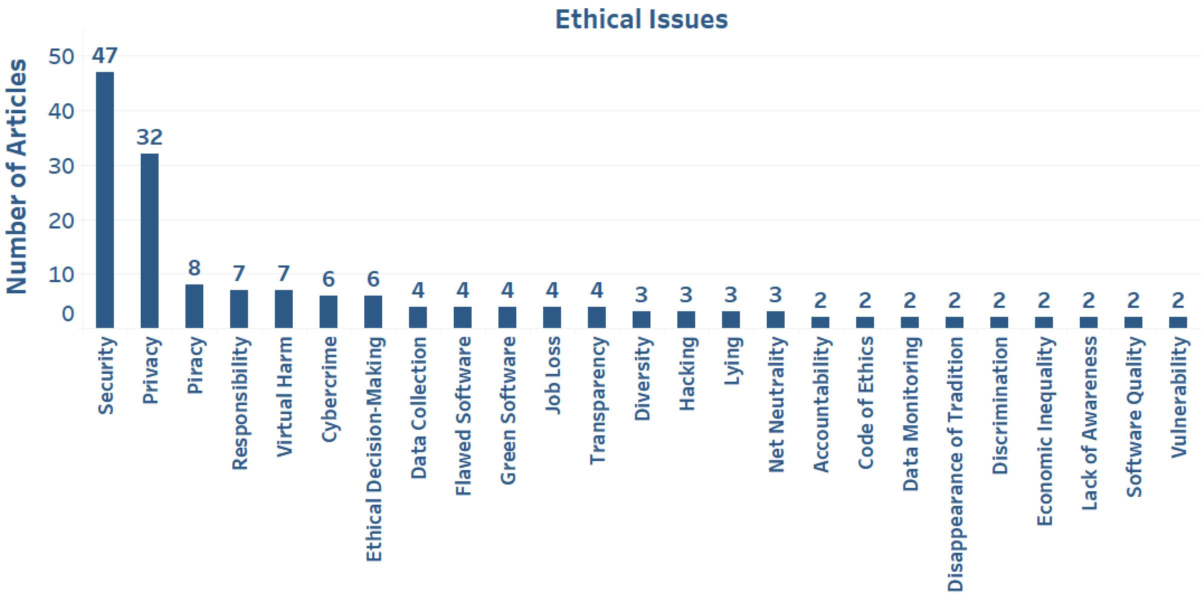

Figure 12: Top 25 most occurring ethical issues.

Figure 11 showed that the rise of articles related to ethics started in 2013. In Figure 13 we focus on this time-frame and present the three most frequent issues per year from 2013 to 2018. In 2013, Privacy was the most frequent ethical issue, followed up by Security and Green Software. In 2014, the year before the first optimum, Security surpasses the issue of Privacy as the most frequently occurring issue. The issue of Data Monitoring completes the top-three. In 2015, the year of the first optimum, the issues of Security and Privacy extend their position as the two most occurring ethical issues, except now being accompanied by the ethical issue of Green Software, which may be caused by the Volkswagen emissions scandal that took place in the same year [49]. In 2016, the issue of Security and Privacy were, again, the most occurring ethical issues with the issue of Diversity completing the top-three. In 2017, several articles reported on the worldwide cyberattack by the WannaCry ransomware cryptoworm which is reflected by the top-three of the same year: Security, Privacy, and Cybercrime [196]. In the first four months of 2018, Privacy surpassed Security as the most frequent issue, Security comes second in place, and the issue of Accountability completed the top-three.

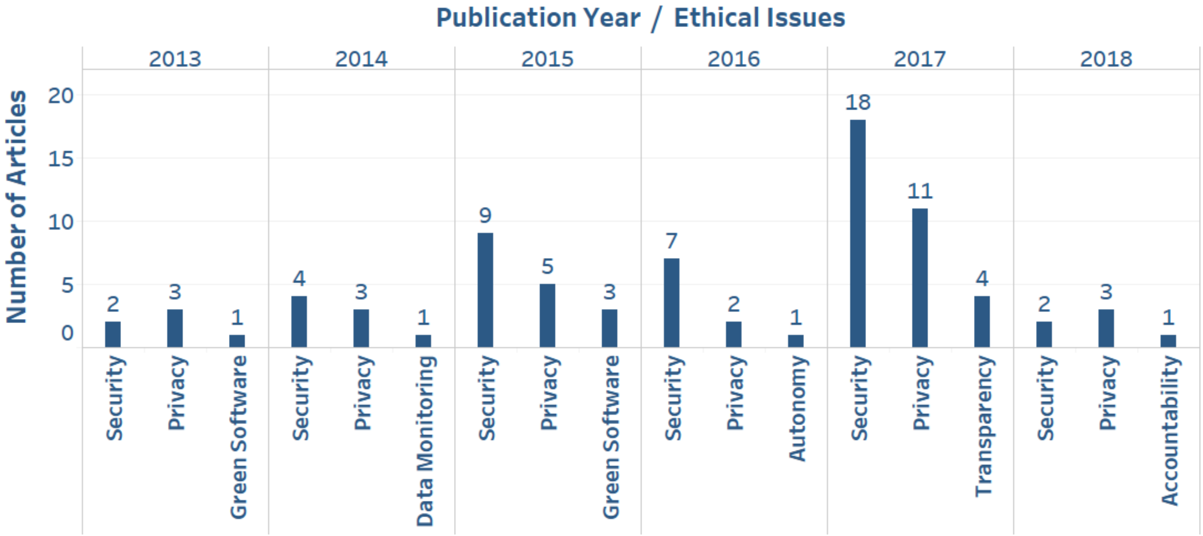

Figure 13: Most frequent ethical issues per year since 2013.

6.4 Threats to Validity

In this section, we will discuss the potential threats to validity that need to be taken into account while considering the results of the grey literature review. Moreover, measures taken to minimise these threats are discussed as well.

The grey literature review was based on the guidelines by Garousi et al. combined with parts from the systematic literature review of the previous section to increase the rigour of the review [41, 23]. The first potential limitation is the is the way we defined grey literature. Grey literature has discrepant definitions and the concept of grey literature reviews are relatively new to the software engineering field [24, 197]. We aimed to overcome this problem by abiding by the interpretation of Garousi et al. in which news reports are considered grey literature.

Another possible threat to validity is the selection of the news outlets and the search process. We intended to capture the public discourse concerning ethical issues related to requirements engineering by capturing ethical issues from news articles from different news sources. The sources were selected based on their popularity according to rankings of Alexa and 4IMN and were limited to 5 different newspapers. However, being a popular newspaper does not mean that the content is in line with the vox populi, which means that ethical issues that were discussed in less popular news outlets may not be considered and that the selected articles may be an unrepresentative sample of ethical issues.

Moreover, some of the popular news outlets (e.g. The Washington Times) that had to be paid for were not selected and were replaced by the next in rank. Another limitation is the fact that we focused only on news reports and articles written in English. Furthermore, we were subjected to the search algorithm of Google for obtaining the relevant articles. Also, the selected news outlets are only a small sample of the available news outlets that are present in the world, and the ethical issues may be restricted geographically. To mitigate this threat, we tried to use news outlets that are active in each continent to increase the completeness of the ethical issues. However, no news outlet was found that focused mainly on south-America. In an attempt to overcome this threat, two outlets with a wide range were chosen from the United States.

Finally, the relevant articles were not assessed on their quality. We did not conduct contextual research both intrinsic to the paper (e.g. fact check), and extrinsic (e.g. author background). This was considered to be out of the scope of this research. Furthermore, this review was conducted by one researcher only which may have resulted in a one-dimensional interpretation.

6.5 Conclusion

In this study, we have reviewed grey literature to analyse the public discourse about ethical issues related to requirements engineering. Together with the results of the systematic literature review, this work can form a solid basis for further research and provides us with a global view of the ethical issues in this domain as perceived by the public. A total number of 350 news articles were considered from 5 different sources from all over the world. Less than half, 126 articles, proved to be relevant to the research question. These articles were read in full and ethical issues were extracted from them. These ethical issues were captured through a predefined data extraction form. In total, we have identified 34 ethical issues from the relevant articles.

The captured data was analysed quantitatively, and the resulting list of all ethical issues can be found in Appendix D. This list of ethical issues is our answer to sub-research question 2. Further analysis of these issues shows that there is an increase of public discourse visible regarding ethical issues related to requirements engineering and software. Furthermore, the year 2017 was the year in which most of the relevant articles were captured (see Figure 11).

Our results show that the issues of Security, Privacy, and Piracy were the most reoccurring ethical issues. The first two issues are heavy-related since lack of security can result in a breach of privacy. Another interesting phenomenon was that the issue of Security was consistently present among the ethical issues that were the most frequent per year. Which shows that this issue is a constant issue in the course of the years.

6.5.1 Comparison between SLR and GLR results

In this section and the section before, literature was reviewed to outline the common ethical issues in the field of requirements engineering and software development as a whole. These reviews were conducted to answer the first research question: "*What are the ethical issues related to requirements engineering?*". In this section, we will briefly discuss the overlap and differences of the ethical issues in numerical terms.

Overlap

Respectively 97 and 34 ethical issues were obtained from the systematic literature review and grey literature review. Analyzing the two lists of ethical issues resulted in an overlap of the following 27 issues: Accountability, Autonomy, Code of Ethics, Data Collection, Data Monitoring, Digital Divide, Disappearance of Tradition, Discrimination, Diversity, Ethical Decision-Making, Green Software, Hacking, Intellectual Property, Job Loss, Lack of Awareness, Lying, Piracy, Plagiarism, Privacy, Responsibility, Safety, Security, Software Quality, Transparency, Virtual Harm, Vulnerabilities, and Workplace Ethics. This overlap increases the completeness of the ethical issues, especially the ones that were captured through the grey literature review since the vast majority of them is backed up by peer-reviewed academic literature (27 out of 34).

Interestingly enough, the temporal distribution of related articles of both the SLR as the GLR showed optima in 2015 and 2017. The most reoccurring issues in 2015 according to the GLR were Security, Privacy and Green Software. According to the SLR, the most frequent issues were Privacy, Responsibility and Code of Ethics. In 2017, Security, Privacy, and Transparency comprised the top-three of most frequent issues in the grey literature. In the SLR, the issues of Responsibility, Privacy, and Ethical Decision-Making formed the top-three. We were not able to find a direct relationship between the two spikes in both reviews.

Differences

In contrary to overlapping issues, there were also several issues that did not occur in both lists. Taking the grey ethical issues as a starting point, we see that the following seven issues were not discussed in the scholarly papers: Cybercrime, Economic Inequality, Fake News, Flawed Software, Freedom, Net Neutrality, and Wealth Bubble. From the SLR point of view, we more than two-thirds (68) of the ethical issues that were obtained from the systematic literature review were not mentioned in the grey literature. However, analysing this difference is considered out of the scope of this research.

Despite the limitations of this review, we believe that our research was useful and add an extra dimension of completeness to the ethical issues that were acquired through the systematic literature review. Furthermore, the acquired ethical issues (Appendix D) can be used for further research and as a means to raise awareness about the issue or as a basis for solutions.

7 Automatically Identifying Ethical Issues

In this section we will present several methods of feature extraction to answer the second research question as presented in Chapter 2.2: "*How to automatically identify ethical issues in requirements engineering?*" The goal of this section is to prove the applicability of acquired ethical issues of literature reviews and to raise ethical awareness among the requirements engineer and the engaged parties. We will first discuss the data set, and then we will proceed to lay out the tools of feature extraction with which we intend to identify meaningful features that can be used for machine learning.

