



Ethical Issues in Software Requirements Engineering

Seblewongel Esseynew Biable ^{1,*} , Nuno Manuel Garcia ² , Dida Midekso ³ and Nuno Pombo ²

¹ Department of Information Systems, College of Computing Science, Debre Berhan University, Debre Berhan P.O. Box 94, Ethiopia

² Instituto de Telecomunicações, Universidade da Beira Interior, 6200-001 Covilhã, Portugal; ngarcia@di.ubi.pt (N.M.G.); ngpombo@di.ubi.pt (N.P.)

³ Department of Computer Science, Addis Ababa University, Addis Ababa P.O. Box 1176, Ethiopia; dida.midekso@aau.edu.et

* Correspondence: sebess2011@gmail.com; Tel.: +251-913-25-6851

Abstract: Abstract **Context:** Ethics have broad applications in different fields of study and different contexts. Like other fields of study, ethics have a significant impact on the decisions made in computing concerning software artifact production and its processes. Hence, in this research, ethics is considered in the context of requirements engineering during the software development process.

Objective: The aim of this paper is to discuss the investigation results regarding ethical problems of requirements engineering processes by taking sample software developing companies and exposing existing research gaps.

Method: This research uses interviewing, focus group discussions, purposive sampling, and qualitative analysis research methods. **Result:** This research finds an absence of industry practices, professional responsibility code of conduct standards, and other guidelines within companies when integrating ethical concerns of software during requirements engineering. It also indicates that almost all companies have no identification methods and checking mechanisms for ethical concern considerations. Furthermore, the major identified ethical concerns are classified into six categories as requirements identification problems, quality-related problems, carrying out unpermitted activities, unwillingness to give requirements, knowledge gaps and lack of legal grounds/rules for accountability.

Conclusion: From the findings of this research, it can be concluded that, in the case software companies, there is no specific method for identifying ethical concerns. Additionally, there are no standards and guidelines used within the companies. This implies the need to overcome the existing and emerging ethical issues of requirements engineering.

Keywords: software engineering; ethical concerns; requirements engineering; software development



Citation: Biable, S.E.; Garcia, N.M.; Midekso, D.; Pombo, N. Ethical Issues in Software Requirements Engineering. *Software* **2022**, *1*, 31–52. <https://doi.org/10.3390/software1010003>

Academic Editor: Stefan Wagner

Received: 25 January 2022

Accepted: 15 February 2022

Published: 18 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Motivation for the Research and Statement of the Problem

Ethics have broad applications in different fields of study and different contexts [1]. Like other fields of study, ethics have a significant impact on the decisions made by end-users, customers, professionals, and companies concerning software artifact production and its processes [2]. There are different meanings of ethics [3]. Diverse scholars and philosophers define ethics in a variety of ways. Vallor et al. [1] define ethics as a philosophical discipline of study, as “[...] a systematic approach to understanding, analyzing, and distinguishing matters of right and wrong, good and bad, and admirable and deplorable as they relate to the well-being of and the relationships among sentient beings”. The word ‘ethics’ originates from the Greek word *ethikos*, which refers to the ethos, meaning a distinguishing feature, essence, or insolence [3].

Directly or indirectly, there is a close interrelationship between ethics and software. The work of software engineering is intimately associated with human beings. This makes the field more dependent on the ethical knowledge, experiences, and professional competencies of software engineers [3]. Accordingly, software-related problems could have negative

impacts on stakeholders. In many cases, the origin of numerous technological failures comes from software-related concerns [4]. The failure of software systems can have a disastrous impact on living and cause damage to the public [1]. Since the early days, software project failures have been a well-known problem. Although several pieces of research have been published regarding software quality, and many improvements have been introduced in the software development process, software project failure is a critical challenge for the software development community. Currently, most software projects may not fail in economic terms. Instead, they produce wealth in the short term but, in the medium/long term, they can cause substantial financial losses, as well as societal damages due to the scarcity or even absence of compliance with regulations, policies, rules, and ethical concerns [4]. Many studies assessed software projects' success/failure rates [5–7]. These studies showed the presence of major problems within different sections of industries regarding the development of successful software projects [8]. Furthermore, they also indicated that, even though the failure/malfunction of software systems tends to be due to project management issues, it is also related to humans, clients, and other stakeholders' factors [4]. In most cases, such malfunction consequences derive, directly or indirectly, from the fact that software engineers pay no attention to their professional duty through performing unethical professional behavior [1,9,10].

Explicitly identifying and determining ethical concerns [4] is at the heart of the System Development Life Cycle (SDLC), in order to develop useful and valuable software artifacts for stakeholders and for general society. Regardless of different software development models, there are fundamental ethical requirements that must be clearly considered in each phase of the SDLC [4,11]. By means of clearly considering ethical concerns during SDLC phases, particularly in the requirements engineering phase [12,13], developers of software systems can tackle and resolve the contradictory and necessary stakeholder values, as well as the larger system context [6].

During software development, ethical requirements should be considered to clearly have an expressive and justified impact on development. Certainly, ethical concerns can and must take part in significant functions within the early phases of SDLC, i.e., the requirements engineering phase [4]. Requirements engineering is a crucial phase of SDLC, insofar that its success determines the success of software projects [12,14]. Although the implementation of a software system is performed accurately, the software product will fail in the long term if it does not meet the adequate requirements [14]. Requirements engineering is fundamental and, at the same time, the hardest phase of SDLC [12,14] since it decides precisely what to build and ascertains the specification and validation of detailed technical and non-technical requirements [14]. Congruently, this phase is prone to be the most difficult phase for the introduction of ethical concerns. Indeed, requirements engineering is usually carried out without explicit consideration of ethical concerns [12]. Accordingly, requirements engineering has been exposed to many ethical violations [15–21], for example holding up toxic requirements [17], requirements that have long-term and dangerous unidentified effects. Moreover, improper requirements [18], deception requirements [19–21], inadequate and/or incomplete requirements, ignoring relevant aspects for the business model, insufficient specification, and unnecessary complexity and functionality, are among the other ethical violations [15,16].

There are many previous cases in which requirements engineering became unethical. The literature review identified three examples [17–19]. One of the best examples is the Volkswagen Turbo Diesel Injection (TDI) 2009–2015 models [19–21]. Volkswagen deceived the emissions testing system by building its diesel cars to look less contaminating than they really were [19]. The other case in which unethical practices were observed due to the presence of improper requirements were incorrect radiation dose calculations at the National Oncology Institute in Panama [18]. Furthermore, the loss of communication between the Federal Aviation Administration (FAA) air traffic control center and airplanes in Los Angeles can be taken as another instance where ethical concerns were not considered during requirements engineering processes [17]. Moreover, there are many cases in

which software brought catastrophes due to the improper and inadequate requirements engineering processes [16,18,21].

This evidence reveals that developing software systems without clearly considering ethical concerns, especially in the early phase of software development life cycle phases, i.e., requirements engineering, may lead to the development of software artifacts that could fail to fulfill societal value and may cause damage to the public at large [16–21].

To mitigate this, ethical concerns must be considered in requirements engineering processes. In the requirements engineering phase, ethical principles may lead to a better engagement of stakeholders, who are mainly end-users, clients, and technical staff members.

Therefore, the motivation of this research initiated by the researchers was to investigate the ethical concerns of requirement engineering, which should be considered during the requirements engineering processes by software developers. Therefore, this opens the door for further investigations to resolve the identified problems.

2. Literature Review

2.1. Software Engineering

Currently, software engineering has acquired the role of one of the best professions internationally [22]. With the development of the field of software engineering, the importance of value-oriented software has dramatically increased towards the application of new methods and techniques to address ever-increasing business demands, stakeholder requirements, market situations, competitiveness, and the profitability of businesses and organizations [23]. Whittle et al. [24] strengthen the significance of the value-oriented design of software systems through their investigations of real-life projects. As they stated, value-oriented software development assists in integrating human values in software artifacts. Even though there are differences between values and ethics [24], there is also a strong interrelationship between human values and ethical values [25]. As a result, considering ethical values in software engineering is essential for assisting software engineers in the product they produce [26].

