Addis Ababa University

School of Graduate Studies

IT Doctoral Program

Framework for Software Requirements Engineering Ethics

Seblewongel Esseynew

A Dissertation Submitted to

The Track of Software Engineering

Presented in Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Technology (Software Engineering)

Addis Ababa, Ethiopia

May 2025

Addis Ababa University

School of Graduate Studies

IT Doctoral Program

This is to certify that the dissertation prepared by Seblewongel Esseynew, entitled: *Framework for Software Requirements Engineering Ethics*, and submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in Software Engineering complies with the regulations of the university and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

Examiner: Professor Ana Moreira Signature _____Date _____

Examiner: Dr. Mesfin Kifle Signature _____ Date _____

Principal Advisor: Professor Nuno M. Garcia Signature _____ Date _____

Co-Advisor: Dr. Dida Midekso Signature _____Date _____

_____

Track Coordinator or IT Doctoral Program Director

ABSTRACT

Framework for Software Requirements Engineering Ethics

Seblewongel Esseynew

Software Engineering

Addis Ababa University, 2025

Ethics has an extensive array of uses throughout many fields of study and in diverse circumstances. Ethics significantly impacts decisions made in requirements engineering. Requirements engineering is a fundamental process in software development projects. At the same time, it is a difficult phase and exposes to many ethical problems. In software engineering, many well-known codes of ethics address a wide range of issues that are significant for software engineers with different perspectives. However, it is found to be challenging to follow and implement them in the day-to-day work of software engineers.

This research aimed to study ethical issues raised in requirements engineering processes. In this research, we investigated requirements engineering ethical concerns. Data were collected from seven case software companies using interviews and focus group discussions.

Themantic qualitative analysis is employed to analyse the collected data. The analysis result revealed that companies lack industry practices, use of professional code of conduct standards, methods, or any other ethical guidelines to be followed during requirements engineering. Based on the analysis, one hundred seventy six codes, thirty four categories and five themes emerged. The five main themes include importance, practice, fundamental ethical concerns, management-related problems, and methods.

In this research, we proposed a framework with five basic components for software requirements engineering ethics to address critical ethical concerns which were

screened and identified during the analysis per their imprance to achieve the research questions. These components are Ethics for Knowledge Gap, Ethics for Requirements Identification, Ethics for Requirements Quality, Ethics for Unwillingness, and Ethics for doing Unpermitted Activities. Each of them gives specific pieces of advice that can be solutions to address the identified requirements engineering ethical concerns.

The validity and reliability of the proposed framework were evaluated using an expert evaluation involving five-point Likert scale questionnaires and focus group discussions. Additionally, open-ended questionnaires are used. The experts are purposively selected from the software industries.

The evaluation results indicate that the suggested framework is beneficial to use it as a daily guideline. It can be used as an advising guide for software engineers to minimize the occurrence of those identified ethical concerns during requirements engineering.

Keywords: Concerns**,** Categories, Components, Ethics, Evaluation, Framework, Requirements Engineering, Validity

ACKNOWLEDGMENTS

Above all, I express my heartfelt gratitude to the Almighty God, the ultimate source of wisdom and strength, for being my guiding light throughout the challenging journey of pursuing a Ph.D. study.

I want to express my sincere gratitude to my esteemed supervisors, Professor Nuno M. Garcia and Doctor Dida Midekso, for their invaluable guidance and support throughout my Ph.D. journey. Their mentorship extended beyond conventional advising, encompassing coaching, encouragement, thorough editing, and insightful reviews, all of which played a crucial role in shaping the research work.

I would like to convey my gratitude to Professor Nuno M. Garcia for inviting me for a short-term research visit abroad and providing funds for the publication of papers. I am also grateful to Doctor Dida Midekso for his exceptional leadership, offering constructive feedback, and skillfully enhancing various aspects of the research through thorough editing and in-depth reviews of the dissertation and the published papers.

I am thankful to Addis Ababa University, the Post Graduate Program, the IT Doctoral Program, the Software Engineering Track, and the Computer Science Department for their support during my Ph.D. study. I would like to thank Dr. Ayalew Belay, the dissertation committee member, for his review of the dissertation and encouragement.

I sincerely thank Debre Birhan University for the helpful opportunity to pursue my Ph.D. Special thanks go to Mr. Tsegaye Siyum, Mr. Dawit Seifu, Mr. Dawit Abate, and Mr. Behailu Maire for their support during the research data collection. I extend my gratitude to Mr. Andinet Nigusie for his precious support and encouragement. Sincere thanks to Dr. Seid Mohammed Ali and Dr. Getahun Melaku for their support and feedback in editing the published research paper.

My sincere gratitude extends to the software companies for their plentiful collaboration during the data collection, which was vital to the study's success. Their willingness to share valuable insights enhanced the depth and quality of my research,

providing real-world context to my findings. Special thanks to Mr. Semahegn Abebe from CNET software company, Mr. Baheru Zeyenu from Africom, Mr. Elyas Abebe, Mr. Robel Tesfaye, and Mr. Dilinesachew Girma from Ethiopia Information Network Security Agency for their warm welcome, valuable insights, and collaborative efforts.

I would like to express my sincere thanks to the participants for giving their valuable time and insights during the evaluation of the proposed framework. Special thanks go to the team members of the Ambient Assisted Living Computing Telecommunication Laboratory (ALLab) at the University of Beira Interior, Covilha, Portugal, for their considerable help during the evaluations of the proposed framework.