7.1 Introduction

Software systems become increasingly present in our daily lives, considering their ethical impact is therefore essential [198]. In an attempt to diminish the potential unethical impact of software systems (e.g. breach of privacy), we approach this problem from a requirements engineering point of view. As reiterated throughout this study, requirements engineering is the process of discovering the intended objective of a software artefact by identifying its stakeholders, individuals or organisations who hold a stake in a system, and their needs [13]. To raise ethical awareness in this critical process, we identified several ethics issues that need to be considered while drawing up requirements. In this section, we conduct natural language processing and assess several features for their usability in future machine learning.

7.2 Rationale

Natural language is commonly used to express requirements in software development projects [26]. In this research, we aim to extract the underlying values of requirements to identify their potential ethical implications. By identifying valuable features for further research, we attempt to aid in the process of automatic identification of ethical issues in requirements that may have an unethical implication. The higher social goal is to raise awareness and consciousness among the requirement engineer and stakeholders concerning the compounded values of their requirements. In contrary to the approach by Detweiler and Harbers, we proceed from requirements to values and not the other way around [15]. In this research, we adhere to the definition of requirements engineering by Loucopoulos and Karakostas and consider the three main processes of RE: elicitation, specification, and validation [199]. The results of this research can be used to construct a baseline classifier to assist in the process of requirements elicitation by providing the involved parties with moral guidance.

It is worth mentioning that we initially attempted to build a classifier ourselves. However, for reasons that we will explain later on in this section, we have refrained from doing so. We will explain this later on in this section. It is therefore that we focused on feature extraction instead.

7.3 Methodology

The goal of this section is to explore and assess several text features on their usability in classifying requirements based on their ethicality. To do so, we have created a script that identifies and compares features from a real-world data set consisting of user stories.[10] Assessing these features on their usability for classifying the text fragments into one or more labels is commonly known as Text Categorisation within the field of supervised machine learning [200].

The proposed tool that facilitates feature extraction was built in Python using several machine learning

[10] The feature_extraction.py script is accessible on: `https://github.com/Aberkane/ethics-in-re/tree/master/RQ2`.

libraries. The aim is to identify a set of predefined ethical issues[11] in requirements using these features. The ethics labels are a subset of the 16 most frequent ethical issues that emerged from both the systematic literature review as the grey literature review. The approach that we took reads as follows: we started off with data collection, then the data was tagged manually, and finally we conducted feature extraction.

7.3.1 Environment

This tool is written in python version 2.7.3 on the Ubuntu operating system. Python was chosen due to its plethora of libraries that facilitate machine learning, data processing, feature extraction, and because the author of this study was familiar with that programming language. The libraries that were used are scikit-learn and the Natural Language Toolkit (NLTK). Scikit-learn is a library that provides simple and efficient tools for data analysis and is built upon the libraries of NumPy, SciPy, and matplotlib [201]. NLTK is a platform that provides, among other things, text processing libraries for classification, and tokenisation [202]. It is with these libraries that we aim to conduct feature extraction.

7.3.2 Classes

As mentioned in the introduction of this section, we aim to identify ethical issues in requirements engineering. These ethical issues are based on the outcomes of both the systematic and the grey literature review. However, not all obtained labels will be used. We will take subsets from both lists of ethical issues obtained from the SLR and GLR. The results of the systematic literature review have shown us in Chapter 5.3 that the top-15 most occurring ethical issues consist of the following: Responsibility, Privacy, Code of Ethics, Teaching Ethics, Intellectual Property, Ethical Decision-Making, Security, Informed Consent, Autonomy, and Quality. The results of the grey literature review (Chapter 6.3) showed us that the top-5 most reoccurring ethical issues consist of the issues of Security, Privacy, Piracy, Responsibility, Virtual Harm, and Cybercrime. Table 10 shows the final list of ethics labels after merging both lists from the SLR and GLR and removing the duplicates.

7.3.3 Data Collection

The next step was to collect requirements that could be used for feature extraction, which proved to be a challenging task. After approaching several companies without success, we redirected our focus to obtaining data from the internet. After attempting several unsuccessful search queries, we followed a lead by a fellow researcher who advised us to look for user stories on Github, a large software development platform.[12] After conducting several searches using the search functionality on the website, the following query proved to be the most successful: “user stories”. The resulting user stories were captured manually from 49 different repositories and collected in a Microsoft Excel sheet.[13] The final amount of obtained user stories through this means was 1009 user stories. After collecting the data, the next step was to label the data manually. However, since we attempt to conduct feature extraction, the data needs to be labelled first to compare the results between labelled and unlabelled data.

7.3.4 Manual Tagging

As mentioned before, the requirements will be classified into the following 11 labels: Teaching Ethics, Responsibility, Privacy, Code of Ethics, Intellectual Property, Ethical-Decision Making, Security, Informed Consent, Autonomy, Software Piracy, Transparency, and Virtual Harm. In order to conduct supervised machine learning, data needs to be classified beforehand in order to “train” the classifier. The data was labelled manually by one single researcher using Microsoft Excel. The final outcome of the data tagging is presented in Figure 14.

[11] Interchangeably used with “ethics labels” throughout this section.

[12] Github software development platform, accessible on: `https://github.com/`.

[13] The data was collected on 17 May 2018.

Table 10: Ethics labels used as a basis for identifying ethical issues in requirements engineering.

Identifier	Ethical issue	Description
EL1	Responsibility	*Ethical responsibility of software developers, software professionals, and the discipline itself*
EL2	Code of Ethics	*Code that comprises guiding ethical principles aiming to aid the software professional in ethical* decision making
EL3	Privacy	*Ethical handling of personal data*
EL4	Teaching Ethics	*Increase ethical awareness among professionals and future professionals*
EL5	Software Piracy	*Software Piracy*
EL6	Intellectual Property	*Intellectual property (rights) and ownership of computer software code*
EL7	Ethical Decision-Making	*Ethical decision-making of software engineers or users*
EL8	Security	*Ethics of computer, user, and data security*
EL9	Informed Consent	*Informed consent from subjects in experiment and software consumers*
EL10	Virtual Harm	*Exposure to harmful websites and harmful information*
EL11	Autonomy	*Autonomy of computational artefacts; Individual's freedom to make voluntary and informed choices*

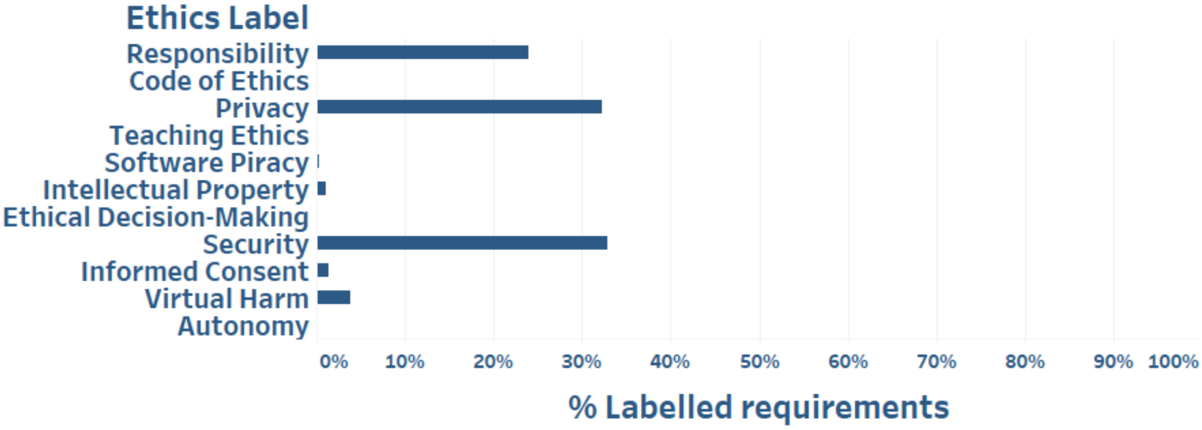

Figure 14: Distribution of the labelled data.

The results in the figure above show that the data set is highly imbalanced which may influence future machine learning processes negatively [203]. Out of the 12 labels, no less than nine labels were represented by less than 5% of the requirements. Furthermore, requirements related to the labels of Autonomy, Code of Ethics, Ethical Decision-Making and Teaching Ethics were not present at all. This may indicate that these particular labels are not suited for this process in requirements engineering. Teaching Ethics, for example, has more to do with the discipline of requirements engineering than the requirements itself. It is therefore that we decided to discard all labels that are not well represented in the data set, i.e. all labels that were represented by less than 20% of the requirements were removed. The labels of Autonomy, Code of Ethics, Ethical Decision-Making, Informed Consent, Intellectual Property, Piracy, Quality, Teaching Ethics, and Virtual Harm were removed, which means that from now on the focus lies on the issues of Responsibility, Privacy, and Security.

7.4 Feature Extraction

As stated in the Chapter 7.2, we intended to build a classifier to classify requirements based on their potential ethical implications. However, the imbalanced data resulted in a very accurate classifier. Requirements were classified into Responsibility, Privacy, and Security with a precision of respectively 93%, 89%, and 91%. At first glance, these results look promising, were it not for the fact that the classifier did not consider the minority class at all. According to Chawla et al., standard classifiers tend to ignore small classes [204]. In our case, this resulted in a classifier that assigned all requirements to the majority class. We attempted to overcome this problem by conducting several oversampling techniques such as SMOTE and but these attempts were to no avail [205, 204]. For these reasons, we directed our focus on feature extraction.

The process of feature extraction is a common way to extract interesting information in text classification [206]. In this study we will adopt a bag-of-words approach, i.e. the words of which a requirement is composed are seen as independent tokens. This approach is also known as the word frequency model and is the most popular text feature extraction model [207]. These features enable classification methods to act upon more focused elements of the text. In this section, we will experiment with several features and compare labelled and unlabelled requirements. The script that is used to conduct these experiments is available on the Github repository of the author of this research. [14]

7.4.1 Number of Words

The first feature that we will consider is the number of words for each requirement. In Table 11 we have laid out the average number of words for both labelled and unlabelled requirements. The results show that for all three labels, the mean of the labelled requirements was less than the mean of the requirements that were not labelled. This indicates that the labelled requirements consist of a lesser amount of words.

Table 11: Comparison of the number of words in labelled and unlabelled requirements.

Label	Mean labelled	Mean unlabelled
Responsibility	17.92	20.37
Privacy	18.37	20.39
Security	17.20	20.52

7.4.2 Number of Characters

The second feature that we will discuss is the average number of characters. In Table 12 the average number of characters of both the labelled and the unlabelled requirements are presented. In the case of Responsibility, we see that the average number of characters of labelled requirements is less than the average number of characters of unlabelled requirements. However, this is not the case with Privacy and Security. In both cases, the labelled requirements consist, at average, of a lesser number of characters.

7.4.3 Average Word Length

The third feature that we will consider is the average word length. In Table 13 the average word length of both the labelled and the unlabelled requirements are presented. We see that for both Responsibility and Privacy the averaged word length is higher for the labelled requirements than for the unlabelled ones. For the issue of Security, however, the contrary is the case.

[14] `https://github.com/Aberkane/ethics-in-re`

Table 12: Comparison of the number of characters of both labelled and unlabelled requirements.

Label	Mean labelled	Mean unlabelled
Responsibility	98.29	97.00
Privacy	87.61	97.37
Security	77.15	98.49

Table 13: Comparison of the average word length of both labelled and unlabelled requirements.

Label	Mean labelled	Mean unlabelled
Responsibility	3.66	3.47
Privacy	3.66	3.47
Security	3.38	3.50

7.4.4 Number of Stopwords

Another feature that we will discuss is the number of stopwords. Commonly, the stopwords are filtered out in natural language processing to focus on more meaningful words. However, we will compare the number of stopwords for both labelled and unlabelled requirements in an attempt to extract potential valuable information. The words that are considered stopwords were imported from the NLTK library. In Table 14 we see that in all three cases, the number of stopwords is continuously less in the labelled requirements than in the unlabelled requirements.