2.2. Requirements Engineering

Requirements engineering is part of engineering and focuses on the actual goals, functions, and constraints of a system [27]. It is the process of eliciting, analyzing, specifying, validating, and maintaining the requirements of a system [12,27]. It entails identifying stakeholders' needs; recognizing the context of the requirements; representing those requirements in an easily understandable manner; and negotiating, validating, documenting, and managing those requirements. Failing to identify correct needs leads to flaws in specifications and the other components of a system [28]. Inadequately defined requirements can cause many difficult problems, which have an impact on different direct and indirect stakeholders of software systems. Requirements engineering offers suitable instruments designed for recognizing and examining customers' real needs, reviewing viability, stipulating the solution explicitly, confirming the requirements, managing the requirements, and settling on a practical solution as they are transformed into an operational system [29]. Requirements engineering encompasses many sub-processes. The major five sub-processes are requirements elicitation, requirements analysis and negotiation, requirements specification, requirements validation, and requirements management [27–31].

2.3. Software Engineering and Ethics

2.3.1. Engineering Ethics

In ancient times, ethics were mainly linked to philosophical activities. However, currently, ethics are not only philosophical theoretical facets but also the basic practical aspects of many fields of study, including engineering [3]. The study of ethics is found in many fields [1]. In engineering, ethics are defined as "the study of moral issues and decisions confronting individuals and organizations involved in engineering" [32] (p. 196). Engineering ethics are mainly inclined to the professional ethics of software engineers and

seek to identify and question engineering standards and the ways that those standards are performed in specific circumstances.

2.3.2. Software Engineering Ethics and Code of Ethics

Ethics are the greatest wisdom, which indicates how human beings continuously, presumably, know how best to live [1]. Ethics refer to every deliberate act that affects the lives of others, either positively or negatively [11]. Considering ethics in engineering is important since it provides safe, useful, and technological products and services [3]. Software engineering ethics can describe software engineers' practical and ethical decision-making activities, a group of principles, guidelines, or ethical imperatives [11]. According to Quigley [33] (p. 266), in software engineering, ethics are defined as “[...] a code of professional standards, containing aspects of fairness and duty to the profession and the general public”.

There is a high interrelation between software engineering and ethics. Software engineers build software artifacts, starting from simple artifacts such as, for instance, application programs, and progressing to very large systems using sophisticated software programs, such as safety-critical systems, including software for rockets, airplanes, and cars, among others. Such kinds of systems have either positive or negative vulnerable impacts on society, as well as on the environment [1,32].

Software engineering considers itself principally as a technical profession, which entails developing software products and services. However, nowadays, directly or indirectly, human beings depend on software systems for their day-to-day activities. During the software development process, software engineers make critical decisions on behalf of the system. These decisions have a great impact on the clients, society, and the environment. This hampers the work of software engineering and leads to subjective decision judgments. It also implies that software engineers' technical decisions should be guided by ethical decisions concerning societal health, safety, and wellbeing [11]. To develop useful and harmless software systems and/or services, software engineers should make careful decisions, namely by investigating both technical and ethical aspects.

Software engineers should have sound ethical standards on professional and personal aspects [4,10]. The work of software engineers has a direct impact on the well-being of society. There is nothing outside the people; their work is intimately associated with the welfare, safety, and health of human beings. Unlike medical professionals and lawyers, who primarily focus on their clients, the primary interest and obligation of software engineering professionals is the “public interest”. He/she must have concerns about employers, clients, professionals, and other parties [34].

In the field of software engineering, the software code of ethics requires high consideration in research [4,9]. Considering the ethical concerns of the software development projects will increase the public's trust and the professional image of the software artifacts, as well as interior and exterior communication among different stakeholders and their satisfaction. With these principles in mind, there should be an explicit guideline/framework that entails which things should be considered and integrated regarding ethical concerns in each phase of the software development project. Furthermore, expressing ethical concerns in the software development project will help as a tool for software developers to soundly recognize and uphold the code of ethics in their profession [10]. There are many codes of ethics in the software engineering field [11,35–39]. However, in practice, ethical concerns are not explicitly considered during software development [4,9,39].

The software engineering code of ethics (SWECODE) [9] is one of the areas of software engineering that needs increased research consideration. It can take a significant part of the practice and has an enormous impact on the daily lives of human beings. The level of harm on human beings because of defective software equipment greatly depends on the aptitudes of software engineers to pursue high performance in software engineering. The research on practices of the software engineering code of ethics during the software development process is understudied [40].

Many organizations, such as the Institute of Electrical and Electronics Engineers (IEEE), Association for Computing Machinery (ACM), the British Computer Society, and others have developed codes of ethics for computing professionals [41,42]. Nevertheless, these accessible codes of ethics offer a broad roadmap for software engineering professionals regarding one specific facet: professional ethics. These codes of ethics lack breadth and depth. They do not assess and go further in terms of detailed ethical concerns for software artifacts and development phases [10]. Most of these software engineering codes of ethics have become challenging to implement on a day-to-day basis of software development, since they are at a high level, encompassing a comprehensive range of ethics [40].

2.4. Ethical Considerations in Software Development

Currently, the use of software is ubiquitous. It is possible to find the implementation of software in virtually all day-to-day activities. Kids' electronic toys, smartphones, televisions, refrigerators, cars' onboard computer systems are just a few examples of software applications. Software plays a crucial function in offering business information solutions for enterprise resource planning, enabling, organizing, and protecting the activities of large, medium, and small-sized organizations. Software engineers, who are responsible for the development of software systems, have an important perspective role to put in place good practices. This means that software engineers should be highly committed to making software artifacts usable and beneficial for the stakeholders and society at large [4].

Software developers must be responsible for implementing technical mechanisms, which shape the corresponding organization's ethical dimensions [43]. Substantial software is being used daily and the situation becomes critical when it affects users negatively. For the duration of software SDLC, developers face critical ethical decisions to guarantee a minimal impact on users and stakeholders. There are numerous reasons why it is decisive to add ethical practices and considerations in decisions regarding any software development project [37]. Software developers should carefully ponder the incorporation of technical and ethical dimensions in the decisions of software development projects, seeking to protect the software, while preventing prospective harm to others.

Software developers are responsible for their profession and the public at large within the scope of their profession. They should have the willingness to hold to their professional code of ethics as part of their contributions to society by developing high-quality and valuable software [9].

Software engineers are accountable for both the result of their software artifacts and the faults of such works. They have the duty to identify and understand the vulnerability of such hazardous requirements, making sure that they are securely operated. Clients cannot recognize the impact of those requirements. The software developer/engineer is responsible and accountable for caring for those defective software requirements and producing quality software [9,44]. During software development decisions, clearly considering ethical concerns is very useful to integrate both technical and ethical issues of the software artifacts. As stated above, software engineers are accountable for their profession and society at large [40]. Moreover, they have the responsibility to do the right things for both clients and society. As a result, a careful investigation of ethical concerns in software engineering is necessary.

2.5. Related Work

Ethical issues have multidimensional angles. Different researchers have investigated ethical concerns of software systems in diverse aspects. For instance, various scholars look at the ethical issue with respect to socio-political-related issues, human value aspects of the software artifacts, and issues related to digital addiction. Additionally, scholars have explored the ethical concerns of software development processes, software engineering codes of ethics, and professional practices. In this article, the authors reviewed some of these.

ACM/IEEE jointly proposed eight principles of the Code of Ethics and Professional Practice [35]. The ACM and IEEE computing society, the two most well-known computing societies, carried out a wide-ranging review on the software engineering code of ethics, modifying the third version and published version 5.2 codes of ethics [35]. The proposed code holds information interrelated to the behaviors and decisions made by different stakeholders. The principles find ethically accountable relationships, in which individuals, groups, and organizations participate, alongside the primary obligations within these relationships. Table 1 depicts the ACM/IEEE code of ethics.

Table 1. The Eight IEEE/ACM Codes of Ethics [36].