My deepest thanks go to FCT/MEC and COST Action for their financial support in funding the publication costs of our published papers.

I am deeply grateful to my lovely children for their support throughout my challenging Ph.D. journey. Their patience, love, and understanding were vital for making this rewarding experience possible. I am profoundly thankful to my brother, Gizachew Esseynew, for being my guiding light from the beginning of my educational voyage. His sacrifice, significant support, and encouragement were crucial in shaping who I am today.

I appreciate the support and inspiration from my dear friends, Mrs. Mintamir Lykun and Mrs. Dibora Abate. I thank them for standing by me and being uplifting presences that made this journey more meaningful.

My heartfelt appreciation goes out to the many individuals who, directly or indirectly, supported me throughout my Ph.D. study. Thank you to everyone involved in this journey.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ACM: Association for Computing Machinery

AI: Artificial intelligence

BABOK: Business Analysis Body of Knowledge

BPR: Business Processing Reengineering

CAS: Collision Avoidance Systems

Cl_n: Lists of Clauses (Cl) number n

Cn: Component number n

EPA: Ethics for Prohibited Actions

EKG: Ethics for Knowledge Gap

ERI: Ethics for Requirements Identification

ERQ: Ethics for Requirements Quality

EU/D: Ethics for Unwillingness or Disinclination

FGDs: Focus Group Discussions

FAA: Federal Aviation Administration

GDP: Gross Domestic Product

IBM: International Business Machines

IEEE: Institute for Electrical and Electronics Engineers

IEEE-CS: Institute of Electrical and Electronics Engineers Computer Society

IT: Information Technology

IEEE-CS/ACM JTFSECEPP: Institute of Electrical and Electronics Engineers Computer Society and Association for Computing Machinery Joint Task Force on Software Engineering Code of Ethics and Professional Practice

KGP: Knowledge Gap Problems

NASA: National Aeronautics and Space Administration

PAHO: Pan American Health Organization

RE: Requirements Engineering

RIP: Requirements Identification Problems

RQP: Requirements Quality-related Problems

PA: Doing Prohibited Activities/Actions

SWECOE: Software Engineering Code of Ethics

SDLC: System Development Life Cycles

TDI: Turbo Diesel Injection

U/D: Unwillingness or Disclination

VSCS: Voice Switching and Control System

Chapter One – Introduction

1.1 Background

In this rapidly growing and competitive world, software has become the prevailing deriver for growth, innovation, and prosperity [1, 2, 3, 4]. As Atkinson [1] explained, software is essential in creating high-paying jobs, covers a significant share of the Gross Domestic Product (GDP), and compels productivity and GDP growth. It also derives innovation, provides different opportunities, and is a key tool to take over competitive advantages. On the other hand, software has turned out to be a decisive economic resource [5]. Every activity of organizations, businesses, and individuals directly and/ or indirectly depends on software. Moreover, most of our day-to-day activities are dependent on software applications. For instance, in our occupation, business transactions, educational systems, healthcare systems, entertainment, and others can be negatively or positively affected by software systems [6].

In earlier times software had received minimal focus [7]. Organizations used to give less attention to software as it was not an integral part of organizational systems. Now, however, it is not only an integral part but also a critical part of many systems. As technologies have become more advanced, developing software systems is becoming a more challenging task since it involves technical and non-technical aspects, including ethical issues [3, 6, 8].

A software engineer is someone who takes accountability for both the ethical concerns surrounding software objects and their profession [9]. The software engineer must have enough knowledge, experience, and skills in his or her profession since he or she holds the key, software package [10]. On the other hand, customers are reliant on software artifacts, and they cannot recognize details of abstractions behind software or its artifacts. Because of the interaction between these two technical and non-technical aspects, it is important to maintain ethical considerations and balance them to regulate the software system and ensure that it is free from harm. Developing software systems unethically leads to many negative effects. Some of these include personal harm, financial crises, loss of life because of unusable and unfeasible

software artifacts, and loss of faith in companies that possess them [11, 12, 13]. Eventually, this could result in severe lawful consequences that influence people's lives, organizations, and the overall society [6].

The success of software projects is significantly impacted by ethical concerns that will occur in the processes of requirements engineering (RE) [14]. In software development projects, RE is the fundamental phase of systems development through which requirements of a software project are identified and decided. It involves identifying the needs of stakeholders, understanding contexts, depicting requirements clearly and logically, and requirements negotiation, validation, documentation, and management. It is the hardest phase [15, 16, 17] meanwhile, it determines rigorously what to build and determines detailed requirements specifications and validations, including functional and non-functional requirements details [18]. It determines whether a software project succeeds or fails. In RE, ethical concerns might result in serious issues that eventually lead to project failure.

Starting from early times to these days, software project failure has become a well-known problem [19]. Software project failures have been widely studied by different researchers [20, 21, 22, 23]. These researches disclosed the existence of chief problems that lead to the failure of software development projects [24]. A few of the reports regarding this are presented below.

Research done by McKinsey and Company together with Oxford University upon 5,400 Information Technology (IT) projects indicated that above 44.8% of software projects have delivered less output than expected [25]. Mainly, this research demonstrated that developing IT projects pays little attention to stakeholders' needs and expectations.

A survey conducted on 600 individuals involved in software project development during 2010 - 2011 [26] revealed that, even at the initial stages of their software projects, many businesses anticipated potential failures. According to this study, the major causes of software project failures include unclear enterprise objectives and requirements, disagreements with stakeholders, and excessive rework. As stated in the

study, 75% of project participants presented shortages in terms of self-assurance of the success of their projects.