Table 14: Comparison of the number of stopwords in both labelled and unlabelled requirements.

Label	Mean labelled	Mean unlabelled
Responsibility	8.64	10.20
Privacy	9.07	10.20
Security	8.31	10.28

7.5 Pre-processing

In the previous paragraphs, we have extracted several basic features from the requirements data set. However, to extract more valuable information from the text, we have to pre-process the text [208]. In this section, we will conduct pre-processing to increase the focus on valuable features and reduce invaluable terms from our lexicon. We will start off with data cleaning.

7.5.1 Data cleaning

We first started by transforming all letters that comprised the requirements to the lower case. By transforming all letters to the same case, one avoids multiple representations of the same word. While computing word frequency, one may count "Software" and "software" as two different words, which is not desirable since we are adopting the bag-of-words approach. After that, we focused on punctuation. All punctuation was filtered out of the requirements, to have a clear-cut set of words as a result, without different representations of the same word. Moreover, stopwords were filtered out as well using the previously mentioned NLTK library. Again, the rationale behind removing the stopwords was to reduce the feature size.

Finally, we have conducted lemmatisation on the requirements. Lemmatization is the process of grouping the different inflected forms of a word so they can be analysed as a single entity [209, 210]. Lemmatization translates a word to its base form (i.e. lemma), which differs from stemming, which is a somewhat similar operation except for the fact that stemming results in a word (i.e. stem) that may not exist: "studying" will result to "stud", whereas in lemmatization "studying" will be transformed to "study". Furthermore, lemmatisation takes the morphological analysis of words into account while reducing them to their base form (e.g. should a word be reduced to the base form of a verb or the base form of a noun) [211].

7.6 Text processing

After cleaning the data and converting the words to their base form, we can start by extracting several interesting features from the text using natural language processing techniques. We will first start analysing unigrams, bigrams to be specific. After that, we will take a look at the term frequency-inverse document frequency of the words of which the data set is composed.

7.6.1 Bigrams

One of the limitations of a bag-of-words approach is that the context of a word is not taken into account. One way to overcome this is by making use of so-called unigrams where occurrences of pairs of consecutive words are counted [212]. In this research, we opted for analysis using bigrams which are a specific case of unigrams. Bigrams are adjacent pairs of letter sequences from a text string [213, 214]. The reason why we opted for bigrams instead of higher dimensional unigrams is due to the small size of our sentences (i.e. requirements). Bigrams may provide us with more information about the content of the text, in this section, we will analyse whether using bigrams is constructive for text classification in our case. All bigrams were computed using the scikit-learn library.

The initial computations of the bigrams were not fruitful. Numerous word combinations consisted terms that did not aid in classifying the requirement in question. Words such as "user", "want" do not add much value to the process of categorizing the requirements in the designated ethics labels. We tried to overcome this problem by excluding several words that were not necessarily useful. The following words were deemed to be of no value and were removed from the lemmatized corpus: "*user*", "*want*", "*able*", "*would*", "*like*", and "*need*".

In Table 15 we present the most frequent bigrams from both labelled and unlabelled user stories for the ethical issue of Responsibility. The bigram that tops the list in the labelled requirements is "*targeted abuse*". This is indeed related to responsibility in a sense that target abuse should be prevented by the ones responsible. Another interesting bigram is the "*live subtitle*". This can have an unethical implication if the one responsible for providing live subtitles is for example biased. Another interesting observation is that there are several bigrams present that relate to communication and media. However, there are also some bigrams that are not of value, such "ar company" which is a name of a company.

While considering the bigrams captured from the unlabelled requirements, several bigrams can be identified that actually may have something to do with responsibility. Say "*Server admin*" and "*verify see*" for example could very well be classified as requirements that may have an ethical implication, since the admin of a server may be responsible for administrating the server. The same is true for "*service provider*" and "*author post*".

Table 15: Top-10 bigram frequency of both labelled and unlabelled requirements regarding the issue of Responsibility.

Responsibility			
Labelled requirements		*Unlabelled requirements*	
Bigram	Frequency	Bigram	Frequency
targeted abuse	5	server admin	23
live subtitle	5	task requester	22
ar company	4	author post	21
text audio	3	service provider	19
video message	3	service seeker	18
image text	3	cal poly	17
client playout	3	post create	13
playout server	3	ebu tt	12
broadcaster transport	3	group chat	11
transport live	3	verify see	11

In Table 16 we present the most frequent bigrams from both labelled and unlabelled user stories for the ethical issue of Privacy. The most frequent bigram in the list of labelled requirements is "*targeted abuse*" which is not directly linked to privacy; however, it could be a result of a breach of privacy. The rest of the list is made up out of bigrams that describe transmittance of media or information. All of these bigrams that describe the transmittance of media or information may be related to privacy, since the information may contain sensitive or personal information (e.g. location). If we observe the bigrams that were extracted from the list of requirements that was not tagged as being related to the ethical issue of Privacy (i.e. the unlabelled requirements), no bigrams seem to be related to privacy.

Table 16: Top-10 bigram frequency of both labelled and unlabelled requirements regarding the issue of Privacy.

Privacy			
Labelled requirements		*Unlabelled requirements*	
Bigram	Frequency	Bigram	Frequency
targeted abuse	5	server admin	24
text audio	4	author post	22
video message	4	task requester	20
group chat	4	service provider	19
audio video	4	cal poly	17
send medium	4	service seeker	17
image text	4	post create	14
medium image	4	ebu tt	12
community group	3	verify see	12
current location	3	tt live	10

The last issue for which we captured bigrams is the issue of Security. Table 17 shows the most frequent bigrams from both labelled and unlabelled user stories for the ethical issue of Security. Similar to the list of Responsibility, "*targeted abuse*" is the bigram that tops the list of the bigrams that were extracted from the labelled requirements. Again, this bigram is not directly related to Security, but, it could be a product of a security breach. Second in place comes "*budget category*", which can very well be related to security since it can be linked to money. The rest of the bigrams is related to communication, which is indeed linked to security since information is desired to be communicated over a secure line. Looking at the list of bigrams that were captured from the unlabelled requirements, no bigram seems to be directly related to the ethical issue of Security.

Table 17: Top-10 bigram frequency of both labelled and unlabelled requirements regarding the issue of Security.

Security			
labelled requirements		*Unlabelled requirements*	
Bigram	Frequency	Bigram	Frequency
targeted abuse	5	server admin	23
budget category	5	author post	22
text audio	4	task requester	20
audio video	4	service provider	19
create account	4	service seeker	18
video message	4	cal poly	17
medium image	4	post create	14
image text	4	verify see	13
group chat	4	ebu tt	12
send medium	4	create private	10

7.6.2 Term Frequency–Inverse Document Frequency

Another interesting metric is the term frequency-inverse document frequency (TFIDF). TFIDF is a numerical statistic that shows how important a word is to a corpus and consists of two terms: the term frequency and the inverse document frequency [215]. Term frequency is the number of times a term appears in a sentence, or in our case a requirement [215, 216]. When measuring the term frequency, all terms are treated as equally important even though not all are. The inverse document frequency measures how infrequently a term occurs across all requirements, i.e. the uniqueness of a term. The product of the term frequency and the inverse document frequency is the product of the frequency of a term in a document multiplied by the uniqueness of the term with relation to the corpus [215, 216]. In this section, we have computed the average TFIDF score of all words across the requirements that are related to a particular ethics label, in an attempt to find terms that can be used for text classification.

Table 18 shows the top-10 TFIDF scores for terms from both labelled and unlabelled requirements concerning the ethics label of Responsibility. The results show that several terms appear in both lists of labelled and unlabelled requirements (e.g. "*post*"). The most frequent term that is unique to the top-10 of labelled requirements is "*article*", which is not necessarily related to the ethics label of Responsibility. The same is true for the other terms in the list. Looking at the list of top-10 TFIDF scores of terms captured from unlabelled requirements, we observe that several terms can be linked to the ethics label of Security. "*Verify*" for example is a term that could be related to security.

Table 18: Comparison of TFIDF of labelled and unlabelled requirements regarding the issue of Responsibiliy.

Responsibility			
labelled requirements		*Unlabelled requirements*	
Term	Weight	Term	Weight
post	0.041494	see	0.028478
server	0.035238	post	0.026360
verify	0.035222	author	0.025615
article	0.033162	verify	0.025589
location	0.032247	service	0.021248
abuse	0.030733	recipe	0.018267
targeted	0.030733	create	0.017392
employee	0.030092	page	0.016938
share	0.028307	game	0.016237
image	0.027218	task	0.015948

In Table 19 the top-10 TFIDF scores are presented for terms from both labelled and unlabelled requirements concerning the ethics label of Privacy. If we observe the list of most frequent terms from the labelled requirements, we can identify several terms that are directly related to privacy (e.g. location). The majority of these related terms can be linked to information that may be valuable (e.g. "*image*"). However, there are some terms that are not necessarily related to the ethics label of Privacy, for example, "*author*" and "*operator*". The list of the top-10 TFIDF scores of terms captured from unlabelled requirements does not include terms that are directly linked ethics label of Privacy.

Table 19: Comparison of TFIDF of labelled and unlabelled requirements regarding the issue of Privacy.

Privacy			
labelled requirements		*Unlabelled requirements*	
Term	Weight	Term	Weight
see	0.039943	post	0.028450
location	0.039129	see	0.027477
employee	0.033377	verify	0.026157
change	0.033303	author	0.025746
operator	0.028853	service	0.021319
image	0.028733	recipe	0.018639
video	0.027461	create	0.018180
chat	0.027411	page	0.017504
author	0.027387	game	0.016410
send	0.027067	topic	0.016177

Table 20 shows the top-10 TFIDF scores are presented for terms across labelled and unlabelled requirements with regard to the ethics label of Security. The list of terms that were extracted from the labelled requirements consist mostly of terms that describe "things" that are preferably secured since they may contain valuable information, for example "*account*" and "*password*". On the other hand, other terms such as "*create*" were directly related to security. The list of the top-10 TFIDF scores of terms captured from unlabelled requirements does not include terms that are directly related to the ethics label of Security.

Table 20: Comparison of TFIDF of labelled and unlabelled requirements regarding the issue of Security.

Security			
labelled requirements		*Unlabelled requirements*	
Term	Weight	Term	Weight
account	0.057165	see	0.029510
create	0.039058	post	0.029296
change	0.038264	verify	0.027431
log	0.037314	author	0.026457
budget	0.035886	service	0.022133
app	0.030740	recipe	0.018223
password	0.027571	topic	0.017066
sign	0.026024	page	0.016951
company	0.025979	employee	0.016886
mobile	0.024451	game	0.016112

7.7 Threats to Validity

There are several threats to the validity of this research that need to be considered while interpreting the results. The first set of threats have to do with the validity of the data set. Firstly, the amount of requirements is too low for any machine learning technique which might have affected the results. Moreover, the requirements were obtained from Github repositories. These requirements were not validated, nor was the source validated. Which means that we are not sure whether the data set resembles "professional" user stories.

After collecting the data, the data set was manually tagged by one researcher. This process may have been affected by the bias of the researcher. Determining whether a requirement can have ethical implications is subject to researcher interpretation and may affect this research. Furthermore, the analysis of the computed results was also conducted by one researcher. Determining whether a bigram or term is related to a particular ethics label is a difficult undertaking and subject to the interpretation of the terms. The same applies to the terms with the highest TFIDF.

7.8 Future Work

In this research we have explored different textual features; however, there is much more to investigate in the line of this study. First of all, exploring features from more voluminous sources, and sources from different domains, may contribute to more conclusive findings that might be useful for conducting classification. Conducting actual classification, comparing different classifiers and considering different features are also in the line of this research. The same is true for validating such a tool with professionals and stakeholders in the actual process of requirements engineering. Finally, another interesting research would be exploring whether class imbalance is intrinsic to requirements or user stories [204].