No.	Principles	Meaning in Short
1	Public	Software engineers shall act consistently within the public interest.
2	Client and Employer	Software engineers shall act in a manner that is in the best interests of their client and employer consistent within the public interest.
3	Product	Software engineers shall ensure their products and related modifications meet the highest professional standards possible.
4	Judgment	Software engineers shall maintain integrity and independence in their professional judgment.
5	Management	Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance.
6	Profession	Software engineers shall advance the integrity and reputation of the profession consistent with the public interest.
7	Colleagues	Software engineers shall be fair to and supportive of their colleagues.
8	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 shown in Table 1, the code addresses many issues relevant for software engineers, managers, practitioners, and others with respect to the public, clients, employers, products, judgments, the profession itself, colleagues, and oneself. However, the present ACM/IEEE software engineering code of ethics has become challenging to implement on a day-to-day basis, since it is at a high level [10,40].

To overcome the challenges observed on the ACM/IEEE code of ethics, AlAmmar et al. [9,40] conducted a research study on SWECO by taking one software project development company known as SADA as a case study in Saudi Arabia. The research investigated the practices of SWECO by software developers. The researchers used interviews, questionnaires, and secondary sources of information. By using these tools, the researchers analyzed SWECO and proposed new classifications of it according to the software development life cycles. The researchers took the four principles of the ACM/IEEE code of ethics (public; client and employer; product; and judgment) and classified them based on the requirements engineering, design, code, test, and the SDLC's maintenance phase, and proposed new SWECO classification. The proposed framework was validated by professionals. The result of this research study indicated that there was no proper ethical guideline for the software development phases. Moreover, there was a lack of awareness and knowledge of the available ACM/IEEE code of ethics. This research worked as an opening door to adapt the ACM/IEEE codes of ethics for the need to consider ethical practices explicitly in the software development process. However, the researchers did not make any modifications to the ACM/IEEE code of ethics. Simply they directly took the ACM/IEEE codes of ethics and classified them according to lifecycle phases. This does not make the code detailed and clear. Moreover, they did not have any justification for why

they selected only four principles and left the other four out. Furthermore, the research was carried out for a single company. To enhance the validation of the research results, there should be more case studies in the research area. Generally, the work of AlAmmar et al. [10,45] showed that further investigation should still be carried out to analyze the greater ethical concerns in SDLC phases in a more detailed manner.

On the other hand, Thew and Sutcliffe [45] investigated three socio-political issues of requirements engineering, specifically values, motivations, and emotions. They identified the most important value concepts, motivations, and emotions for a requirements analysis. As for the elicited requirements techniques, researchers employed interviews, workshops, prototyping, and telephone conferences. To evaluate the proposed method, researchers used case study and expert evaluations. The researchers proposed a method for analyzing socio-political issues in the requirements engineering processes. In doing so, the method complements the analysis of non-functional requirements. Furthermore, the proposed method brought in new concerns found on the requirements engineering processes by focusing on these three parameters. The results showed various issues regarding the need for appropriate guidance for managing socio-political issues in the requirements engineering process. However, Thew and Sutcliffe's [45] research only focused on values, which are relatively constant personal beliefs, and motivations, which are psychological concepts associated with personal characters, and these may be observed as stakeholders' long-term goals in requirements engineering. From this, we observe two limitations of this research. Firstly, the research did not look at the values and motivations of technical persons, i.e., software engineers. Secondly, the ethical values of stakeholders as well as software engineers in the context of requirements engineering has not yet been studied.

Rashid et al. [46] assessed the ethics of software engineering in a digital world. The authors stated that software engineering exercises and practices should explicitly integrate ethical concerns. We are living in a globally growing digital world, in which various technologies are pervasively accessible. This enhances ease of use and collective communication. However, such technologies also have negative impacts; they raise many ethical concerns, such as privacy, monitoring, and data protection, to name a few. The digital world has many benefits for the public; nevertheless, it also causes numerous unethical activities. The authors also stated this situation, i.e., the problem of a single technology having the ability to bring about both good and bad, as a dual-use dilemma. To demonstrate the concept of dual-use dilemma in the context of the digital world, the authors took three sample applications: Google maps; chat rooms and peer-to-peer networks; and Facebook. They stated that the dual-use dilemma emerges because the same single technology has, every so often, the possibility of being used for harm as well as for good. As stated by the authors, instead of a simple comparison of the benefits and drawbacks of technology, software engineers should investigate critically and deeply the damaging ethical concerns of individual software engineering technology. Moreover, they should search for alternative solutions, approaches, and/or methods for the dual-use dilemma for the corresponding technology.

Similarly, Cemiloglu et al. [47] investigated the ethical requirements for addictive technologies, in the case of online gambling. As a research methodology, they used Reno Model I-V. Through their study, they identified three categories of ethical requirements for gambling: creating an environment that supports informed choice; monitoring user data to identify risk factors; and introducing measures to tackle problematic behavior. As a contribution, this research work may give initial insights for software developers regarding the ethical requirements for addictive technology, specifically for gambling. Nevertheless, there are some limitations of the research. Firstly, the work did not give focus on how such addictive technologies should be developed. Prior to making it available for use, the ethical requirements of software should be assessed and investigated. Secondly, the research only focuses on one addictive technology: gambling. To address ethical requirements for many software technologies once, more work should be conducted to come up with a comprehensive reference model as well as a framework or guideline.

Additionally, Ali et al. [48] explored digital addiction labels as requirements of emerging technology. The researchers explore the duty of software industries to make their users aware of the possibilities of the addictive natures of software products and artifacts. Similarly, Alrobai et al. [49] examined the factors of digital addiction with respect to requirements engineering. The researchers used an empirical investigation of the literature to differentiate between the factors that cause digital addiction. To validate their findings, they used discussion forums on the theme. The researchers offer the design of a mind map and a model for the purpose of elucidating digital addiction problems in the context of requirements engineering. The research findings showed that, directly or indirectly, the cause for the existence of problems, including ethical problems, is intimately associated with the requirements engineering phases of SDLC [49]. Both Ali et al. [48] and Alrobai et al. [49] suggested the need for more work generally in software engineering, particularly in requirements engineering ethics to address ever-increasing technological advances and their associated challenges in the area, especially since requirements engineering is the place in which users' needs and values are collected and analyzed.

Whittle et al. [24] discovered the means of the integration of human values into software development processes and proposed new methods for considering those values. To investigate, the researchers used three real-life projects. The researchers used exploratory research methodology. From their investigation, the researchers observed that, unlike functional and non-functional requirements, which represent the "what" and 'how' of a software system, respectively, values represent the 'why' of a software system. As they stated, as with quality requirements of software systems, great consideration should be given to human values such as respect, empowerment, transparency, and community responsibility, among others. Technical and non-technical solutions must go together jointly to achieve solutions for the integration of human values in software development processes. Nevertheless, the research work of Whittle et al. [24] mainly focuses on human values rather than ethical values.

Paech and Kurt [50] argued the need for understanding users' views of software. Users could not discuss software in the same way that developers could. While users can discuss the external view of software, developers must compare features of the software, identify compliance to the software usability, and differentiate the constraints including ethical values of it. The researchers proposed an intermediary communication language (user view language) as a common ground for discussions among users and developers. The emphasis of the proposed tool is to help users to more elaborate on the external view of software as well as to assist developers, in order to better understand their views and to map them into technical concepts. Nevertheless, this proposed work is not focused on ethical values. Additionally, the researchers are not clearly showing the methodology used to develop the proposed work.

Lurie and Shlomo [4] proposed an ethical framework that amalgamates software engineers' professional knowledge with their ethical responsibilities. The proposed framework has three parts as SDLC phases, in which questions will be asked and answered by different stakeholders, and a risk analysis. The main purpose of the research was to combine software development lifecycles with the professional responsibilities of software engineers. As the reviewers understand, the proposed work of Lurie and Shlomo [4] has both strengths and weaknesses. As a strength, instead of having a standalone software engineering code of ethics, this research demonstrates the possibilities of integrating both technical knowledge with the professional code of ethics, regardless of the type of development methodologies. However, as those authors mentioned, even though the framework is abstract at a high level, the proposed work did not suggest the basic questions that should be answered at each level. For example, we can see the yes/no questions that were prepared for the requirements engineering phases. As indicated in their research, there are five yes/no questions. These include questions concerning requirements decision-makers, functional requirements, customers resistance, the existence of special negative consequences and contradictory issues, and requirements prioritization, respectively. Nevertheless, by an-

swering only these questions, it is difficult to address the ethical concerns of requirements engineering. Additionally, none of these questions deal with nonfunctional and quality requirements, including ethical requirements. Moreover, the researchers do not mention the methods utilized to propose this framework as well as to validate the proposed framework.