The other study, which was investigated by International Business Machines (IBM) using about 1,500 managers concerning project failure rate [27], found that the largest obstacle to project success was found to be the people factor, which includes changing attitudes and mindsets, which account for 41.73%, corporate culture (35.25%), and a shortage of senior management support (23.02 %). The study's conclusions draw attention to the crucial relationship that exists between corporate culture, human factors, and senior management's support when it comes to ethical issues in RE, particularly when those issues arise in software projects. It suggests that "ignoring the needs of stakeholders and changing views might give rise to ethical issues, which may then result in software that falls short of ethical standards and/or user expectations". It also demonstrates how corporate cultures, that prioritize schedule or speed over detail, may ignore ethical considerations in a rush to complete a project. In addition, it underlines how these issues are made worse by senior management's lack of support, which makes it more difficult to make ethical decisions and allocate resources that are required for meaningful stakeholders' involvement.

Gulla [28] investigated that failure to address issues, including stakeholder conflicts, poor user input, and insufficient communication, encompassing occasional open communication and deprived communication with sponsors and business users, are some of the major identified reasons for software project failure. This investigation showed "certain principal causes that subsidize software project failure, and these factors can often be worsened by ethical problems in RE, such as conflict of interest, disregard for user input, and insufficient communication".

The Standish Group studied 50,000 projects (from 2011 to 2015) all over the world about software project success and failure. The research outcome showed that further work should be done to obtain flourishing results from software projects. Table 1.1 shows the results obtained from the investigation [29].

Table 1.1 Software Project Failure Rate [29]

	2011	2012	2013	2014	2015
Successful	29%	27%	31%	28%	29%
Challenged	49%	56%	50%	55%	52%
Failed	22%	17%	19%	17%	19%

In addition to the above data depicted in Table 1.1, the 2015 Standish Group Chaos Report indicates that 44% of software projects failed due to insufficient requirements [29]. In the same way, the Standish Group's Chaos Report 2021 shows that 83.9% of IT projects fail due to many reasons. As stated in the report, among them, the first top factor that leads to software project failure is incomplete requirements [30]. Besides, according to their report, the first top indicators of challenged IT projects are inadequate specification, deficiency in customer feedback, and insufficient requirements. These problems not only hinder software project success but also raise ethical concerns. Incomplete requirements can lead to misunderstandings and the delivery of products that fail to meet stakeholder needs, significantly harming users and wasting resources. Similarly, lacking user input undermines the ethical necessity to prioritize user well-being and satisfaction. Furthermore, ambiguous requirements and specifications are obstacles to transparency and accountability and threaten project integrity. In the same way, a recent study showed that the leading problems of software projects come from RE problems that have deep ethical implications for the success of software projects [31].

From early times to these days, RE has been discovered as one of the most significant phases for developing flourishing software products [15]. The ethical issues of RE contribute to software project failure. So, they should be carefully analyzed to get an understanding of their impacts, leading to successful software development practices [32]. Software artifacts developed by software engineers affect the existence and livelihoods of clients, other stakeholders, and society as well. Thus, they should give

attention to the requirements engineering processes [10, 32]. For this purpose, software engineers' practical skills and ethical responsibility should be synergized to have usable and valuable software products and services [10].

While different scholars have varied perspectives on ethics, many of the definitions match with the Institute of Electrical and Electronics Engineers Computer Society and Association for Computing Machinery Joint Task Force on Software Engineering Code of Ethics and Professional Practice (IEEE-CS/ACM JTFSECEPP). To see some of them, Immanuel Kant prioritizes obligation and human value ethics, focusing on duty and inherent human value [33].

Stuart [34] linked ethics to maximizing happiness and minimizing pain. Aristotle [35] viewed ethics as a balance between excess and deficiency, but Stuart associated ethics with maximizing enjoyment and minimizing misery. Donald Gotterban [36] underscored how crucial it is for software engineers to behave in the public interest and take societal effects into account. According to Michael Davis [37], ethics is the study of moral principles and how they relate to making decisions. Baase [38] highlighted issues like security and privacy in computing, and she defined ethics as the rules that govern human behaviour.

IEEE-CS/ACM Joint Task Force defines ethics as upholding principles prioritizing public and client interests, professional integrity, and lifelong learning first. Collectively, these place a strong emphasis on accountability, transparency, and the larger societal effects of decision-making [9, 39]. According to IEEE-CS/ACM JTFSECEPP, principles refer to the basic ethical guidelines that professionals should follow to ensure integrity, responsibility, and professionalism in software engineering. The IEEE-CS/ACM JTFSECEPP ethical principles are formulated based on guiding values such as autonomy, beneficence, justice, integrity, etc. They serve as the foundation for ethical decision-making and the professional conduct of software engineers [9].

For the purpose of this research, ethics is described as adhering to the proposed RE ethical framework for ethical decision-making during RE processes, guaranteeing client needs, confidentiality, and public safety, all the while upholding stakeholder trust and ongoing professional development.

1.2 Motivation

In this digital society, the pervasive influence of software in various aspects of our lives is unquestionable. Software has become the basis of our existence, from critical systems that manage financial transactions to applications that facilitate social interactions. As the dependence on software continues to grow, the ethical implications associated with its development, starting from the initial stage at RE, become increasingly significant [40].