7.9 Conclusion

In this section, we attempted to answer the second research question which reads as follows “How to automatically identify ethical issues in requirements engineering?”. The rationale behind this issue was to prove the usability of the captured ethical issues in the domain of requirements engineering and to raise ethical awareness among the requirements engineers and the engaged stakeholders. Initially, we attempted to answer this question by building a classifier to identify ethical issues in a set of requirements. However, imbalanced data threw a spanner in the works. This happening made us focus on feature extraction instead, and trying to identify interesting features that could be used for further classification and research.

First, we conducted supportive basic feature extraction and compared the number of words, characters, average word length, and the number of stopwords in both labelled and unlabelled requirements. The labelled requirements consisted in general of fewer words than the unlabelled requirements. With regards to the number of characters we can conclude that in the case of the ethics label of Responsibility, the number of characters was slightly in favour of the labelled requirements. However, in the other two cases, i.e. the ethics labels of Privacy and Security, the average number of characters were less than in the unlabelled requirements. Therefore these numbers are not conclusive. The same is true for the average word length: in the case Responsibility the average word length was higher under the labelled requirements, but the contrary was the case with the ethics labels of Privacy and Security. With regards to the number of stopwords, the results are unequivocal: the labelled requirements consist of, on average, fewer stopwords than their labelled counterparts.

Secondly, we conducted more advanced text processing and compared bigrams and TFIDFs for the labelled and unlabelled requirements for all three considered ethics labels. In most cases, the bigrams proved to be useful for categorising the requirements. The TFIDF, on the other hand, was for us, humans, more challenging to use. The terms were in most cases not of value in the process of categorising the requirements. In conclusion: the bigrams proved their value in text categorisation, and the TFIDF not necessarily, however, one should keep the threats to validity in mind which means that this conclusion is far from conclusive.

8 Conclusion

In this thesis, we have explored ethics in requirements engineering. This study aimed to raise awareness about the ethical implications of software from a requirements engineering perspective. To do so, three research questions were formulated and addressed. To conclude this research, we present the answers to our research questions. We finalise this section by highlighting some possibilities for future work.

RQ1

The first research question reads as follows: "***What are the ethical issues related to requirements engineering?***". This question was raised to raise ethical awareness about the implications of software in the process of requirements engineering. This question was divided into two sub-research questions. The first sub-research question is: "**What does current scientific literature state about ethical issues related to and requirements engineering?**". To answer this question, we have conducted a systematic review of academic literature. We identified 680 studies from searching through the databases, of which 150 studies proved to be relevant to this question.

The relevant articles were analysed for evidence to answer the research question. These answers were interpreted using a predefined data extraction form that enabled us to extract the evidence rigorously and systematically. Related to the research question we have identified 98 different ethical issues (see Appendix B). The analysis shows an increase in publications concerning ethical issues related to requirements engineering and software. The most reoccurring ethical issues in the literature were Responsibility, Code of Ethics, Privacy, and Teaching Ethics.

To answer the second sub-research question, "**What does grey literature state about ethical issues related to requirements engineering?**", we have reviewed the grey literature to analyse the public discourse about ethical issues related to requirements engineering. In total, 350 news articles were considered from 5 different sources from all over the world. Less than half, 126 articles, proved to be relevant to the research question. These articles were read in full and ethical issues were extracted from them. These ethical issues were captured through a predefined data extraction form. In total, we have identified 34 ethical issues from the relevant articles.

The captured data was analysed quantitatively, and the resulting list of all ethical issues can be found in Appendix D. This list of ethical issues is our answer to sub-research question 2. Further analysis of these issues shows that there is an increase in public discourse visible regarding ethical issues related to requirements engineering and software. Furthermore, the year 2017 was the year in which most of the relevant articles were captured (see Figure 11). Our results show that the issues of Security, Privacy, and Piracy were the most reoccurring ethical issues.

RQ2

The second research question: "**How to automatically identify ethical issues in requirements?**" was posed to prove the applicability of the obtained ethical issues in the field of requirements engineering. In our approach, we focused on feature extraction to identify interesting features that could be used for further classification and research. Among other things, we conducted advanced text processing and compared bigrams and TFIDFs for labelled and unlabelled requirements. In contrary to TFIDFs, bigrams proved to be useful for categorising requirements.

8.1 Discussion

Since the aim of this thesis was to explore ethics in requirements engineering to increase ethical awareness in the discipline, we will discuss the results that are related to this goal.

Concerning the results of the systematic literature review, we have concluded that the most reoccurring ethical issue is Responsibility. This is not a surprise since understanding the ethical implications of one's work is key to taking responsibility. However, in contrary to the studies that discussed software ethics as part of a university curriculum, the literature did not mention any method or tooling that could be used to assess or raise ethical awareness among the software professional besides the Code of Ethics. Furthermore, no study discussed or proposed apparatus to increase the ethicality of requirements engineers.

In contrary to educating software professionals, educating future professionals was reiterated a significant number of times in peer-reviewed literature under the issue of Teaching Ethics. The majority of the articles regarding Teaching Ethics proposed curricula that touched upon ethics, but it would be interesting to validate the teaching itself. Berry and Berenbach experienced a "disturbing experience" when they assessed the ethicality of their students after learning ethics [156]. This was one of the few studies that focused on assessing the state the ethicality of students teaching them ethics. The results of these validations and assessments could lead to new insights and could even be used among the software professionals, including requirements engineers.

8.2 Future work

With this research, we aimed to raise ethical awareness in software from the requirements engineering perspective. The results of this research provided us with new insights for future research.

One of the criticisms regarding the Code of Ethics was that it was too general [10]. It would be interesting to identify domain-specific ethical issues that can form the basis for domain-specific Codes of Ethics. A possible approach could be to analyse the ethical issues from the SLR and GLR of this research and categorise them into the different domains and positions in software development.

Another way to extend this research is by validating the ethical issues in the real-life processes of requirements engineering or other disciplines of software development. Getting insights from software professionals could improve the body of ethical issues and maybe even present solutions on how to overcome them.

A different approach to support the ethical issues and giving them more weight is by developing a tool that identifies actual news reports related to ethical issues of a particular Code of Ethics, to confront the requirements engineer and stakeholders with real-life implications of unethical software. This knowledge may result in new insights and increase consciousness of the implications of one's work.

Furthermore, in line with our second research question, it would be interesting to explore how machine learning could be used to aid in raising ethical awareness in the processes of requirements engineering. One way to do it is to build a classifier that can identify possible ethical implications of particular requirements which may cause the requirements engineers and stakeholders to review the requirement in question and avoid undesirable outcomes. In this approach, it would be interesting to explore the different outcomes for the classification task with different requirement types.

References

[1] R. Wieringa, *Design science methodology for information systems and software engineering*. Springer, 2014, 10.1007/978-3-662-43839-8.

[2] T. Bynum, "Computer and information ethics," in *The Stanford Encyclopedia of Philosophy*, winter 2016 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2016.

[3] J. H. Moor, "If aristotle were a computing professional," *ACM SIGCAS Computers and Society*, vol. 28, no. 3, pp. 13–16, 1998.

[4] D. Gotterbarn, K. Miller, and S. Rogerson, "Software engineering code of ethics," *Communications of the ACM*, vol. 40, no. 11, pp. 110–118, 1997.

[5] ——, "Software engineering code of ethics is approved," *Communications of the ACM*, no. 10, pp. 102–107, 1999.

[6] ——, "ACM approve software engineering code of ethics," *Computer*, no. 10, pp. 84–88, Oct 1999.

[7] D. Gotterbarn, "How the new software engineering code of ethics affects you," *IEEE Software*, no. 6, pp. 58–64, 1999.

[8] N. McBride, "The ethics of software engineering should be an ethics for the client," *Communications of the ACM*, vol. 55, no. 8, pp. 39–41, 2012.

[9] P. Oriogun, O. Akinbule, C. Ibecheozor, and Z. Nyako, "Software engineering ethical decision making and professional responsibility," 2012, pp. 7–14.

[10] R. M. Mason and K. Gallagher, "A pragmatic framework for ethical decision making: The limits of professional codes," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*. IEEE, 2009, pp. 1–7.

[11] C. Wohlin *et al.*, *Engineering and managing software requirements*. Springer Science & Business Media, 2005.

[12] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *Proceedings of the 9th international conference on Software Engineering*. IEEE Computer Society Press, 1987, pp. 328–338.

[13] B. Nuseibeh and S. Easterbrook, "Requirements engineering: a roadmap," in *Proceedings of the Conference on the Future of Software Engineering*. ACM, 2000, pp. 35–46.

[14] N. Ochara, D. Asmelash, and S. Mlay, "Groupthink decision making deficiency in the requirements engineering process: Towards a crowdsourcing model." International Business Information Management Association, IBIMA, 2012, pp. 1654–1674.

[15] C. Detweiler and M. Harbers, "Value stories: Putting human values into requirements engineering." CEUR-WS, 2014, pp. 2–11.

[16] R. Epstein, "Demons in the it workplace," 2004, pp. 37–48.

[17] F. Kammuller, J. Augusto, and S. Jones, "Security and privacy requirements engineering for human centric iot systems using efriend and isabelle." Institute of Electrical and Electronics Engineers Inc., 2017, pp. 401–406.

[18] C. Cary, H. Wen, and P. Mahatanankoon, "Data mining: Consumer privacy, ethical policy, and systems development practices," *Human Systems Management*, no. 4, pp. 157–168, 2003.

[19] A. Kung, C. Jouvray, and F. Coudert, “Salt frameworks to tackle surveillance and privacy concerns,” in *2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, Feb 2015, pp. 665–673.

[20] R. Vaughn Jr. and J. Boggess III, “Integration of computer security into the software engineering and computer science programs,” *Journal of Systems and Software*, no. 2, pp. 149–153, 1999.

[21] L. Xiao, P. Lewis, and A. Gibb, “Developing a security protocol for a distributed decision support system in a healthcare environment,” 2008, pp. 673–682.

[22] N. Rozanski and E. Woods, *Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives*, 2nd ed. Addison-Wesley Professional, 2011.

[23] S. Keele *et al.*, “Guidelines for performing systematic literature reviews in software engineering,” in *Technical report, Ver. 2.3 EBSE Technical Report. EBSE.* sn, 2007.

[24] V. Garousi, M. Felderer, and M. V. Mäntylä, “Guidelines for including the grey literature and conducting multivocal literature reviews in software engineering,” *arXiv preprint arXiv:1707.02553*, 2017.

[25] G. Génova, M. R. González, and A. Fraga, “Ethical education in software engineering: Responsibility in the production of complex systems,” *Science and engineering ethics*, vol. 13, no. 4, pp. 505–522, 2007.

[26] M. Kassab, C. Neill, and P. Laplante, “State of practice in requirements engineering: contemporary data,” *Innovations in Systems and Software Engineering*, vol. 10, no. 4, pp. 235–241, 2014.

[27] M. I. Kant, “Foundations of the metaphysics of morals (translated by lewis white beck),” in *Seven Masterpieces of Philosophy*. Routledge, 2016, pp. 285–336.

[28] (2018) Cambridge dictionary. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/ethics

[29] D. Berdichevsky and E. Neuenschwander, “Toward an ethics of persuasive technology,” *Communications of the ACM*, vol. 42, no. 5, pp. 51–58, 1999.

[30] R. Kraut, “Aristotle’s ethics,” in *The Stanford Encyclopedia of Philosophy*, summer 2017 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2017.

[31] T. W. Bynum, “Computer ethics: basic concepts and historical overview,” 2001.