Aydemir and Fabiano [10] introduced an ethics-aware software engineering method which captures, analyses, and reflects stakeholders' values regarding software artifacts and software engineering processes. The proposed work aims to help stakeholders by analyzing the ethical concerns of software artifacts and software engineering processes, their pertinent value, and in danger entities such as developers and users. The proposed method involves all stakeholders of software development projects. Additionally, the method is proposed for all development process models as a plugin. In addition, it has five separate phases including articulation, specification, implementation, verification, and validation. Additionally, the method is proposed with four different enablers: awareness, conscious valuing, ethics knowledge, and transparency. The proposed approach is interesting since it introduces new concepts regarding ethics-aware software development approaches. However, the work covers a wide range of cases. It deals with ethical concerns in the software engineering discipline along with its focus on the entire stakeholders of a software system, including developers, users, business organizations, and others. Additionally, the proposed work discusses all life cycles of software development. These and others make the proposed approach too broad. To achieve depth, ethical case-specific development phases and specific target groups should be considered. Especially at the requirements engineering phase, since it is the foundation of the others, a list of detailed ethical concerns, with appropriate resolving mechanisms, should be provided.

Aside from the above-related works, there are lots of studies that were conducted on social networks and requirements engineering [51–54]. From these studies, we can understand the necessity of further research and investigations on requirements engineering ethics in a variety of aspects.

As we see from all the above reviews, scholars investigated ethical aspects with different viewpoints. In one or in another, all of them showed the need for more works to investigate ethics requirements, generally in software engineering, and particularly in requirements engineering. Nonetheless, all of them did not explicitly inspect ethical value concerns of the requirements engineering processes in a detailed manner. Therefore, the main task of this research was to investigate the ethical concerns of the requirements engineering processes in a detailed manner. Using the results of this research, our next work will address the identified problems through developing an ethical framework that helps software developers to give a clear map concerning the integration of ethical concerns in the requirements engineering phase of SDLC.

3. Research Methods Used

3.1. Introduction

This research investigates software companies that operate in Ethiopia. Among the available software companies, we selected the participating companies purposely based on the richness of the information they have and their willingness to be involved in this research work. These companies ranged from small (having less than ten employees) to large companies (having more than 100 employees) regarding development tasks. They also have between 5 and 29 years of work experience.

Almost all the selected companies are engaged in the provision of affordable information technology (IT)-based solutions to any business enterprise and customers in E-Government, enterprise resource planning, human resource management, inventory and stock management, medical software, customer relationship management, mobile applications, etc. In this section, we discuss the preparation of data collecting tools, participants' demographic information, data collection procedures, and data analysis techniques.

3.2. Preparation of Data Collection Tools

Firstly, about fifteen interview questions were developed. Twelve professionals in the field of computing evaluated these questions to check the questions' validity. These professionals are PhD holders and professors. Constructive and essential comments and feedback were obtained from these scholars. The questions were modified using comments and feedback from them. Finally, thirteen edited and modified interview questions and five focus group discussion questions were used to gather the data.

3.3. Research Site and Participant Selection

This research investigates the ethical concerns of requirements engineering by taking seven cases of software companies that operate in Addis Ababa, the capital of Ethiopia. Among the available software companies, we selected the participating companies using purposive sampling based on the richness of the information they have and their willingness to be involved in this research work. These participating companies ranged from small (having less than ten employees) to large companies (having more than 100 employees) regarding development tasks. For confidentiality purposes, we used codes for the participating companies. Except for one company, which has five years of experience, most of them are well-experienced and have more than eight years of experience. The oldest company, company B, has twenty-nine years of experience in developing software products and services. In general, the years of experience ranged from five, only one company, to twenty-nine years. Companies A, B, C, D, and F have been in the industry from sixteen to twenty-nine years. The youngest company, company E, has only five years of experience. On the other hand, company G has more than ten years of experience.

Table 2 gives information about the involved software companies and participants.

Table 2. Software companies and participants.

Company Code	Years of Experience	No. of Participants	Participants' Job Titles	No. Employees
A	19 years	3	Chief Company Manager Requirements Engineering Team Leader Requirement Engineers	9
B	29 years	7	Chief Company Manager Requirements Engineer Consultant and Project Manager Requirements Team Leader Requirements Engineer	628
C	20 years	3	Middle-Level Manager Software Developer	12
D	16 Years	3	ICT Director Team Leaders Developers	501
E	5 Years	2	Team Leader Software Developers	6
F	17 years	16	Director for Security Department Business Analysts Requirements Engineers	Above 3000
G	10 Years	1	Software Developer	3

As shown in Table 2, the participating companies have different sizes and years of experience. This helped us to make the research results applicable to various types of software-developing companies in terms of their size and maturity levels.

In this research, the participating companies and individuals are selected for purposive sampling. The seven participating case companies are selected in accordance with the richness of the information they have and their willingness to participate in the research. In the

research, small, medium, and large software companies are considered. After selecting the companies, thirty-five willing experts who have theoretical as well as practical experiences and knowledge of software development, particularly requirements engineering processes, were selected.

Among the 35 participants, 62.9% (22) of them were male and the remaining 37.1% (13) were female. The age of the participants ranged from 25 to 50 years old. Additionally, most participants had worked in his/her employer software companies for more than five years. Regarding participants' qualifications, 2.9% (1 participant) had a PhD, 71.4% (25 participants) had MSCs, and the remaining 25.7% (9 participants) were BSC holders in computer science, software engineering, and information systems. Table 3 depicts the demographic information of the participants.

Table 3. Participants' Demographic Information.

Variables	Category	Number of Participants	Percentage (%)
Gender	Male	22	62.9%
	Female	13	37.1%
Age	From 20–35	9	25.72%
	From 36–50	22	62.86%
	Above 50	4	11.43%
Educational Level	BSC	9	25.7%
	MSC	25	71.4%
	PhD	1	2.9%
Years of Experience	1 year	3	8.57%
	5 years	3	8.57%
	7 years	4	11.43%
	8 years	3	8.57%
	9 years	8	22.86%
	Above 14 years	14	40%

3.4. Data Collection Procedure

To answer the research questions, we designed interview and focus group discussion questions. In the beginning, we developed fifteen interview questions and five focus group discussion questions. Most of the interview and focus group discussion questions were similar. However, we used them to triangulate the results and to achieve valid and representative data from the participants.

The questions addressed ethical concerns during the requirements engineering phases of SDLC. The interview was audio-recorded, with the average length of the interview being 45 min. During the interview sessions, we used probing questions to discover relevant ethical concerns in a detailed manner. We conducted two focus group discussions with four members each. On average, each of the focus group discussions took an hour. These focus group discussions were very open and interactive sessions. Moreover, before starting the interview, as well as the focus group discussion questions, each participant agreed and signed a consent form.

The interview and focus group discussion questions were designed in accordance with the research questions. The research has three research questions. Here, we discuss each of the research questions with respect to the designed data collection tools.

The first research question investigates the practices of software companies regarding ethical concern considerations during requirements engineering processes. To carry out this, three interview questions (interview questions 2, 3, and 4) and two focus group discussion questions (focus group discussion questions 2 and 5) were designed. These questions

helped to assess the practices of software companies in their utilization of professional codes of conduct, guidelines, and the way they were exercised in day-to-day activities to take into account the ethical issues of requirements engineering.

The second research question addressed the fundamental ethical concerns that should be considered during requirements engineering processes. To answer this research question, we designed five interview questions (questions 5, 6, 8, 9, and 10) and two focus group discussion questions (questions 2 and 4). The findings from this investigation are discussed under Section 4.3.

The third research question investigates whether there are methods/techniques that companies use to identify ethical concerns during the requirements engineering processes. To gather data through this research question, three interview questions (questions 7, 11, and 12) and one focus group discussion question (question 3) were developed. The findings from this investigation are indicated in Section 4.4. Sample interview and focus group discussion questions are indicated in Appendix A.