There have been many cases in which RE has become unethical. Here, three examples are explained in which requirements of software become unethical by holding up deceptive requirements, toxic requirements, and improper requirements [41, 42, 43, 44, 13, 45, 46].

a. The 2009–2015 Volkswagen Turbo Diesel Injection (TDI) vehicles are one example. The TDI ethical problems of these model cars primarily come from the company's use of software that manipulated emissions test results for their diesel vehicles. [41, 42, 43, 44, 13].

   Volkswagen had installed deceptive software in their diesel cars to cheat on emissions tests. The software allowed the vehicles to emit pollution within acceptable limits during the testing laboratory, but 35 times exceeded in real-world driving conditions. This deception violated Environmental Protection Agency standards, impacting direct and indirect stakeholders. This deliberate deception led to environmental concerns, regulatory inquiries, and legal actions and damaged the company's reputation. Volkswagen faced criticism for prioritizing profit over environmental and public health considerations. When we look at this case for RE, the primary unethical problem was that software engineers deliberately used deceptive requirements to design those model cars to detect and cheat emission levels differently.

b. The National Oncology Institute in Panama experienced unethical practices related to improper requirements, leading to miscalculated radiation doses [46]. Among 478 cancer patients treated, 3 likely died from radiation overdose. The health minister sought an investigation by international experts, revealing that 56 patients received incomplete teletherapy fields, with treatment periods calculated using a faulty computerized system. Dosage errors of +10% to +105% affected 28 patients, resulting in 23 deaths, at least 18 due to direct radiation impact. This case exemplifies the consequences of improper requirements. In this case, the unethical perspective of RE lays in the inaccurate definition and validation of software requirements for radiation calculation. The software was supposed to accurately calculate radiation dosage for cancer treatment, but due to errors in requirements specification and validation, incorrect dosages were administered to patients, leading to serious harm, including life death.

c. The breakdown of communication between the Federal Aviation Administration's (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 [45]. An air traffic supervisor faced a crisis when voice communication suddenly failed among 400 airplanes heading to the southwestern United States. The FAA's radio system, Voice Switching and Control System (VSCS), in Palmdale, California, malfunctioned without warning, impacting 450 flights and causing over 30,000 traveller' disturbances. The fault originated from a flawed timer design in the VSCS, leading to a three-hour radio breakdown and communication loss. This case illustrates the dangers of toxic requirements. Toxic requirements refer to requirements that have long-term, unidentified hazardous effects on direct and indirect stakeholders of a software project [32, 47]. This case demonstrates that safety-critical requirements are not properly incorporated. The failure to ensure clear and effective communication of safety-critical requirements can be taken as an unethical issue observed from a requirements engineering standpoint.

As seen from each of the above case examples, unethical doing during requirements engineering comes from specific actions or failures related to how requirements are defined, validated, communicated, and/or manipulated.

The motivation for developing a framework for RE ethics stems from recognizing the ethical challenges that arise in RE, where decisions reached in the preliminary phase of development can have extensive consequences [48, 49]. Thus, investigating ethical issues specific to RE is essential to develop a guideline that software engineers can follow in their day-to-day RE work.

Therefore, we are motivated to study the applicability of explicit consideration of ethical concerns during RE by proposing a framework for RE ethics.

1.3 Statement of the Problem

The importance of ethical software development has greatly increased to handle the ongoing growth of businesses' necessities, competition, users' needs, and the public's safety. It is also a key in enhancing the success of industries [50]. Presently, software industries are obliged to enter the rapidly changing global competitive marketplace [51]. Hence, software industries must make the best use of software through the appropriate and explicit integration of ethical principles when they develop software artifacts, products, and services.

Throughout software artifacts development, it is significant for software industries to focus on ethical considerations to maintain sustainable competitiveness in a period of ever-increasing uncertainty and market globalization [52].

Without software, computer systems become collections of electronic and mechanical components that cannot perform useful functions. This implies that software determines the use of its artifacts. Software related issues are frequently the cause of a great deal of technological failures [10]. If a software system fails, it can bring numerous catastrophes to living things [53]. In most cases, such failures are assumed to be problems associated with project management. However, there are many situations in which a software project fails due to ethical issues that will occur during RE [10]. Typically, these malfunctioning outcomes stem, either explicitly or implicitly, from software engineers' tendency to disregard their professional duties by engaging in unethical activity [6, 53].

Software project fails due to different reasons [19]. Many studies have been done concerning this. However, this issue still becomes a serious problem in the ecosystem of software engineers [19, 54]. Now, many software initiatives are unlikely to fail financially. In contrast, it creates profit in the short term but can lead to significant harm to society over the medium and long term due to noncompliance with policies, rules, and software engineering codes of ethics [10]. As studies show, in most cases software engineers did not follow professional codes of conduct standards in their daily software development work to consider ethical concerns [6, 10, 55].

Throughout software development, ethical concerns should be clearly and appropriately considered to have an expressive and justified result. These concerns will be potentially identified during RE [10, 52, 56]. Although the implementation of a software system is performed accurately, the software product will fail if a worthless requirement is incorporated [15].

RE has been exposed to many problems [16]. Some of them include holding up toxic requirements, improper requirements, deception requirements, inadequate engineering of requirements, ignoring issues that need critical focus, insufficient specification, unnecessary complexity and functionality, vague requirements, the presence of a conflict of interests, etc. [6, 57, 58, 41, 42, 43, 44, 13]. These, in turn, have impacts on the other phases, and the final products and services. These problems many security vulnerabilities, misrepresentations of stakeholder needs, delays, usability problems, partiality, dissatisfaction among stakeholders, unethical software products and services, conflicts among stakeholders, unfair requirements that undermine the integrity of the software project, loss of trust among stakeholders, a lack of transparency, even legal issues, and many others will occur in RE [59, 60, 61].