[32] ——, “A very short history of computer ethics,” *APA Newsletters on Philosophy and Computers*, vol. 99, no. 2, 2000.

[33] W. Maner, “Is computer ethics unique?” 1999.

[34] L. Floridi, Ed., *Blackwell Guide to the Philosophy of Computing and Information*. Cambridge, MA, USA: Blackwell Publishers, Inc., 2003.

[35] L. Floridi and J. W. Sanders, “Mapping the foundationalist debate in computer ethics,” *Ethics and information Technology*, vol. 4, no. 1, pp. 1–9, 2002.

[36] J. H. Moor, “What is computer ethics?” *Metaphilosophy*, vol. 16, no. 4, pp. 266–275, 1985.

[37] D. Gotterbarn, “Reducing software failures: Addressing the ethical risks of the software development lifecycle,” *Australasian Journal of Information Systems*, vol. 9, no. 2, 2002.

[38] ——, “Raising the bar: A software engineering code of ethics and professional practice.” Association for Computing Machinery, Inc, 1998, pp. 26–28.

[39] P. J. Phillips, F. Jiang, A. Narvekar, J. Ayyad, and A. J. O'Toole, "An other-race effect for face recognition algorithms," *ACM Transactions on Applied Perception (TAP)*, vol. 8, no. 2, p. 14, 2011.

[40] S. Lohr, "Facial recognition is accurate, if you're a white guy," *The New York Times*, 2018, accessed on 20.02.2018. [Online]. Available: https://www.nytimes.com/2018/02/09/technology/facial-recognition-race-artificial-intelligence.html

[41] V. Garousi and A. Sedighi, "Efficiently manage your iterative software projects," *IEEE Software*, no. 5, pp. 101–102, Sep. 2007.

[42] R. T. Ogawa and B. Malen, "Towards rigor in reviews of multivocal literatures: Applying the exploratory case study method," *Review of Educational Research*, vol. 61, no. 3, pp. 265–286, 1991.

[43] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, "Systematic mapping studies in software engineering." in *EASE*, vol. 8, 2008, pp. 68–77.

[44] M. Kuhrmann, D. M. Fernández, and M. Daneva, "On the pragmatic design of literature studies in software engineering: an experience-based guideline," *Empirical software engineering*, vol. 22, no. 6, pp. 2852–2891, 2017.

[45] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80, no. 4, pp. 571–583, 2007.

[46] E. Georgiadou and P. Oriogun, "Professional issues in software engineering curricula: Case studies on ethical decision making." Institute of Electrical and Electronics Engineers Inc., 2001, pp. 252–261.

[47] P. R. Srivastava, "Appraisal for lokpal body using fuzzy multicriteria approach," in *Advanced Software Engineering & Its Applications (ASEA), 2015 8th International Conference on*. IEEE, 2015, pp. 5–12.

[48] D. Gotterbarn, K. Miller, S. Rogerson, S. Barber, P. Barnes, I. Burnstein, M. Davis, A. El-Kadi, N. Fairweather, M. Fulghum, N. Jayaram, T. Jeweth, M. Kanko, E. Kallman, D. Langford, J. Little, E. Mechler, M. Norman, D. Phillips, P. Prinzivalli, P. Sullivan, J. Weckert, V. Weil, S. Weisband, and L. Werth, "Software engineering code of ethics and professional practice," *Science and Engineering Ethics*, no. 2, pp. 231–238, 2001.

[49] "Volkswagen emissions scandal," Jul 2018. [Online]. Available: https://en.wikipedia.org/wiki/Volkswagen_emissions_scandal

[50] T. Moores and J.-J. Chang, "Ethical decision making in software piracy: Initial development and test of a four-component model," *MIS Quarterly: Management Information Systems*, no. 1, pp. 167–180, 2006.

[51] B. Begier, "Users' involvement may help respect social and ethical values and improve software quality," *Information Systems Frontiers*, no. 4, pp. 389–397, 2010.

[52] G. Eden, M. Jirotka, and B. Stahl, "Responsible research and innovation: Critical reflection into the potential social consequences of ict," 2013.

[53] A. Thomson and D. Schmoldt, "Ethics in computer software design and development," *Computers and Electronics in Agriculture*, no. 1-3, pp. 85–102, 2001.

[54] M. A. Ferrario, W. Simm, S. Forshaw, A. Gradinar, M. T. Smith, and I. Smith, "Values-first se: Research principles in practice," in *2016 IEEE/ACM 38th International Conference on Software Engineering Companion (ICSE-C)*, May 2016, pp. 553–562.

[55] A. Rashid, K. Moore, C. May-Chahal, and R. Chitchyan, "Managing emergent ethical concerns for software engineering in society." IEEE Computer Society, 2015, pp. 523–526.

[56] A. Ema, H. Osawa, H. Hattori, N. Akiya, N. Kanzaki, R. Ichise, M. Kukita, T. Otani, A. Kubo, K. Komatani, R. Saijo, M. Tanaka, K. Honda, N. Miyano, Y. Yashiro, and G. Yoshizawa, "Breaking down silos: Involving various researchers for driving hci research." Association for Computing Machinery, 2017, pp. 837–847.

[57] D. Gotterbarn, T. Clear, W. Gray, and B. Houliston, "Developing software in bicultural context: The role of a sodis inspection," *International Journal of Technology and Human Interaction (IJTHI)*, no. 2, pp. 1–23, 2006.

[58] B. Barn, "Do you own a volkswagen? values as non-functional requirements," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 151–162, 2016.

[59] A. Alrobai, J. McAlaney, H. Dogan, K. Phalp, and R. Ali, "Exploring the requirements and design of persuasive intervention technology to combat digital addiction," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 130–150, 2016.

[60] R. Ali, N. Jiang, K. Phalp, S. Muir, and J. McAlaney, "The emerging requirement for digital addiction labels," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 198–213, 2015.

[61] T. Busch, "Capabilities in, capabilities out: Overcoming digital divides by promoting corporate citizenship and fair ict," *Ethics and Information Technology*, no. 4, pp. 339–353, 2011.

[62] L. Yilmaz, A. Franco-Watkins, and T. S. Kroecker, "Coherence-driven reflective equilibrium model of ethical decision-making," in *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, March 2016, pp. 42–48.

[63] G. Dodig-Crnkovic and B. Çürüklü, "Robots: Ethical by design," *Ethics and Information Technology*, no. 1, pp. 61–71, 2012.

[64] A. Kornai, "Bounding the impact of agi," *Journal of Experimental and Theoretical Artificial Intelligence*, no. 3, pp. 417–438, 2014.

[65] W. Fleischman, "Just say "no!" to lethal autonomous robotic weapons," *Journal of Information, Communication and Ethics in Society*, no. 3-4, pp. 299–313, 2015.

[66] P. Watson, P. Duquenoy, M. Brennan, M. Jones, and J. Walkerdine, "Towards an ethical interaction design: The issue of including stakeholders in law-enforcement software development," 2009, pp. 313–316.

[67] C. Huff, D. Johnson, and K. Miller, "Virtual harms and real responsibility," *IEEE Technology and Society Magazine*, no. 2 SPEC, pp. 12–19, 2003.

[68] Z.-U.-H. Usmani, F. Alghamdi, A. Tariq, and T. Puri, "I know what you did this summer - users' behavior on internet," 2010, pp. 19–24.

[69] M. Iqbal, M. Abid, and M. Ahmad, “Catching webspam traffic with artificial immune system (ais) classification algorithm.” IEEE Computer Society, 2017, pp. 402–405.

[70] W. Kuchinke, “Ethical concerns caused by integrative patient empowerment solutions for personalized medicine.” Institute of Electrical and Electronics Engineers Inc., 2013, pp. 4775–4778.

[71] M. Craven, A. Lang, and J. Martin, “Developing mhealth apps with researchers: Multi-stakeholder design considerations,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, no. PART 3, pp. 15–24, 2014.

[72] A. Sutcliffe, P. Rayson, C. Bull, and P. Sawyer, “Discovering affect-laden requirements to achieve system acceptance.” Institute of Electrical and Electronics Engineers Inc., 2014, pp. 173–182.

[73] C. ölvingson, J. Hallberg, T. Timpka, and K. Lindqvist, “Ethical issues in public health informatics: Implications for system design when sharing geographic information,” *Journal of Biomedical Informatics*, no. 3, pp. 178–185, 2002.

[74] G. Aumayr, D. Bleier, G. Chroust, and N. Sturm, “Understanding needs and requirements of the target group: Systematic perspective on interaction between system and environment.” Institute of Electrical and Electronics Engineers Inc., 2017.

[75] B. Begier, “Effective cooperation with energy consumers: An example of an ethical approach to introduce an innovative solution,” *Journal of Information, Communication and Ethics in Society*, no. 2, pp. 107–121, 2014.

[76] H. Khalid, E. Shihab, M. Nagappan, and A. Hassan, “What do mobile app users complain about?” *IEEE Software*, no. 3, pp. 70–77, 2015.

[77] D. Thain, T. Tannenbaum, and M. Livny, “How to measure a large open-source distributed system,” *Concurrency Computation Practice and Experience*, no. 15, pp. 1989–2019, 2006.

[78] G. Zhang, H. N. Liang, and Y. Yue, “An investigation of the use of robots in public spaces,” in *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, June 2015, pp. 850–855.

[79] E. Kaasinen, M. Niemelä, T. Tuomisto, P. Välkkynen, and V. Ermolov, “Identifying user requirements for a mobile terminal centric ubiquitous computing architecture,” 2006, pp. 9–16.

[80] R. Chwastek, “Cognitive systems in human resources,” in *2017 International Conference on Behavioral, Economic, Socio-cultural Computing (BESC)*, Oct 2017, pp. 1–4.

[81] M. Tee, R. Abdullah, J. Din, S. Abdullah, and L. Wu, “Green sd adoption using knowledge mangement facilitation – a motivational perspective,” *Journal of Theoretical and Applied Information Technology*, no. 17, pp. 4291–4303, 2017.

[82] M. Moraga, I. García-Rodríguez de Guzmán, C. Calero, T. Johann, G. Me, H. Münzel, and J. Kindelsberger, “Greco: Green code of ethics,” *Journal of Software: Evolution and Process*, no. 2, 2017.

[83] R. Glass, J. Rost, and M. Matook, “Lying on software projects,” *IEEE Software*, no. 6, pp. 90–95, 2008.

[84] H. Smith and M. Keil, “The reluctance to report bad news on troubled software projects: A theoretical model,” *Information Systems Journal*, no. 1, pp. 69–95, 2003.

[85] D. Hepting, “Design and development of s.o.l.e. software,” 2010, pp. 1910–1916.

[86] M. Staples, L. Zhu, and J. Grundy, "Continuous validation for data analytics systems." IEEE Computer Society, 2016, pp. 769–772.

[87] M. Larsson and I. Crnkovic, "Possible implications of design decisions based on predictions," 2004, pp. 651–656.

[88] M. Kutar and B. Light, "Exploring cultural issues in the packaged software industry: A usability perspective," 2005.

[89] W. Pieters and A. van Cleeff, "The precautionary principle in a world of digital dependencies," *Computer*, no. 6, pp. 50–56, June 2009.

[90] M. Turilli, "Ethics and the practice of software design," *Frontiers in Artificial Intelligence and Applications*, no. 1, pp. 171–183, 2008.

[91] C. Lundestad and A. Hommels, "Software vulnerability due to practical drift," *Ethics and Information Technology*, no. 2, pp. 89–100, 2007.

[92] K. Miller, "Software informed consent: Docete emptorem, not caveat emptor," *Science and Engineering Ethics*, no. 3, pp. 357–362, 1998.

[93] P. Black, "Static analyzers: Seat belts for your code," *IEEE Security and Privacy*, no. 3, pp. 48–52, 2012.