3.5. Data Analysis

To identify the important concepts of the findings, the collected data were transcribed and read repeatedly. The data were organized to make them suitable for analysis. To analyze the collected data, four qualitative coding strategies were used. In this research, before conducting a detailed analysis, we used holistic coding. This coding strategy is used for grasping the main themes and categories that are generated from the data. This coding technique helped us to obtain the overall sense of content and to prepare part of the data for further coding and categorization processes. Moreover, we used initial coding to examine each part of the data closely. Then, we generated initial codes from each distinct section of the data. In addition to the above-mentioned strategies, by using *in vivo* coding, some important actual words of the respondents were taken as they are. Furthermore, we used attribute coding to discuss the participating software companies and individual participants' demographic information.

4. Findings with respect to the Research Questions

By using the above coding strategies, we carried out different stages of the coding process. In the first stage, we identified anchor code labels for each of the three research questions. At that juncture, using holistic coding, the overall concepts of the corpus were identified and coded with groups of data. Moreover, after catching up on the core points of each group of data, we applied a detailed analysis of each word, phrase, and sentence carefully. Then, in the second stage, relevant statements were coded, and they were put under their respective anchor codes. Additionally, some vital actual statements of the respondents were taken directly using *in vivo* coding. In the third stage of the coding process, the list of initial codes was compiled. In the fourth stage, these codes are grouped based on the anchor codes or based on the research questions. In the fifth stage, each code is depicted with its respective frequencies. In the sixth stage, categories are generated and examined to develop themes that address the research questions. Finally, we obtained major themes from the research findings. In this section, we present the major findings of the research corresponding to each theme.

4.1. Importance

The first concern was investigating the awareness of participants regarding the importance of clearly considering ethical concerns during software requirements engineering processes. Regarding this, all the participants replied that clearly considering ethical concerns during requirements engineering processes is essential. Mainly, the participants discussed two major categories. These are considering ethics in requirements engineering processes, which help to obtain high-quality software and minimize ethical concern violations. This implies that if we consider ethical concerns in the early stage of the system development life cycle, it is possible to ensure privacy and security issues, enhance software

project success, refine artifacts, and enhance ease to use, enabling the retrieval of in-depth information, and so on. Additionally, considering ethical concerns helps to minimize ethical violations. This indicates that considering ethical concerns helps us to elude deceptive requirements, protect confidentiality, respect social values, and find complete requirements, and so on. In this regard, we indicate a few respondents' responses as follows.

One respondent replied that "Requirements engineering is the first phase of system development, and it is a foundation. As a result, if there is something wrongly done in this phase, the problem will be reflected in the design, architecture, and all other phases of SDLC. Besides, the problem done in this phase will have a high impact on the other phases. As a result, requirement engineering is the foundation for other phases, and it is a very critical one." As he said, "there is one saying, 'a well understood problem is half done.' So, we have to work carefully and with great effort to get valuable and quality software."

Similarly, the other respondent said, "Of course considering ethical concerns have many benefits. In requirements gathering activities, the more we go into detail the more information we will obtain. Nevertheless, this is not exercised practically. Only functional requirements have got high focus and will be gathered in most situations."

Additionally, the other respondent said that "Considering ethical concerns helps to avoid requirements duplications, to give priority and attention to critical ethical issues according to organizational behavior, needs and wants. . . ."

"Considering ethics in requirements engineering helps us to respect social value, to empower customers, to protect confidentiality and privacy issues, to protect unauthorized access of information, to avoid unnecessary and unwanted requirements, to avoid deceptive requirements, and so on. However, as I know there are no laws/rules/regulations, which make developers accountable for their products. There is no legal ground to punish those developers who omit ethical concerns and issues. Something should be done to make developers liable for their faults. That thing may be a procedure, policy, or something else. There should be legal procedures, which enforce the inclusion of ethical concerns and the accountability of software developers."

4.2. Practices

We were inspired to conduct this research to investigate the ethical concerns of software requirements. This research has its own research objectives and research questions that should be answered. The main objective of the research was to investigate the ethical concerns of requirements engineering and have these propose a framework for requirements engineering ethics as the future investigation that will be follow this research. The first research question, "How are software engineers practicing the integration of ethical concerns of software during requirements engineering processes?" is answered by performing research objective one: "Investigate how ethical concerns are being put into practice during RE."

To investigate the practices of software engineers in the direction of integrating ethical concerns during requirements engineering, we assessed software-developing companies. In this regard, the practice was assessed with respect to industry practices, the usage of professional responsibility code of conduct standards, and guideline utilization practices. Regarding industry practices, except one, all the participating companies have a traditional way of carrying out requirements engineering processes. The single company is trying to focus on the early identification of ethical concerns, especially security aspects of ethics during the requirements engineering processes. Some of the manners in which this company is practicing include ensuring security issues by paying great attention to them. Additionally, the company is using issue-tracking software (such as requirements traceability matrices and document matrices) for filtering requirements. Further, this company is dividing departments in accordance with employees' specialization, such as business analysts, solution engineers, database designers, developers, and architects. In this regard, respondents replied, "this helps us to improve our performance and to produce quality software". Regarding the use of professional responsibility code of conduct standards, respondents of

six companies replied that they do not have any professional responsibility standards of systems development that they follow in their day-to-day work. One company claimed that it uses the IEEE code of ethics via the customization for requirements engineering. However, some employees of this company said, “We have not” and others stated only limited practices of IEEE. Similarly, almost all the companies, except company F, stated that they currently do not have any guidelines in place for considering ethical concerns throughout software development lifecycles.

In general, the findings refer to the absence of industry practices, professional responsibility code of conduct standards, and other guidelines within the companies for integrating the ethical concerns of software during requirements engineering processes.

4.3. Fundamental Ethical Concerns

The second research question, “What are indispensable ethical concerns of software, which should consider explicitly during requirements engineering processes?”, is addressed through research objective two: “Find out ethical concerns which should be clearly integrated with requirements engineering processes”. This objective is achieved through investigating fundamental ethical concerns (interview question five and focus group discussion question four), factors/hindrances of being ethical (interview question six), assessing the time taken up by hard challenges during their software development (interview question ten), and exploring the main ethical violations that occurred during software development (focus group discussion question two). With regard to this research question, we received two broad groups (themes) of problems. The first theme entails problems that are very critical, i.e., serious problems associated with ethical concerns of requirements engineering. We named this theme “fundamental ethical concern theme”, which covers the critical ethical concerns obtained from participants during interviewing and focus group discussion sessions.

Regarding these different opinions and reflections observed among the many discussions of participants, we present here only a few.

One participant said that ethical concerns could be categorized under nonfunctional requirements. He said, “Nonfunctional requirements are very vague to capture them, they are costly to test them. Users could not identify them correctly. Since there are these and other similar problems, taking non-functional requirements as a requirement are very challenging.” Furthermore, he replied that in our country context, even functional requirements are not stable. Having these unstable functional requirements taking non-functional requirements such as ethical concerns is a very difficult task. This is because it requires capacity in finance, platform, and other aspects.

The other respondent stated that most of the ethical concerns are personal. That means they are related to the way a person acts. As he said, “The challenge regarding requirements engineering is that in most cases, even though there are frameworks and guidelines, practically the proposed framework is not used. The other challenge is lack of knowledge both from the customers’ and business analysts’/requirements engineers’ side. The ability to identify requirements, to hear, to gather, and to appropriately document those requirements is a very challenging problem. Besides, from the customer side, they do not know how their business is doing.”

Another participant said, “The fundamental challenge during requirements engineering is the frustration of customers to use an automated system. Because they assumed that automation might cause them to lose their jobs, might cause overload (workload), bring them towards it. Due to this, when they are asked to give requirements, they could not be willing to give the real requirements (both functional and non-functional requirements). They hide every piece of asked information. In addition, unclear business process, ambiguous requirements, unfulfilled infrastructure, and usage of outdated tools and technologies are the other ethical concerns that I observed.”