Furthermore, there are numerous cases in which software caused disasters due to the abovementioned and other RE problems [62, 63, 64]. Such cases show a lack of attention to ethical concerns during the RE processes [65]. Most of the time, software engineers place a high focus on functional requirements [10].

Many organizations, such as IEEE, ACM, the British Computer Society, and others, have developed codes of ethics for computing professionals [55, 66, 9, 67, 68].

Moreover, the Electrical and Electronics Engineers Computer Society (IEEE-CS) and ACM jointly revised their code of ethics [69, 9]. Nevertheless, those codes of ethics are broad and lack the breadth and depth to apply them in specific domain areas like RE [6]. Most of them have become challenging to exercise on a day-to-day basis since they are broad and cover a wide range of aspects that a software engineer takes into account in many dimensions [6, 48].

Practically, software developers could not uphold these standards, and most did not know what these codes of ethics were [6, 55]. Surprisingly, as Lurie and Mark [10] stated, most of the members of these societies do not even understand the codes. In addition, Ammar and Karim *et al*. [6, 11] explored the absence of a framework to aid software engineers as guidelines to consider ethical concerns during software development and the need for more research work in the software engineering code of ethics.

Even though some works were done to integrate software engineering code of ethics to software development lifecycles [4, 6, 10, 48], none explicitly investigated the RE phase's ethical concerns. It is impossible to effectively address ethical concerns and talk about software project success when RE is practiced without considering ethical concerns [10, 70].

To minimize wrong requirements, ethical concerns should be considered wisely [71, 72]. The above discussions indicate the need for a framework for RE ethics that will be used as a guideline to uphold ethical principles that promote the consideration of ethical concerns in requirements. Developing such a framework helps to enhance the quality and usability of software products, services, and deliverables [10, 66].

To the best of the researchers' knowledge, not much research that were done on RE ethics. Therefore, this research work proposes a RE ethics framework that guides software engineers to consider and address the identified concerns during RE by answering the following research questions:

1 What are indispensable ethical concerns of software that should be considered explicitly during requirements engineering?

   1.1 How do software engineers address ethical concerns when conducting requirements engineering?
   1.2 Are there any guidelines or methods software companies use to identify serious ethical concerns in requirements engineering?

2 What are the basic components of the proposed framework?
3 What should be a suitable framework, in the software development process, for integrating the ethical concerns of requirements during software requirements engineering?

1.4 Objectives of the Research

General Objective

The general objective of this research is to develop a framework for software requirements engineering ethics aimed at enhancing the consideration of ethical concerns throughout requirements engineering.

Specific Objectives

The specific objectives of the research are to:

1 Identify ethical concerns that should be considered during requirements engineering.
   1.1 Investigate how ethical concerns are being put into practice in the real working environments of software companies during requirements engineering.
   1.2 Discover guidelines or methods employed by software companies for addressing ethical concerns in the process of software requirements engineering.
2 Find out the core components that can be used to develop a framework for software requirements engineering ethics.
3 Propose and evaluate the framework.

1.5 Research Methodology

For the research, we followed an inductive research approach and qualitative research design. Data were collected using purposive sampling, interviewing, and Focus Group Discussions (FGDs). Four thematic analysis techniques are utilized to analyse the acquired data. Besides, the proposed framework is evaluated by purposively selected experts using five-point Likert scale questionnaires and focus group discussion questions. In the sequel, we discuss the research process and major research methodologies utilized for the research.

1.5.1 The Research Processes

The interpretivist paradigm is appropriate for inductive qualitative thematic research as it is concerned with understanding human experience, exploring meanings, and acknowledging subjectivity [73]. Unlike positivism, which tries to identify objective facts based on quantifiable data, interpretivism seeks reality as social constructs and context dependent [74]. This makes it more suitable for thematic analysis that is aimed at capturing patterns and meanings in qualitative data [75].

We chose interpretivism as a research paradigm since it is more suitable for this research. One of the reasons for using interpretivism is that it fits with inductive research, where themes are not preconceived but taken from data [76]. Knowledge is created between participants and researchers, as opposed to being an objective reality [77]. It also recognizes subjectivity and meaning-making, realizing that knowledge is shaped by language, culture, and social interaction [78]. Contextual understanding is also the other since interpretivism emphasizes transferability, making certain that findings are significant in a particular context and institutional setting [79].

This research employed an inductive approach rather than relying on a preexisting theoretical framework or hypotheses. We attempted to construct knowledge from empirical evidence collected through interviews and FGDs with seven case companies.

The inductive research approach starts with data collection and analysis to arrive at theories or patterns. It is involved in building knowledge from the bottom up,

allowing data to dictate findings [73]. In inductive research, the researcher will initially gather data, typically using interviews, FGDs, cases, and observations, and then seek the emerging patterns or themes directly emanating from data [75].

The inductive research approach is very useful for the context of this research as it directs the analysis of data obtained from interviews and FGDs. This study does not aim to disprove or prove already formulated hypotheses but allows themes to be emergently constructed, aiding in the development of the proposed framework based on the data [80]. Upon analysis, the findings are synthesized into a framework, presenting an ordered solution for the identified problems.