[94] D. Gotterbarn, "Enhancing risk analysis using software development impact statements." Institute of Electrical and Electronics Engineers Inc., 2001, pp. 43–51.

[95] J. van den Hoven, "Ict and value sensitive design," *IFIP International Federation for Information Processing*, pp. 67–72, 2007.

[96] H.-M. Järvinen, "Ethics as a skill of a software engineer?" European Society for Engineering Education (SEFI), 2017, pp. 856–862.

[97] K. Sutling, Z. Mansor, S. Widyarto, S. Letchmunan, and N. Arshad, "Agile project manager behavior: The taxonomy." Institute of Electrical and Electronics Engineers Inc., 2014, pp. 234–239.

[98] J. Carlisle, "Ethical considerations of the software-dependent organization," *Journal of Systems and Software*, no. 3, pp. 251–255, 1999.

[99] S. Athavale and M. Singh, "Modeling work-ethics spread in software organizations." Association for Computing Machinery, Inc, 2014, pp. 2–7.

[100] A. Y. Gheni, Y. Y. Jusoh, M. A. Jabar, N. M. Ali, R. H. Abdullah, S. Abdullah, and M. S. Khalefa, "The virtual teams: E-leaders challenges," in *2015 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*, Aug 2015, pp. 38–42.

[101] F. Cafer and S. Misra, "Effective project leadership in computer science and engineering," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, no. PART 2, pp. 59–69, 2009.

[102] S. Radevski, H. Hata, and K. Matsumoto, "Real-time monitoring of neural state in assessing and improving software developers' productivity." Institute of Electrical and Electronics Engineers Inc., 2015, pp. 93–96.

[103] S. Van Der Graaf, "Imaginaries of ownership: The logic of participation in the moral economy of 3d software design," *Telematics and Informatics*, no. 2, pp. 400–408, 2015.

[104] J. Sipior and B. Ward, “Ethical responsibility for software development,” *Information Systems Management*, no. 2, pp. 68–72, 1998.

[105] U. Becker-Kornstaedt, “Descriptive software process modeling - how to deal with sensitive process information,” *Empirical Software Engineering*, no. 4, pp. 353–367, 2001.

[106] G. Gell, “Side effects and responsibility of medical informatics,” *International Journal of Medical Informatics*, no. 2-3, pp. 69–81, 2001.

[107] L. Werth, “Certification and licensing for software professionals and organizations,” in *Proceedings 11th Conference on Software Engineering Education*, Feb 1998, pp. 151–160.

[108] J. B. Thompson, “Knowledge, professionalism and free movement of labour (visions of the software engineering future),” in *Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000*, 2000, pp. 28–37.

[109] Y. Lurie and S. Mark, “Professional ethics of software engineers: An ethical framework,” *Science and Engineering Ethics*, no. 2, pp. 417–434, 2016.

[110] T. Vartiainen, M. Siponen, and G. Moody, “Gray-area phenomenon in information systems development: A call for research,” 2011.

[111] D. Gotterbarn, “Not all codes are created equal: The software engineering code of ethics, a success story,” *Journal of Business Ethics*, no. 1, pp. 81–89, 1999.

[112] D. Gotterbarn and K. W. Miller, “The public is the priority: Making decisions using the software engineering code of ethics,” *Computer*, no. 6, pp. 66–73, June 2009.

[113] P. Aiken, R. Stanley, L. Anderson, and J. Billings, “Use of the ieee codes of conduct to resolve legal disputes,” *Computer*, no. 99, pp. 1–1, 2010.

[114] A. Takanen, P. Vuorijärvi, M. Laakso, and J. Röning, “Agents of responsibility in software vulnerability processes,” *Ethics and Information Technology*, no. 2, pp. 93–110, 2004.

[115] N. Karim, F. Ammar, and R. Aziz, “Ethical software: Integrating code of ethics into software development life cycle.” Institute of Electrical and Electronics Engineers Inc., 2017, pp. 290–298.

[116] J. B. Thompson, “Industry and academia: can they agree on best practice within software engineering?” in *29th Annual International Computer Software and Applications Conference (COMPSAC’05)*, July 2005, pp. 173 Vol. 2–.

[117] D. Payne and B. Landry, “A uniform code of ethics: Business and it professional ethics,” *Communications of the ACM*, no. 11, pp. 81–84, 2006.

[118] M. Simonette, M. Magalhaes, and E. Spina, “Extending essence kernel to deal with ieee code of ethics.” Institute of Electrical and Electronics Engineers Inc., 2016, pp. 383–387.

[119] O. Burmeister and J. Weckert, “Applying the new software engineering code of ethics to usability engineering: A study of four cases,” *Journal of Information, Communication and Ethics in Society*, no. 3, pp. 119–132, 2003.

[120] H. Münzel, “Towards an ethical foundation of green software engineering.” Institute of Electrical and Electronics Engineers Inc., 2015, pp. 23–26.

[121] R. Mason and K. Gallagher, “A pragmatic framework for ethical decision making: The limits of professional codes,” 2009.

[122] K. Judy, “Agile principles and ethical conduct,” 2009.

[123] H. Rahanu, E. Georgiadou, K. Siakas, and M. Ross, "Towards developing a software process improvement strategy through the application of ethical concepts," *Communications in Computer and Information Science*, pp. 627–641, 2017.

[124] M. Sojer, O. Alexy, S. Kleinknecht, and J. Henkel, "Understanding the drivers of unethical programming behavior: The inappropriate reuse of internet-accessible code," *Journal of Management Information Systems*, no. 3, pp. 287–325, 2014.

[125] T. Ahmed and A. Srivastava, "Understanding and evaluating the behavior of technical users. a study of developer interaction at stackoverflow," *Human-centric Computing and Information Sciences*, no. 1, 2017.

[126] M. Siponen and T. Vartiainen, "Unauthorized copying of software and levels of moral development: A literature analysis and its implications for research and practice," *Information Systems Journal*, no. 4, pp. 387–407, 2004.

[127] S. Duda and V. Peters, "Thou shalt not...: a look at the ethics of copying software code." Institute of Electrical and Electronics Engineers Inc., 2014.

[128] L. An, O. Mlouki, F. Khomh, and G. Antoniol, "Stack overflow: A code laundering platform?" Institute of Electrical and Electronics Engineers Inc., 2017, pp. 283–293.

[129] K. Kimppa, "Socially responsible international intellectual property rights in software and other digitally distributable material," *IFIP International Federation for Information Processing*, pp. 37–50, 2006.

[130] S. Sedlet, "Computers, ethics, law and society: what do we teach undergraduates?" 1999, pp. 249–253.

[131] E. Towell, "Teaching ethics in the software engineering curriculum." Institute of Electrical and Electronics Engineers Inc., 2003, pp. 150–157.

[132] E. Towell, J. Thompson, and K. McFadden, "Introducing and developing professional standards in the information systems curriculum," *Ethics and Information Technology*, no. 4, pp. 291–299, 2004.

[133] A. Wilk, "Cyber security education and law." Institute of Electrical and Electronics Engineers Inc., 2016, pp. 94–103.

[134] D. J. Bagert, "The challenge of curriculum modeling for an emerging discipline: software engineering," in *Frontiers in Education Conference, 1998. FIE '98. 28th Annual*, Nov 1998, pp. 910–915 vol.2.

[135] M. Lutz, J. Vallino, K. Martinez, and D. Krutz, "Instilling a software engineering mindset through freshman seminar," 2012.

[136] R. Epstein, "A software engineering course with an emphasis on software processes and security," 2008, pp. 67–76.

[137] R. Duley, G. Hislop, T. Hilburn, and A. Sobel, "Engineering an introductory software engineering curriculum." Institute of Electrical and Electronics Engineers Inc., 2003, pp. 99–106.

[138] X. Liu and O. Baiocchi, "An iot course for a computer science graduate program." CRC Press/Balkema, 2017, pp. 751–755.

[139] C. De Melo and T. De Sousa, "Reflections on cyberethics education for millennial software engineers." Institute of Electrical and Electronics Engineers Inc., 2017, pp. 40–46.

[140] R. Miñano, Á. Uruburu, A. Moreno-Romero, and D. Perez-López, “Strategies for teaching professional ethics to it engineering degree students and evaluating the result,” *Science and Engineering Ethics*, no. 1, pp. 263–286, 2017.

[141] J. Magenheim and C. Schulte, “Social, ethical and technical issues in informatics-an integrated approach,” *Education and Information Technologies*, no. 3-4, pp. 319–339, 2006.

[142] E. Gehringer, “Building an ethics in computing website using peer review,” 2001, pp. 2561–2570.

[143] I. M. Ali, “Integrating playful activities in software engineering teaching,” in *2015 International Symposium on Mathematical Sciences and Computing Research (iSMSC)*, May 2015, pp. 89–93.

[144] S. Wang and X. Zhang, “Performance-oriented design model of the content & style of software engineering textbook for undergraduate computer software related majors,” 2011, pp. 152–155.

[145] S. Caltagirone, P. Ortman, S. Melton, D. Manz, K. King, and P. Oman, “Design and implementation of a multi-use attack-defend computer security lab,” 2006, p. 220c.

[146] J. Gibson, “Software reuse and plagiarism: A code of practice,” 2009, pp. 55–59.

[147] S. Niezgoda and T. Way, “Snitch: A software tool for detecting cut and paste plagiarism,” 2007, pp. 51–55.

[148] W. Fleischman, “Meta-informatics and ethical issues in computing,” 2006, pp. 232–236.

[149] B. Maxim and K. Akingbehin, “Experiences in teaching senior design using real-world clients,” 2006.

[150] A. Zeid, “Lessons learned from establishing a software engineering academic programme in developing countries,” 2007, pp. 11–18.

[151] A. Trippe, “Designing programming courses for et students,” 2002, pp. 8755–8762.

[152] J. Drake, “Class to survey application programming topics,” no. 1. Association for Computing Machinery (ACM), 1998, pp. 161–165.

[153] B. Carroll, B. Weems, and B. Khalili, “A professional practices course in computer science and engineering,” no. 122nd ASEE Annual Conference and Exposition: Making Value for Society. American Society for Engineering Education, 2015.

[154] G. Dodig-Crnkovic, “Preparing next generation of software engineers for future societal challenges and opportunities.” Association for Computing Machinery, Inc, 2015, pp. 49–52.

[155] D. Suri, “Introducing requirements engineering in an undergraduate engineering curriculum: Lessons learnt,” 2002, pp. 3175–3183.

[156] D. Berry and B. Berenbach, “Ethics test results before and after ethics training: A disturbing experience,” 2010, pp. 70–76.

[157] B. von Konsky, J. Ivins, and S. Gribble Susan J., “Engaging undergraduates in discussions about ethics in computing,” 2007, pp. 163–169.

[158] A. Sajeev and I. Crnković, “Will they report it? ethical attitude of graduate software engineers in reporting bad news,” 2012, pp. 42–51.

[159] D. Chinn and T. Vandegrift, “Uncovering student values for hiring in the software industry,” *ACM Journal on Educational Resources in Computing*, no. 4, 2008.

[160] C. B. Fleddermann, “Engineering ethics cases for electrical and computer engineering students,” *IEEE Transactions on Education*, no. 3, pp. 284–287, Aug 2000.

[161] K. W. Bowyer, "Goodearl and aldred versus hughes aircraft: a whistle-blowing case study," in *30th Annual Frontiers in Education Conference. Building on A Century of Progress in Engineering Education. Conference Proceedings (IEEE Cat. No.00CH37135)*, Oct 2000, pp. S2F/2–S2F/7 vol.2.

[162] E. A. Clancy, P. Quinn, and J. E. Miller, "Assessment of a case study laboratory to increase awareness of ethical issues in engineering," *IEEE Transactions on Education*, no. 2, pp. 313–317, May 2005.