In the same way, the other person said, “Customers might not be willing to give all necessary formats, documents, and reports. There is a tendency to hide them. Some of

them may be due to lack of knowledge, but most of them do it purposely for some reasons, such as some of them may be afraid to be jobless because of automation, some of them may lose unnecessary benefits associated with corruption, etc. . . . ”

In general, in the analysis, we found fundamental ethical concerns and grouped them into six categories: requirements of identification problems, quality-related problems, performing unpermitted activities, unwillingness to give requirements, knowledge gaps, and lack of legal grounds/rules for accountability. The corresponding codes of some of requirements identification problems include insufficient requirements engineering processes, incomplete requirements, lack of attention, focusing only on functional requirements, not identifying “to be” requirements, not identifying the real needs and wants of customers, unclear requirements, the presence of ambiguous requirements, omitting requirements, etc.

The other category under this theme is quality-related problems. In addition to requirements identification problems during data collection, we obtained problems, which were associated with quality. Some of those include security concerns, privacy concerns, cyber-attacks, confidentiality issues, usability issues, and others. Furthermore, performing unpermitted activities is another category. As most of the respondents stated, there are many unethical activities within software companies, such as lack of honesty in work, carelessness, adding evil code to source code, plagiarism, and so on.

The other category under this theme, fundamental ethical concerns, is an unwillingness to give appropriate requirements to engineers. From the customer’s perspective, many of the customers become reluctant to give necessary data of the intended system. As mentioned above, this is the main problem from the customer’s perspective. Some of the codes under this category are unwilling to give clear, real, appropriate, and precise requirements, being impure and too complex requirements.

The other one is the lack of legal grounds/rules for accountability. This category shows the absence of rules and regulations for software development in the country. As respondents of company E said, “We are rendering services for service giving organizations. However, in practice, this is too difficult and challenging. Because we develop services, and we give that package to service provider companies. Those companies may make the consent manually/by hand or orally and they may send authorized messages without an agreement to get wrong benefits from that person . . . ”. Additionally, as they said, “As we know, there are no laws/rules/regulations which make developers accountable for their product. There is no legal ground to punish those developers who violate ethical concerns and issues. Hence, there should be something that makes developers liable for their unethical activities. That thing may be a procedure, policy, or other. There should be legal procedures which enforce the inclusion of ethical concerns and the accountability of software developers.”

The last but not the least category is the knowledge gap. As discussed above, there is a lack of knowledge about both technical people and customers.

The second theme which is obtained here is management-related problems. Management-related problems are problems that are obtained during data analysis. This theme is inclined towards software project management issues. Different concerns are discussed in this. To mention only the two of the problems, frequent change requests and being reluctant to make changes, are the major ones. Lots of concerns are identified under these categories including scope creeping, conflicts due to differences of interests, denial of agreements, unwillingness to accept the artifact, unclear business problems, weak requirements change management, etc.

4.4. Methods

The third research question, “What are the methods/techniques that can be used to determine the upholding of ethical concerns along with requirements engineering processes?”, deals with methods/techniques. Research objective three, “explore methods which are used by software companies to identify ethical concerns for the purpose of ensuring the inclusion of all the necessary ethical parameters for requirements engineering”, helps to

answer this question. In the research findings, the presence/absence of ethical concern methods/techniques, identification methods, and checking mechanisms become necessary. Therefore, before determining and proposing relevant methods for ethical concerns, we investigated them via interview questions seven, eleven, and twelve. Additionally, we used focus group discussion questions two point three (2.3) and question three.

Under this theme, the methods, presence, or absence of ethical concern identification methods were investigated. Most companies do not have any specified methods to identify ethical concerns during requirements engineering processes. As most of the respondents said, they do not have any methods/checking mechanisms to do such things. Only a few of the respondents mention some methods that they use during requirements engineering. A few companies use some means of communications through discussions, workshops, and prototyping using different techniques, for instance, user acceptance testing and checklists and their own intuitive methods. Nevertheless, they mainly used these measures for functional requirements. In most cases, they are not used for nonfunctional requirements, specifically for ethical concerns. Generally, the research findings indicate that almost all companies do not have well-established identification methods and checking mechanisms whether ethical concerns are considered and incorporated during requirements engineering processes or not.

To conclude, the following table, Table 4, depicts the summary of the major findings/themes with their meaning.

Table 4. Summary of key findings.

Themes	Meaning of Themes	Sample Summarized Findings under Each Theme
Importance	This means the importance of clearly considering ethical concerns during software requirements engineering processes.	<ul style="list-style-type: none"> Helps to obtain quality software (helps to ensure privacy and security issues, enhances software project success, obtains refined artifacts, enhances ease to use, and so on.) Helps to minimize ethical concern violation (such as to elude deceptive requirements, protect confidentiality, respect social values, etc.)
Practice	It shows the practice of software companies regarding industry practices, usage of professional responsibility code of conduct standards, and guideline utilization practices.	<ul style="list-style-type: none"> Limited practice (on professional code of ethics only by one participant company). Do not have guidelines and/or standards.
Fundamental ethical concerns	Refers to basic obtained ethical concern problems.	<ul style="list-style-type: none"> Requirements identification problems; quality-related problems; performing unpermitted activities; unwillingness to give requirements; knowledge gap; lack of legal grounds/rules for accountability
Management related problems	Problems that are associated with software project management.	<ul style="list-style-type: none"> Frequent change request. Be reluctant with changes.
Methods	Refers to the methods/techniques used by sampled software companies to identify ethical concerns during requirements engineering processes.	<ul style="list-style-type: none"> Absence of methods. Little practice only by a few companies using traditional practices.

5. Discussions

In general, this research has its own objectives and research questions. The three research questions are set as follows:

1. How are software engineers practicing the integration of ethical concerns of software during requirements engineering processes?

2. What are the indispensable ethical concerns of software, which should be considered explicitly during requirements engineering processes?
3. What are the methods that can be used to determine the upholding of some serious ethical concerns in requirements engineering processes?

To answer the first research question, we investigate the practices of software industries with respect to industry practice, the consumption of professional responsibility code of conduct standards, and the utilization of any ethical guidelines in place to consider ethical concerns. As discussed above in the research findings in Section 4.2, the case companies in Ethiopia do not have a well-organized way of identifying and integrating ethical concerns during requirements engineering processes. Most of them did not use any guidelines as well as any professional code of conduct standards for the integration of ethical concerns during the early phase of SDLC. Instead of this, most of the respondents suggested the need for clear guidelines, standards, rules, and policies for the incorporation of ethical concerns during requirements engineering processes at a national level.

Moreover, the second research question is “What are indispensable ethical concerns of software, which should be considered explicitly during requirements engineering processes?”. Research objective two, “To find out ethical concerns which should be clearly integrated with requirements engineering processes”, is accomplished via interviews and focus group discussions. This objective was achieved through investigating fundamental ethical concerns, factors/hindrances to being ethical, assessing the time spent on hard challenges during software development, and exploring the main ethical violations that occurred during software development. Using these parameters, we found two broad groups of themes. These themes are fundamental ethical concerns and management-related problems. As discussed under Section 4.3, there are lots of identified issues in each theme.

Furthermore, the third research question, “What are the methods that can be used to determine the upholding of some serious ethical concerns in requirements engineering processes?” deals with methods. In the research, finding the presence/absence of ethical concerns, identification methods, and checking mechanisms become necessary. Therefore, before determining and proposing relevant methods for ethical concerns, we are investigating the available methods and techniques to identify ethical concerns in those case companies. The research finding showed that most companies do not have methods and other well-organized identification means for ethical concerns during requirements engineering processes. As discussed, they are using a traditional way of identification techniques that we used for functional requirements such as prototyping, discussions, and workshops. Only one company, company “F”, is trying to use some techniques to identify ethical concerns during requirements engineering processes such as traceability matrices and Jira tools. However, these techniques and tools do not use specific to nonfunctional requirements; they are used to check functional requirements too. This implies that in the case software companies, there is no specific method for measuring ethical concerns during requirements engineering processes.

With these research findings, our next task will be to propose an ethical framework with its appropriate components and methods for requirements engineering processes.