For this research, following the inductive approach helps to get rich contextual data. The inductive research approach is vital to the development of the proposed framework. The proposed framework is not constrained by current theory but instead is decided by themes emerging within the data.

This research uses a qualitative research design in an effort to gain a detailed insight into the case software industries' practices, trends, and experiences to identify ethical issues of requirements. The qualitative research design and thematic analysis provide a strong finding that becomes the basis for the proposed framework. Data was collected through structured and semi-structured interviews and FGDs from purposively chosen software companies that have a diverse range of experiences and sizes. Subsequent data gathering, thematic analysis is used to identify patterns/themes, categories, and codes that emerge from the data. The use of diverse case companies allowed an in-depth investigation of the topic at hand, while the thematic analysis allowed for conclusions that were translatable into contextually relevant solutions.

The major processes involved during this research are problem formulation, review of literature, data collection, data analysis, framework development and evaluation. The fillowing diagram, Figure 1.1, shows the entire processes involved in the research.

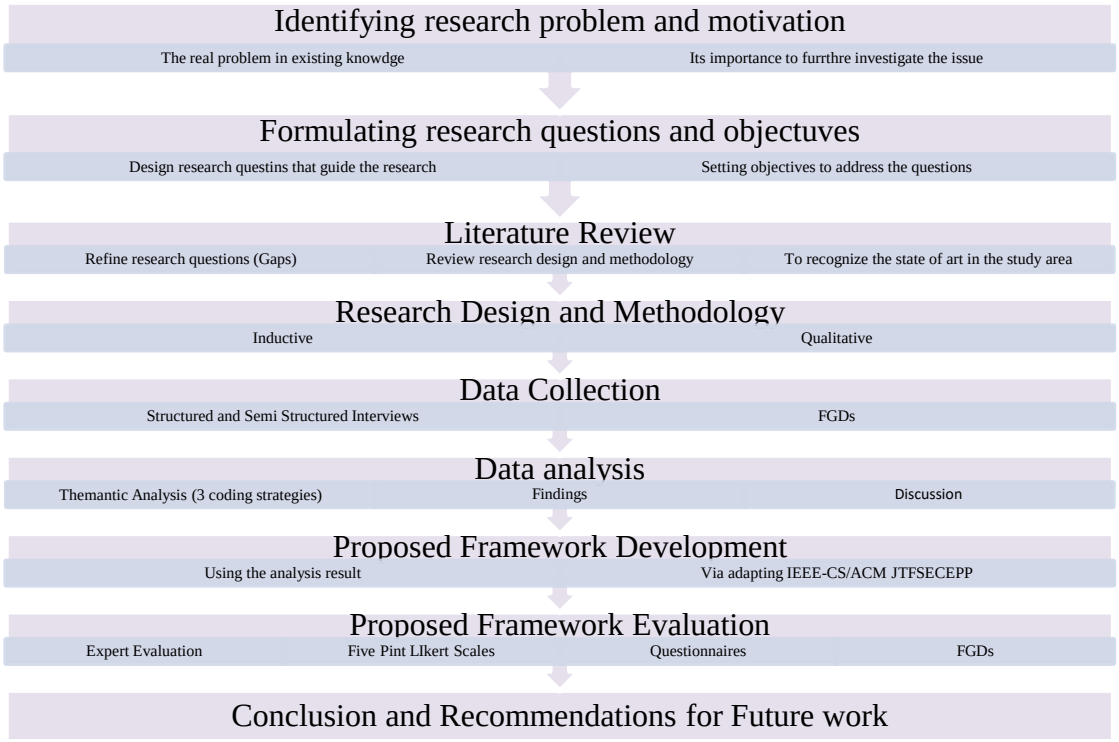

Figure 1.1 Overview of the Research Process

1.5.2 Study Population and Sampling Technique

The purposive sampling technique is used to select samples since it is more appropriate to address the research questions by gathering in-depth, detailed, and reliable information [81]. Purposive sampling is a basic nonprobability sampling technique to get suitable quality and reliable data. It is the process of carefully selecting a sample based on the abilities that the sample has [82].

In this sampling technique, an investigator determines what information is necessary and to be known and searches for individuals who can and are willing to provide about required levels of experience and/or expertise [82]. It applies to quantitative and qualitative data. Unlike probability sampling, in the case of purposive sampling, a sample can be selected either before or after the study [83]. Purposive sampling can also be used with different data-gathering techniques. For example, research might begin through a survey, and afterward, purposive sampling may be carried out following the survey findings. Also, a questionnaire, interviews, cases, and other techniques can be used during purposive sampling research.

When this research began, we tried to search and identify about forty active software companies that are engaged in software development in Ethiopia. Among these, seven software companies were selected as cases for the research, and thirty-five participants were chosen purposively in light of their willingness engagement, and the richness and availability of the required data related to the study. Additionally, individual participants from each company were purposively chosen. Participants were selected based on their involvement in RE processes and their willngness to participate in the research.Additionally, few persons in managerial positions are selected to get strategically high-level information regarding the study. The researchers determined the sample size by taking the saturation point as a criterion. After gathering data from thirty participants from those seven companies, many replies corresponding to each interview question became redundant, and many repeating replies have occurred here and there. The researchers then believed that the data collected were sufficient by adding five more participants from the selected case companies.