[163] P. Oriogun, B. Ogunleye-Johnson, M. Mukhtar, and G. Tobby, "Teaching and assessing software engineering ethics in the 21st century: Case study from american university of nigeria," 2012, pp. 75–81.

[164] S. Yaman, F. Fagerholm, M. Munezero, H. Mäenpää, and T. Männistö, "Notifying and involving users in experimentation: Ethical perceptions of software practitioners," in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Nov 2017, pp. 199–204.

[165] T. Hall and V. Flynn, "Ethical issues in software engineering research: A survey of current practice," *Empirical Software Engineering*, no. 4, pp. 305–317, 2001.

[166] M.-A. Storey, B. Phillips, and M. Maczewski, "Is it ethical to evaluate web-based learning tools using students?" *Empirical Software Engineering*, no. 4, pp. 343–348, 2001.

[167] D. Badampudi, "Reporting ethics considerations in software engineering publications," in *2017 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, Nov 2017, pp. 205–210.

[168] F. Fotrousi, N. Seyff, and J. Börstler, "Ethical considerations in research on user feedback." Institute of Electrical and Electronics Engineers Inc., 2017, pp. 194–198.

[169] S. Baltes and S. Diehl, "Worse than spam: Issues in sampling software developers." IEEE Computer Society, 2016.

[170] K. Moløkken-østvold, "Ethical concerns when increasing realism in controlled experiments with industrial participants," 2005, p. 264.

[171] A. Andrews and A. Pradhan, "Ethical issues in empirical software engineering: The limits of policy," *Empirical Software Engineering*, no. 2, pp. 105–110, 2001.

[172] J. Sieber, "Protecting research subjects, employees and researchers: Implications for software engineering," *Empirical Software Engineering*, no. 4, pp. 329–341, 2001.

[173] Z. Yi, D. Xu, and J. Heales, "The moderating effect of social influence on ethical decision making in software piracy." Pacific Asia Conference on Information Systems, 2013.

[174] S. Giri, H. Mishra, and V. Sharma, "Software buying behaviour of consumer: A factor analysis approach," 2011, pp. 668–672.

[175] D. Seale, M. Polakowski, and S. Schneider, "It's not really theft!: Personal and workplace ethics that enable software piracy," *Behaviour and Information Technology*, no. 1, pp. 27–40, 1998.

[176] T. Moores, A. Nill, and M. Rothenberger, "Knowledge of software piracy as an antecedent to reducing pirating behavior," *Journal of Computer Information Systems*, no. 1, pp. 82–89, 2009.

[177] F. Grodzinsky, K. Miller, and M. Wolf, "Ethical issues in open source software," *Journal of Information, Communication and Ethics in Society*, no. 4, pp. 193–205, 2003.

[178] R. Spinello, “The future of ppen source software: Let the market decide,” *Journal of Information, Communication and Ethics in Society*, no. 4, pp. 217–233, 2003.

[179] P. Tourani, B. Adams, and A. Serebrenik, “Code of conduct in open source projects.” Institute of Electrical and Electronics Engineers Inc., 2017, pp. 24–33.

[180] S. Chopra and S. Dexter, “The freedoms of software and its ethical uses,” *Ethics and Information Technology*, no. 4, pp. 287–297, 2009.

[181] J. Aycock, A. Somayaji, and J. Sullins, “The ethics of coexistence: Can i learn to stop worrying and love the logic bomb?” in *2014 IEEE International Symposium on Ethics in Science, Technology and Engineering*, May 2014, pp. 1–4.

[182] J. Stieb, “A critique of positive responsibility in computing,” *Science and Engineering Ethics*, no. 2, pp. 219–233, 2008.

[183] G. Genova, M. González, and A. Fraga, “Ethical education in software engineering: Responsibility in the production of complex systems,” *Science and Engineering Ethics*, no. 4, pp. 505–522, 2007.

[184] J. Moor, “The nature, importance, and difficulty of machine ethics,” *IEEE Intelligent Systems*, no. 4, pp. 18–21, 2006.

[185] D. Gotterbarn, “Informatics and professional responsibility,” *Science and Engineering Ethics*, no. 2, pp. 221–230, 2001.

[186] B. Stahl, “Reflective responsibility for risk: A critical view of software and information systems development risk management,” *International Journal of Risk Assessment and Management*, no. 3, pp. 312–325, 2007.

[187] “2016 newspaper web rankings,” *4International Media & Newspapers*, accessed on 13.03.2018. [Online]. Available: https://www.4imn.com/

[188] “The top 500 sites on the web,” *Alexa*, accessed on 13.03.2018. [Online]. Available: https://www.alexa.com/topsites/category/News/Newspapers

[189] “About us,” *Al Jazeera English*, accessed on 13.03.2018. [Online]. Available: https://www.aljazeera.com/aboutus/

[190] “About china daily group,” *China Daily*, accessed on 13.03.2018. [Online]. Available: http://www.chinadaily.com.cn/static_e/aboutus.html

[191] “About iol,” *Independent Online*, accessed on 13.03.2018. [Online]. Available: https://www.iol.co.za/

[192] “New york daily news,” *Wikipedia, The Free Encyclopedia*, accessed on 13.03.2018. [Online]. Available: https://en.wikipedia.org/wiki/New_York_Daily_News

[193] “Usa today,” *Wikipedia, The Free Encyclopedia*, accessed on 13.03.2018. [Online]. Available: https://en.wikipedia.org/wiki/USA_Today

[194] “The sydney morning herald,” *Wikipedia, The Free Encyclopedia*, accessed on 13.03.2018. [Online]. Available: https://en.wikipedia.org/wiki/The_Sydney_Morning_Herald

[195] “Bbc gcse bitesize,” *Commissioned writing*, accessed on 17.04.2018. [Online]. Available: http://www.bbc.co.uk/schools/gcsebitesize/english/creativewriting/commissionsrev2.shtml

[196] “Wannacry ransomware attack,” Jul 2018. [Online]. Available: https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

[197] K. M. Benzies, S. Premji, K. A. Hayden, and K. Serrett, "State-of-the-evidence reviews: advantages and challenges of including grey literature," *Worldviews on Evidence-Based Nursing*, vol. 3, no. 2, pp. 55–61, 2006.

[198] S. Sipior, "Jc, & ward, bt (1998). ethical responsibility for software development," *Information Systems Management*, vol. 15, no. 2.

[199] P. Loucopoulos and V. Karakostas, *System requirements engineering*. McGraw-Hill, Inc., 1995.

[200] F. Sebastiani, "Machine learning in automated text categorization," *ACM computing surveys (CSUR)*, vol. 34, no. 1, pp. 1–47, 2002.

[201] "scikit-learn," *Machine Learning in Python*, accessed on 24.06.2018. [Online]. Available: http://scikit-learn.org/stable/index.html

[202] "Natural language toolkit," *NLTK 3.3 documentation*, accessed on 24.06.2018. [Online]. Available: https://www.nltk.org/

[203] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD explorations newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[204] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Special issue on learning from imbalanced data sets," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[205] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[206] M. Z. Asghar, A. Khan, S. Ahmad, and F. M. Kundi, "A review of feature extraction in sentiment analysis," *Journal of Basic and Applied Scientific Research*, vol. 4, no. 3, pp. 181–186, 2014.

[207] J.-p. Bao, J.-Y. Shen, X.-D. Liu, and Q.-B. Song, "A new text feature extraction model and its application in document copy detection," in *Machine Learning and Cybernetics, 2003 International Conference on*, vol. 1. IEEE, 2003, pp. 82–87.

[208] S. Kannan and V. Gurusamy, "Preprocessing techniques for text mining," 2014.

[209] "Lemmatisation," Jul 2018. [Online]. Available: https://en.wikipedia.org/wiki/Lemmatisation

[210] J. Plisson, N. Lavrac, D. Mladenić *et al.*, "A rule based approach to word lemmatization," 2004.

[211] T. Risueno, "What is the difference between stemming and lemmatization?" Feb 2018. [Online]. Available: https://blog.bitext.com/what-is-the-difference-between-stemming-and-lemmatization/

[212] "4.2. feature extraction." [Online]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html

[213] G. A. Rice and D. O. Robinson, "The role of bigram frequency in the perception of words and nonwords," *Memory & Cognition*, vol. 3, no. 5, pp. 513–518, 1975.

[214] R. Bekkerman and J. Allan, "Using bigrams in text categorization," Technical Report IR-408, Center of Intelligent Information Retrieval, UMass Amherst, Tech. Rep., 2004.

[215] "tf-idf," Jul 2018. [Online]. Available: https://en.wikipedia.org/wiki/Tf\T1\textendashidf

[216] B. Goralewicz, "The tf*idf algorithm explained," Mar 2018. [Online]. Available: https://www.elephate.com/blog/what-is-tf-idf/

[217] J. Thompson, “A long and winding road (progress on the road to a software engineering profession),” 2001, pp. 39–45.

[218] J. Barrie Thompson, “Developments in the fields of software engineering: Professionalism, standards and best practice,” *IFIP Advances in Information and Communication Technology*, pp. 109–122, 2005.

[219] E. Towell and J. Thompson, “A further exploration of teaching ethics in the software engineering curriculum,” 2004, pp. 39–44.

[220] P. Oriogun and B. Ogunleye-Johnson, “Computing education: A discussion paper on teaching and assessing ethics,” 2012, pp. 47–54.

[221] M. A. Dyrud, “Cases for teaching engineering ethics,” in *34th Annual Frontiers in Education, 2004. FIE 2004.*, Oct 2004, pp. S1E/10–S1E/14 Vol. 3.

A Systematic Literature Review - Data Extraction Form

Metadata

Identifier	Metadata item	Description
D1	Extraction date	Date of data extraction.
D2	Publication date	Publication year of the study.
D3	Title	Title of the study.
D4	Venue	Name of the publication venue.
D5	Author(s)	Author(s) of the study.
D6	Authors' affiliation	Academia, industry or not clear.
D7	Country/region of affiliation	What country or region is affiliated with the study?
D8	Database	Database from which the study was obtained.
D9	Citations	Number of citations *(captured from Google Scholar)*.
D10	Publication type	Journal article, conference paper or workshop paper.

Ethical awareness

Identifier	Data item	Description
D11	Need for ethical awareness	Does the article mention a need for ethical awareness (yes or no)?
D12	Ethical issue(s)	What ethical issues are stated?

Research methodology

Identifier	Data item	Description
D13	Research domain	Domain of study (requirements engineering, software engineering, etc.).
D14	Paper type	(empirical research etc.)

Empirical research features

(Only if D14 answers to "empirical research")

Identifier	Data item	Description
D15	Research method	Quantitative, qualitative or both.
D16	Data collection	How was the data collected (e.g. questionnaire)?
D17	Sample description	Population (e.g. students) and size.
D18	Research context	(agriculture, consultancy, etc.)