6. Conclusions and Future Work

Harmonizing ethical concerns is a difficult activity. In this regard, software engineers have the influential power to consider the ethical allegations of precise software design alternatives. The field of software engineering and ethical concerns should be integrated to develop useful software that exercises ethical awareness in software engineering practices and artifacts [40]. During software development, clearly considering ethical concerns is very useful for integrating both the technical and ethical issues of the software artifact. Software engineers are accountable for their profession and society at large [4,10]. Even though various approaches differ significantly in terms of their ability to deal with ethical issues, it is possible to apply ethics in the overall approaches of systems design [10]. Requirements engineering is the SDLC foundation phase [12,14]. At the same time, this

phase is exposed to many ethical violations. There are many cases in which requirements engineering becomes unethical [18–21].

So, in this research, we investigated the ethical issues of requirements engineering in the real settings of software industries. We took seven software industries from different sizes (small, medium, and large sizes) and maturity levels (with different years of experience), and data collection tools, interviewing, and focus group discussions were used in our study. To select companies and individual participants from each company, we also used purposive sampling techniques. Additionally, to analyze the collected data, we used qualitative data analysis techniques including holistic coding, initial coding, in vivo coding, and attribute coding strategies.

Through clear identification of the ethical concerns of requirements engineering and their consideration in the early phases of SDLC, requirements engineering, we can get quality software artifacts and services and minimizes the occurrences of ethical violations. In this research, the importance of considering ethics, practices of software industries, methods used to identify ethical issues, and fundamental ethical problems of requirements engineering were investigated. The results of the research showed that considering ethics in the early phases of SDLC is essential for obtaining high-quality software products and/or services and for minimizing the occurrences of ethical violations. However, the research findings showed the absence of industry practices, professional responsibility code of conduct standards, and other guidelines within the companies for integrating the ethical concerns of software during requirements engineering processes. The findings also indicate that almost all companies do not have any identification methods and checking mechanisms, whether ethical concerns are considered and incorporated or not. The major identified issues include requirements identification problems (such as insufficient requirements engineering process, incomplete requirements, focusing only on functional requirements, etc.); quality-related problems (such as security concerns, privacy concerns, cyber-attacks, confidentiality issues, and so on); performing unpermitted activities (for instance lack of honesty on at work, carelessness, adding evil codes to source codes, plagiarism, and so forth); unwillingness to give requirements (such as unwillingness to give clear, real, appropriate and precise requirements); knowledge gaps (such as frequent change requests and reluctance with changes); and a lack of legal ground/rules for accountability.

In general, the findings showed the absence of industry practices, professional responsibility code of conduct standards, and other guidelines within the companies for integrating ethical concerns of software during requirements engineering processes. They also indicate that almost all companies do not have any identification methods and checking mechanisms whether ethical concerns are considered and incorporated or not. Moreover, in the analysis, we found fundamental ethical concerns which are grouped into six categories as requirements identification problems, quality-related problems, performing unpermitted activities, unwillingness to give requirements, knowledge gaps, and the lack of legal grounds/rules for accountability.

Propose an ethical framework for requirements engineering which helps as a clear map concerning the integration of ethical concerns in the requirements engineering phases of SDLC. Our future proposed work will help software engineers to contribute to their profession in a usable way, both for the stakeholder of the software artifact and for the public at large. Additionally, it will help them to safeguard from undertaking unethical activities and practices, which, consequently, will bring many challenges for their career and personality.

Author Contributions: Conceptualization, S.E.B., N.M.G. and D.M.; Methodology, S.E.B., N.M.G., D.M. and N.P.; Validation, S.E.B., N.M.G. and D.M.; Formal analysis, S.E.B.; Investigation, S.E.B., N.M.G. and D.M.; Project administration, D.M.; Supervision, N.M.G. and D.M.; Writing—original draft, S.E.B.; Writing—review & editing, S.E.B., N.M.G., D.M. and N.P.; Funding acquisition, N.M.G. All authors have read and agreed to the published version of the manuscript.

D. Interview Questions

1. Can you please tell me, from your perspective, the importance of clearly considering ethical concerns during software requirements engineering processes?
2. In your company, in what manner is ethics being practiced carrying out requirements engineering processes?
3. Which professional responsibilities related to ethical concerns for software requirements engineering processes of SDLC do you practice in your day-to-day activities?
4. Do you currently follow any guidelines in place for considering ethical concerns throughout software development lifecycles?
5. During requirements engineering processes which fundamental ethical concerns are you considering?

E. Focus Group Discussion Questions

1. What do you think about the concept of ethical concerns in requirements engineering processes?
2. In your company, in what manner is ethics being practiced carrying out requirements engineering processes?
 - 2.1 In your view, how have the main ethical violations occurred during software requirements engineering processes?
 - 2.2 What are those violations?
3. What methods have you used to identify ethical concerns for requirements engineering processes?
 - 3.1 Why did you choose these methods?
 - 3.2 How did you make the decision to use these methods?
 - 3.3 What did you like about these methods?
 - 3.4 What did you not like about these methods?

References

1. Vallor, S.; Narayanan, A.; Regnell, B.; Jones, C.; Skipper, R. An Introduction to Software Engineering Ethics. In *Applied Ethics*; Santa Clara University: Santa Clara, CA, USA, 2015; pp. 1–60.
2. Schon, E.-M. A Framework for Modeling and Improving Agile Requirements Engineering. Ph.D. Thesis, Computer Languages and Systems Department, University of Seville, Seville, Spain, 2017.
3. Bowen, W.R. *Engineering Ethics. Challenges and Opportunities*; Springer International Publishing: Cham, Switzerland, 2014.
4. Lurie, Y.; Mark, S. Professional ethics of software engineers: An ethical framework. *Sci. Eng. Ethics* **2016**, *22*, 417–434. [[CrossRef](#)] [[PubMed](#)]
5. Bloch, M.; Blumberg, S.; Laartz, J. Delivering large-scale IT projects on time, on budget, and on value. *Harv. Bus. Rev.* **2012**, *5*, 2–7.
6. Wojewoda, S.; Hastie, S. Standish Group 2015 Chaos Report. 2015. Available online: <https://www.infoq.com/articles/standish-chaos-2015> (accessed on 11 February 2022).
7. Geneca. Why Up to 75% of Software Projects Will Fail. 2017. Available online: <https://www.geneca.com/why-up-to-75-of-software-projects-will-fail> (accessed on 11 February 2022).
8. International Project Leadership Academy. Advanced Leadership Skills—Developing Project Management Capabilities. Available online: http://callead.com/WTPF/?page_id=1445 (accessed on 11 February 2022).