1.5.3 Data Collection

An interview is the most conventional but very useful Method to collect data. It is an essential tool to gather detailed information from different people in flexible and complex situations. Also, it is crucial to explore the experiences of people [84]. The structured interviews (open and closed-ended) and two FGDs, four members each [12], are employed to gather a rich, in-depth wealth of qualitative data, to get deep insights, and to extract detailed ethical concerns of RE [85, 86]. For this research, we used an intensive interview and FGDs to identify ethical concerns that should be considered during RE.

1.5.4 Data Analysis

Thematic analysis is one of the qualitative research methods that is used to discover, inspect, and interpret concepts from data that are qualitative in nature. It involves categorizing and systematically structuring data according to recurrent themes that show within the data itself. It is used to get a thorough, extensive understanding of the

experiences, viewpoints, and meanings that the data contains about the subjects [87, 88].

In the beginning, the collected data are transcribed and read repeatedly in a recursive manner and made suitable for analysis. Subsequently, the coding strategies are implemented progressively. This research analyzes the collected data using the four types of inductive thematic qualitative analysis coding strategies. These are holistic, initial, attribute, and in vivo codings.

Holistic coding uses a code for every huge group of data in the corpus to catch up on a concept of the broad-spectrum contents and the potential categories that may develop [87]. For our research, we applied holistic coding to capture major themes and categories from the collected corpus data. It is employed to prepare data units for additional coding and categorization and to obtain a general idea of the contents. Initial coding is useful for analyzing discrete parts of qualitative data. After holistic coding, every unit of the data is examined thoroughly, and initial codes are generated as of per discrete unit of the data. In qualitative research, “in vivo” coding is one of the most widely used coding strategies to take the words respondents’ said as it is. For the research, vital real words of the partakers are taken as stated through the use of in vivo coding.

Moreover, attribute coding is utilized to present information concerning the participants' demographics and other vital information that will be used for future reference and administration. In this research, attribute coding is used to analyze the participating firms and each attendee’s demographic details [12].

1.5.5 Evaluation

The proposed framework is evaluated for its validity and reliability [89, 90] using questionnaires, Likert scales, purposive sampling techniques, expert evaluations, and FGDs.

For the evaluation of the proposed framework, we adopted a mixed-method approach by incorporating both closed-ended and open-ended questionnaires, drawing from established literature [91]. This method enables a thorough assessment of the

framework from multiple perspectives. The closed-ended questionnaires were designed using five-point Likert scales to quantitatively assess various aspects of the framework's validity and reliability. These include face validity, content validity (plus the content validity of each of the framework components), construct validity (plus the construct validity of each of the framework components), reliability, and internal consistency. Additionally, open-ended questions are incorporated to collect the perceptions of participants towards the proposed framework. These questions aimed to identify the most challenging aspects to follow and the most crucial components of the framework, providing valuable context to complement the quantitative data. Furthermore, using open-ended questions, we assessed the placement or allocation of clauses for each of the RE's ethical concerns to ensure the framework's thoroughness and relevance.

The Likert scale, as the proposed framework evaluation method [92], was chosen because it allows us to measure the opinions, perceptions, and input that experts have regarding the suggested RE framework [93]. Additionally, we used the odd-number Likert scale choices since, in most cases, this is preferred by most scholars [94].

The sentiments on the Likert scale are “very good,” “good,” “neutral,” “poor,” and “very poor." We added the neutral option to the five-point Likert scale. Because, instead of forcing them to choose an alternative that doesn’t reflect their thinking, adding the neutral option helps respondents to answer neither positive nor negative responses and gives them freedom [95]. The evaluation results of the Likert scale are analyzed using an ordinal data scale [94]. We describe the data as a summary of the percentage of responses in each framework’s component of the Likert item.

Furthermore, expert evaluation is used to evaluate the proposed framework. Expert evaluation is a method mostly used in different fields of study to assess and validate the quality of a system, product, artifacts, or content by looking for opinions and judgments from domain experts [96]. The expert evaluation method is frequently used by people in their everyday tasks [97]. The fundamental assumption of the method is that the combined expertise of a given set of individuals is always greater than the individual knowledge of each member of the group.

Expertise demonstrates that tasks are typically realized more successfully when they draw on the collective knowledge and expertise of individuals [98]. In this research, expert evaluation is used to assess the proposed framework. Forty experts are purposively selected from software development companies specializing in computer science, software engineering, technologies, information multimedia, and information systems. The chosen experts are equipped with well theoretical and practical knowledge of software engineering, particularly RE. In addition, the experts' willingness is considered when choosing them.

In addition, FGDs, among the most common tools used in support of qualitative research, were utilized during the evaluation of the proposed framework. Currently, a focus group discussion is widely used in different fields of study [99]. It is a crucial tool for collecting and analyzing data [100, 101]. Using FGDs has many advantages; for instance, participants can influence one another, and the interaction among groups can create synergy among participants. Especially when there are difficult situations to get participants individually, FGDs are the preferred method.

1.6 Research Scope and Limitations

In this research, ethical issues that should be taken into account during the RE processes are assessed, and a framework for RE ethics is developed. The research does not include other phases of a software development project's lifecycle.

1.7 Ethical Considerations

We tried to keep the privacy of those companies and individuals who participated in the research. To start the research, we obtained formal letters from Addis Ababa University, IT Doctoral Program. Furthermore, an informed consent form is prepared and given for participants to sign to ensure their willingness.

Additionally, we used coded names for both participating software companies and individuals. Besides, real results, whether favorable or unfavorable, are presented as discovered.