B Systematic Literature Review - Ethical Issues

Ethical Issue	Description	Study
Ability to apply Code of Ethics	Ability to apply Code of Ethics to different scenario's	[157]
Accessibility	Appropriate access to software systems	[217]
Accountability	Moral accountability of people and organisations that develop software	[89, 90, 177]
Accuracy	Moral obligation to solve lack of accuracy of a system; Metrics of accuracy	[217]
Anonymity	Collection of data that can be used to identify persons	[71, 168, 167]
Applicability of Code of Ethics	Code of Ethics in Software Engineering is too general to guide software engineers in ethical decision-making	[9]
Autonomy	Autonomy of computational artefacts; Individual's freedom to make voluntary and informed choices; Moral autonomy as part of computing curricula	[90, 117, 154, 74, 177]
Care	Technically assisted care changing the relationship of elderly with themselves and the world in an undesired way	[74]
Code of Conduct	Code that comprises ground rules for communications between participants in open source projects	[179]
Code of Ethics	Code that comprises guiding ethical principles aiming to aid the software professional in ethical decision making	[157, 134, 137, 96, 113, 112, 7, 122, 111, 183, 115, 218, 117, 154, 118, 14, 9, 121, 6, 119, 120, 108, 136, 135, 114, 85, 106]
Competence	Ability to utilise software apparatus; Ability to meet the social and individual needs of stakeholders and societies; Research participants' competence	[107, 117, 166, 167]
Computer Abuse	Unethical or unlawful usage of computers or the cyberspace	[148, 142, 87, 51, 163]
Confidentiality	Unethical control of information about an individual or an organisation provided to an entrusted party	[131, 170, 219, 166, 167]
Conflict of Interest	Compromised decision-making due to offers of financial or other considerations	[131, 219]
Culture	Technical departments' culture may affect ethical considerations in the conduct of research	[165]
Data Access	Patient's access to detrimental or unjust information which may lead patient distress; Unauthorised data access	[70, 219, 123]
Data Collection	Unethical data collection (e.g. without user's knowledge)	[70, 71, 110]
Data Falsification	Falsifying and deceitfully selecting data in order to manipulate and stupefy people	[51]
Data Interpretation	Usage of interpretation dependent data for software design decisions	[87]
Data Monitoring	Passive or active monitoring of (medical) patients; Being under (permanent) surveillance	[71, 51, 102]
Data Ownership	Ownership of collected data	[18]
Data Protection	Secure data connection and transmission	[17]
Deliberately Creating Unethical Code	Purposely creating unethical code	[96]
Digital Addiction	Obsessive and problematic usage of digital media	[59, 143]
Digital Divide	Poor or uneven distribution of computers and unequal availability of information; Access to Active and Assisted Living systems	[141, 51, 148, 74]
Disappearance of Tradition	Disappearance of tradition, declining national pride, cosmopolitanism, cluttering mother tongue with foreign words	[51]

Discrimination	Equality of opportunity	[131, 80, 219]
Diversity	Considering diversity in hiring	[159]
Embedded Values	Embedded (moral) values in technologies	[88, 57, 95, 15, 54]
Encryption	Secure communication of (secret) information	[131, 219, 148]
Erroneous Data	Faulty, uncertain, and/or false data	[51]
Espionage	Espionage and industrial spying	[51]
Ethical Awareness	Unintended harm by well-intentioned software practitioners; Awareness of ethical issues in the workplace	[122, 162]
Ethical Decision-Making	Ethical decision-making as part of computing, software engineering, computer engineering curricula; Ethical decision-making of software engineers; Ethical decision-making of individuals that engage in software piracy	[152, 96, 155, 140, 9, 163, 173, 46]
Ethical Decision-Making for AI	Ethical decision-making of artificial general intelligence; Ethical decision-making of autonomous systems	[64, 62, 65, 63, 56]
Ethical Leadership	Ethical leadership of virtual teams; Ethical behaviour of the agile project manager	[100, 101, 97]
Ethical Neutrality	The belief that software development is an ethically neutral activity	[185]
Ethical Pluralism	Region-unique ethical issues; Ethical pluralism as part of computing courses	[150, 154]
Ethical Requirements	Ethical requirements in healthcare	[71]
Ethical Sampling Techniques	Unethical sampling techniques in empirical software engineering	[169]
Ethical Sensibility	Cultivating ethical sensibility as part of computing curricula	[154]
Ethical Sensitivity	Lack of ethical sensitivity among software developers	[111]
Ethical Validation	Ethical validation of statistical models for data analytics systems	[86]
Ethics in Software Design	Embedding ethical considerations into software design practise	[90, 58]
Fairness	Fairness as a characteristic of an ethical agile project manager	[97]
Green Software	Green software engineering and software development; Green Code of Ethics for software engineering and Professional Practice	[120, 82, 81]
Hacking	Hacking as part of a computer course on ethical issues	[148]
Honesty	Honesty as a characteristic of an ethical agile project manager	[97]
Informed Consent	Usage of informed consent in the process of selling software artefacts in order to help the prospective buyer understand the quality of the software; Informed consent from subjects in experiments; Informed consent from patients	[92, 164, 70, 170, 168, 166, 119]
Integrity	Software professionals need to act with integrity in the decision-making process	[117]
Intellectual Property	Intellectual property (rights) and ownership of computer software code	[128, 131, 127, 126, 110, 124, 219, 148, 142, 87, 53, 163, 129, 173, 103]
Invisible Abuse	Intentional misuse of computer and related resources using invisible operations	[98]
Irreversibility of Technical Impacts	The potential irreversibility of technical impacts	[121]
Job Loss	Decreasing number of jobs due to total informatisation	[51]
Justice	Key concept of any code of ethics; Social justice	[117, 142, 87]
Lack of Awareness	Lack of ethical awareness among computing students	[156, 220, 133]
Lack of Code of Ethics	Lack of a moral and ethical code for requirements engineering	[14]
Lack of Documentation	Ethical considerations regarding documentation of work	[160]

Lack of Research Policy	Lack of research policy in empirical software engineering	[171]
Lying	Lying or relucting to transmit negative information concerning a software project	[84, 83]
Machine Ethics	Asserting or denying machine ethics	[184]
Malleability of Technology	Technology can be shaped by its developers and users according to their own moral compasses	[121]
Measurement-Related Ethics	Covering measurement-related ethics in computing curricula	[219]
Misusing Human Participants	Monitoring and regulating the conduct of software engineering research involving human participants	[165, 172]
Oversupply	Cognitive Human Resource systems that predict the need for certain types of jobs based on past knowledge may lead to supply bubbles	[80]
Organisational Ethics	Spreading ethical values through a software organization	[99]
Plagiarism	Plagiarism among computing students; Plagiarism on StackOverflow	[147, 151, 146, 125]
Privacy	The bounds of governmental access for the public good; Ethical handling of personal (patient) data; Privacy as part of computing curricula	[131, 21, 70, 74, 71, 72, 73, 17, 66, 78, 76, 141, 18, 75, 77, 79, 19, 148, 142, 87, 53, 102, 51, 123, 163]
Quality of Life	Systems that degrade the quality of life (of users)	[53, 123]
Reliability	Data loss or unavailability at optimal times of software systems	[17, 123]
Research Ethics	Covering research ethics in computing curricula	[219]
Respect	Respectfulness as a characteristic of an ethical agile project manager	[97]
Responsibility	Ethical responsibility of software developers, software professionals, and the discipline itself; Ethical responsibility of granting Freedom Zero; Responsibility of actions of autonomous systems; Understanding responsibility	[104, 107, 180, 65, 90, 182, 122, 111, 140, 183, 98, 9, 186, 160, 100, 108, 153, 221, 165, 67, 136, 135, 185, 159, 141, 158, 56, 133, 109, 105, 105, 51, 123, 114, 139, 46]
Risks	Ethical risk of software (project) failure	[219, 94, 123, 148]
Safety	Usage of a mobile phone in unsafe environments can endanger the user by attracting undesirable attention towards the phone or its data	[17, 74]
Security	Ethics of computer, user, and data security; Ethics of attacking a system	[74, 131, 145, 181, 72, 93, 110, 17, 79, 219]
Social Embedding of Software	Ethical concerns that arise from the social embedding of software	[55, 52]
Social Implications of Technology	Social impacts of technology	[219]
Social Influence	Social influence that affects individual decision-making in buying or using pirated software	[173]
Software Failure	(Long-term) Risks of software failure	[51, 98]
Software Piracy	Software piracy; Individual's ethical standard that results in using pirated software	[175, 51, 173, 50, 176, 174]
Software Quality	Creating software that meets requirement specifications	[131, 177, 110, 219]
Software Risks	Any threat to the development and delivery of a quality product including social, professional, and ethical risks	[94]

Software Vulnerability	Ethical aspects of producing potentially flawed software; Including automated analysis tools in the ethical software development process; Security vulnerabilities	[91, 93, 114]
Sustainability	Design and development of Sustainable Organic Local Ethical software	[85]
System Access	Unauthorised use of computers and communications	[131]
Teaching Ethics	Increase ethical awareness among computing students	[134, 156, 137, 131, 20, 155, 138, 140, 130, 153, 136, 135, 141, 149, 133, 144, 143, 163, 139, 46, 162]
Transparency	(Ethical) Transparency of open source software; Transparency of security risks in understandable terms; Transparency of negative information in software projects	[178, 17, 158]
Trust	Unfounded and unethical trust in computer systems	[51, 79, 98]
Unethical Application Actions	Unethical application behavior such as hidden application costs	[76]
Unethical Web Methods	Unethical methods of Search Engine Optimization	[69]
Virtual Harm	Exposure to harmful websites and harmful information; Child abuse activities across social networking facilities; Physical and virtual harm in cyberspace	[68, 66, 67, 148]
Vulnerability	Vulnerability of participants in empirical software engineering	[167]
Whistle-Blowing	Making revelations in order to call attention to threats to the public interest	[131, 160, 161, 219, 219]
Workplace Ethics	Unethical behaviour on the (virtual) workplace	[100, 162, 101, 97, 179]

C Grey Literature Review - Data Extraction Form

Metadata

Identifier	Metadata item	Description
D1	Extraction date	Date of data extraction.
D2	Publication date	Publication year of the study.
D3	Title	Title of the study.
D4	URL	URL of the news article.
D5	News-outlet	Source of the article.
D6	Article Type	Type of the article (e.g. feature article).
D7	Author(s)	Author(s) of the study.
D8	Authors' affiliation	Journalism, academia, industry or not clear.
D9	Country/region of affiliation	What country or region is affiliated with the news-outlet?

Ethical awareness

Identifier	Data item	Description
D10	Need for ethical awareness	Does the article mention a need for ethical awareness (yes or no)?
D11	Ethical issue(s)	What ethical issues are stated?

D Grey Literature Review - Ethical Issues

Ethical Issue	Description
Accountability	Accountability of AI decision-making
Code of Ethics	Code of Ethics for software creators
Cybercrime	Creating and distributing malicious software
Data Collection	Unethical data collection and illegal data trade
Data Monitoring	Monitoring (school) children
Digital Divide	People’s access to the internet
Disappearance of Tradition	Preserve endangered languages
Discrimination	Racially biased software/AI
Diversity	Diversity on the work-floor; Diversity in hiring
Economic Inequality	Economic inequality caused by technology and automation
Green Software	Developing software that is harmful to nature; Eco-friendly software
Ethical Decision-Making	Ethical decision-making of autonomous systems
Fake News	Spread of fake news on social media
Flawed Software	Effects of flawed software
Freedom	Free software
Hacking	Hacking into computer systems to alter or collect data
Intellectual Property	Copyright and intellectual property in e-commerce
Job Loss	Loss of jobs due to technological advancements (replacement by robots)
Lack of Awareness	Lack of exposure of computer science graduates to ethical issues; Lack of ethical awareness in software organisations
Lying	Embedding corrupt testing mechanisms in software; Lying on software projects
Net Neutrality	Freedom of internet users
Plagiarism	Plagiarizing students
Privacy	Creating software that violates people’s privacy
Responsibility	The responsibility of software creators for the ethical implications of software
Safety	User safety against the harmful effects of software
Security	Creating well-secured; Cyberwarfare
Software Failure	Software that does not meet the requirements
Software Piracy	Pirating software and selling illegal software
Software Quality	Intentionally creating low-quality software
Transparency	Transparency of technologies and software
Virtual Harm	Exposure to harmful websites and software
Vulnerability	Software vulnerabilities that can be exploited
Wealth Bubble	High paying software jobs that cause financial segregation
Workplace Ethics	Toxic work-floor culture