9. AlAmmar, F. A Framework for Ethical Practices in Software Development Life Cycle: A Case Study in the Kingdom of Saudi Arabia. Master Thesis, University of Prince Sultan, Riyadh, Saudi Arabia, 2016.
10. Aydemir, F.B.; Dalpiaz, F. A Roadmap for Ethics-Aware Software Engineering. In Proceedings of the 2018 IEEE/ACM 40th International Conference on Software Engineering: Companion (ICSE-Companion), Gothenburg, Sweden, 29 May 2018; IEEE: Washington, DC, USA, 2018; pp. 228–229.
11. Gotterbarn, D. Software engineering ethics. *Encycl. Softw. Eng.* **2001**, *2*, 1–13.
12. Shah, T.; Patel, S.V. A review of requirement engineering issues and challenges in various software development methods. *Int. J. Comput. Appl.* **2014**, *99*, 36–45. [\[CrossRef\]](#)
13. Aberkane, A. Exploring Ethics in Requirements Engineering. Master's Thesis, Utrecht University, Utrecht, Netherland, 2018.
14. Pavan, M. Value Based Requirements Engineering: State-of-art and Survey. Master's Thesis, Faculty of Computing, Blekinge Institute of Technology, Karlskrona, Sweden, 2016.
15. Leveson Nancy, G. The role of software in recent aerospace accidents. In Proceedings of the 19th International System Safety Conference, System Safety Society, Unionville, VA, USA, 10–14 September 2001.
16. Wong, W.E.; Debroy, V.; Surampudi, A.; Kim, H.; Siok, M.F. Recent Catastrophic Accidents: Investigating How Software Was Responsible. In Proceedings of the Fourth International Conference on Secure Software Integration and Reliability Improvement, Singapore, 9–11 June 2010; pp. 14–22. [\[CrossRef\]](#)
17. Linda, G. Lost radio contact leaves pilots on their own. *J. IEEE Spectr.* **2004**, *41*, 16–17.
18. Cari, B. Overexposure of radiation therapy patients in Panama: Problem recognition and follow-up measures. *Rev. Panam. Salud Pública* **2006**, *20*, 173–187.
19. Zhang, B.; Veijalainen, J.; Kotkov, D. Volkswagen emission crisis: Managing stakeholder relations on the Web. In Proceedings of the 12th International Conference on Web Information Systems and Technologies (WEBIST 2016), Rome, Italy, 23–25 April 2016; Volume 1, pp. 176–187.
20. Nazanin, M. A case study of volkswagen unethical practice in diesel emission test. *Int. J. Sci. Eng. Appl.* **2016**, *5*, 211–216.
21. Boston, W.; Varnholt, H.; Sloat, S. Volkswagen Blames 'Chain of Mistakes' for Emissions Scandal. *Wall Str. J.* **2015**. Available online: <https://www.wsj.com/articles/vw-shares-up-ahead-of-emissions-findings-1449740759> (accessed on 15 November 2021).
22. Lee, B.D.; Paredis, C.J. A conceptual framework for value-driven design and systems engineering. *Procedia CIRP* **2014**, *21*, 10–17. [\[CrossRef\]](#)
23. Zakaria, N.A.; Ibrahim, S.; Mahrin, M.N.R. Examining Value-Based Factors in Software Development: A Survey Study in Malaysian Public Sector. In Proceedings of the 2015 9th Malaysian Software Engineering Conference (MySEC), Kuala Lumpur, Malaysia, 16–17 December 2015; Volume 1, pp. 13–17.
24. Whittle, J.; Ferrario, M.A.; Simm, W.; Hussain, W. A case for human values in software engineering. *IEEE Softw.* **2019**, *38*, 106–113. [\[CrossRef\]](#)
25. Evans, W.G. Ethics, values and practice. *South Africa. Dent. J.* **2019**, *74*, 333–334. [\[CrossRef\]](#)
26. Lee, G. Is there a "software engineering ethics"? Comparing commercial and research software engineering. In Proceedings of the 2021 IEEE/ACM 2nd International Workshop on Ethics in Software Engineering Research and Practice (SEthics), Madrid, Spain, 4 June 2021; pp. 13–17.
27. Phillip, L. *Requirements Engineering for Software and Systems*, 2nd ed.; CRC Press, Taylor & Francis Group: Boca Raton, FL, USA; London, UK, 2014.
28. Darwish, N.R.; Megahed, S. Requirements engineering in scrum framework. *J. Requir. Eng.* **2016**, *149*, 24–29.
29. Roger, S.P.; Bruce, R.M. *Software Engineering: A Practitioner's Approach*; McGraw-Hill Education: New York, NY, USA, 2015.
30. Ian, S. *Software Engineering*, 9th ed.; Addison-Wesley: New York, NY, USA; San Francisco, CA, USA, 2011.
31. Babar, M.I.; Ghazali, M.; Jawawi, D.N.; Zaheer, K.B. StakeMeter: Value-Based stakeholder identification and quantification framework for value based software systems. *PLoS ONE* **2015**, *10*, e0121344. [\[CrossRef\]](#) [\[PubMed\]](#)
32. Lynch, W.T.; Kline, R. Engineering practice and engineering ethics. In *Engineering Ethics*; Routledge: London, UK, 2017; pp. 601–631.
33. Quigley, M. *Encyclopedia of Information Ethics and Security*; IGI Global: Hershey, PA, USA, 2007.
34. Génova, G.; González, M.R.; Fraga, A. Ethical Responsibility of the Software Engineer. In Proceedings of the CAISE*06 Workshop on Philosophical Foundations on Information Systems Engineering PhiSE '06, Luxemburg, 5–9 June 2006.
35. Gotterbarn, D.; Miller, K.; Rogerson, S. Software engineering code of ethics is approved. *Commun. ACM* **2003**, *42*, 102–107. [\[CrossRef\]](#)
36. IEEE-CS/ACM Joint Task Force. Software Engineering Code of Ethics and Professional Practice. *Sci. Eng. Ethics* **2001**, *7*, 231–238. [\[CrossRef\]](#)
37. Berenbach, B.; Broy, M. Professional and ethical dilemmas in software. *Computer* **2009**, *42*, 74–80. [\[CrossRef\]](#)
38. Gotterbarn, D.; Miller, K.W. The public is the priority: Making decisions using the software engineering code of ethics. *Computer* **2009**, *42*, 66–73. Available online: <https://sci-hub.se/10.1109/MC.2009.204> (accessed on 9 December 2021). [\[CrossRef\]](#)
39. Volkman, R. Computer ethics beyond mere compliance. *J. Inf. Commun. Ethics Soc.* **2015**, *13*, 176–189. [\[CrossRef\]](#)
40. Karim, N.S.A.; Al Ammar, F.; Aziz, R. Ethical Software: Integrating Code of Ethics into Software Development Life Cycle. In Proceedings of the Computer and Applications (ICCA), International Conference, Doha, United Arab Emirates, 6–7 September 2017; pp. 290–298.

41. Van de Poel, I. An ethical framework for evaluating experimental technology. *Sci. Eng. Ethics* **2016**, *22*, 667–686. [[CrossRef](#)]
42. Gotterbarn, D.W.; Brinkman, B.; Flick, C.; Kirkpatrick, M.S.; Miller, K.; Vazansky, K.; Wolf, M.J. *ACM Code of Ethics and Professional Conduct*; Association for Computing Machinery: New York, NY, USA, 2018.
43. Gogoll, J.; Zuber, N.; Kacianka, S.; Greger, T.; Pretschner, A.; Nida-Rümelin, J. Ethics in the software development process: From codes of conduct to ethical deliberation. *Philos. Technol.* **2021**, *34*, 1085–1108. [[CrossRef](#)]
44. Jones, C. *Software Requirements and the Ethics of Software Engineering*; Capers Jones & Associates LLC.: Pittsburgh, PA, USA, 2012.
45. Thew, S.; Sutcliffe, A. Value-Based requirements engineering: Method and experience. *Requir. Eng.* **2018**, *23*, 443–464. [[CrossRef](#)] [[PubMed](#)]
46. Rashid, A.; Weckert, J.; Lucas, R. Software engineering ethics in a digital world. *Computer* **2009**, *42*, 34–41. [[CrossRef](#)]
47. Cemiloglu, D.; Arden-Close, E.; Hodge, S.; Kostoulas, T.; Ali, R.; Catania, M. Towards Ethical Requirements for Addictive Technology: The Case of Online Gambling. In Proceedings of the 2020 1st Workshop on Ethics in Requirements Engineering Research and Practice (REthics), Zurich, Switzerland, 31 August 2020; pp. 1–10.
48. Ali, R.; Jiang, N.; Phalp, K.; Muir, S.; McAlaney, J. The emerging requirement for digital addiction labels. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany, 23–26 March 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 198–213.
49. Alrobai, A.; Phalp, K.; Ali, R. Digital addiction: A requirements engineering perspective. In Proceedings of the International Working Conference on Requirements Engineering: Foundation for Software Quality, Essen, Germany, 7–10 April 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 112–118.
50. Paech, B.; Schneider, K. How do users talk about software? Searching for common ground. In Proceedings of the 2020 1st Workshop on Ethics in Requirements Engineering Research and Practice (REthics), Zurich, Switzerland, 31 August 2020; pp. 11–14.
51. Ali, N.; Hong, J.E. Requirements-Driven Evolution in Software Product Lines. In Proceedings of the KSII The 10th International Conference on Internet, (ICONI), Phnom Penh, Cambodia, 16–19 December 2018; pp. 161–165.
52. Ali, N.; Hong, J.E. Using social network service to determine the initial user requirements for small software businesses. *arXiv* **2019**, arXiv:1904.12583. preprint.
53. Ali, N.; Hong, J.E. Value-Oriented requirements: Eliciting domain requirements from social network services to evolve software product lines. *Appl. Sci.* **2019**, *9*, 3944. [[CrossRef](#)]
54. Ali, N.; Hong, J.E.; Chung, L. Social network sites and requirements engineering: A systematic literature review. *J. Softw. Evol. Process* **2021**, *33*, e2332. [[CrossRef](#)]