1.8 Organization of the Dissertation

The thesis is organized into seven chapters, including the current chapter. Reviews of literature in the research area are discussed in Chapter Two. We discussed related works under Chapter Three. Chapter Four presents the analysis results of the research. In this chapter, the data collection procedures, the findings, and the analysis results are described. Chapter Five discusses the proposed framework for RE ethics and its components. The proposed framework's evaluation results are discussed in Chapter Six. Finally, Chapter Seven presents concluding remarks and recommendations for future research.

Chapter Two – Literature Review

2.1 Introduction

Nowadays, one of the major careers in the world is software engineering [102, 103]. As the area of software engineering has grown, it has become the backbone for every technological innovation globally [102].

Like other academic fields, software engineering is subject to legal and social constraints that limit the group of employees. Software engineers have a professional responsibility to act and make ethical decisions io complete their work in addition to possessing the necessary technical skills. So, discussing ethics becomes a fundamental issue in the software engineering field of study.

Within this chapter, we talk about basic concepts and the state of the art in the study area. Specifically, the chapter deals with RE, frameworks in RE, ethics in RE, ethical considerations in software development, software engineering ethics, and software engineering code of ethics.

2.2 Requirements Engineering

Requirements engineering is a component of engineering that concentrates on the objectives, constraints, and capabilities of a system [56]. The process of determining the services that a client needs from a system and the limitations that shape its development and operation is known as RE [16, 17].

RE has a great role in developing successful software [104, 105]. Not identifying the right requirements results in faults in system components and specifications [106]. It is essential to ensure whether a software solution correctly solves a problem or not. Initially, we must understand the problem that needs to be solved. Then, the relevance of the problem and stakeholders must be justified and determined, respectively. Poorly specified requirements can lead to several challenges that affect different system stakeholders. RE offers an effective framework for identifying customers' real needs, evaluating their expectations, reviewing feasibility, clearly outlining solutions,

validating and managing requirements, and articulating practical solutions as they are integrated into an operational system [107].

Under RE, there are five key processes including “requirements elicitation, requirements analysis and negotiation, requirements specification, requirements validation, and requirements management” [56, 106, 108].

The first sub-process, requirements elicitation, involves collecting requirements from key stakeholders. Problem identification, constraints and scopes of the system, problem volatility, and understanding are clearly stated in this sub-process of RE [56, 107]. Once requirements have been collected, the necessity, consistency, completeness, and feasibility of requirements will be checked during the requirements analysis sub-process. Through negotiation and prioritization, conflicting requirements will be resolved among a variety of stakeholders [108].

Requirements specification involves specifying and describing the function and nonfunctional requirements of the proposed system, constraints that will control its development, input, and output controls, and determines every assigned system element. The specification also helps to aid communication between stakeholders and developers of the proposed system [106].

Requirements validation inspects systems specifications to ensure whether all system requirements have been described clearly or not. It ensures that discrepancy oversights and faults have been identified and approved, and products and deliverables obey the rules and standards of software artifacts. The requirements management sub-process involves identifying, controlling, and tracking requirements, and change requests throughout the development [56, 102].

There are various forms of requirements. These include user requirements and system requirements. System requirements hold functional and non-functional requirements. Requirements that designate functions and services that a system should offer and its interfaces are known as functional requirements. Non-functional requirements are levied through the situation in which the system is operating [56, 102].

2.2.1 Challenges in Requirements Engineering

RE faces different challenges with different perspectives [16, 105]. To be competitive, organizations, institutions, and individuals face digitalization and globalization challenges from time to time [109]. The growth of software technological innovations also goes fast in different dimensions such as in artificial intelligence [110], critical software systems [111], social media application software [112], and so on. The fast growth of software technologies, the close intimacy between the environments of software and the software itself, customers' growing demands, and globalization make the requirements process of software development more challenging [16].

RE challenges emerge from different dimensions. The challenges may occur due to the RE process; technological crises like obsoleting requirements; stakeholders' conflicts, time frame, economic crisis, and other external factors [105]. Wnuk and Pavan [109] identified the challenges of RE, which include insufficient resources, requirements prioritization problems, the presence of incomplete requirements, aligning product, project, and organizational opinions, implementing business values, and considering time and resource constraints. Similarly, Asghar and Mahrukh [105] identified "comprehending stakeholders' needs; incomplete and inconsistent process description; and verification and validation of requirements" as the leading challenges confronted in requirement engineering. In addition, communication gaps, omitting non-functional requirements, quality-related problems, method utilization, usage problems, and requirement prioritizations are the major challenges recognized by Shah and Patel [16].

Directly or indirectly, software engineers are playing an increasingly relevant role in the RE processes. Accordingly, these professionals have noteworthy prospects to benefit or damage others and to persuade assigned duties or do destruction. To be sure of a positive contribution to society at large, software engineers must endeavour as much as possible to make software engineering a helpful and valued profession.

2.2.2 Frameworks in Requirements Engineering

Different scholars define a framework in different ways. The Cambridge Dictionary [113] explains a framework as "a supporting structure around which something can be built; a system of rules, ideas, or beliefs that are used to plan or decide something; a structure around or over which something is built; the ideas, information, and principles that form the structure of an organization or plan."

Similarly, the Oxford Dictionary [114] describes a framework as "the parts of a building or an object that support its weight and give it shape." It also defines it as "a set of beliefs, ideas or rules that are used as a basis for making judgments, decisions, etc.; and the structure of a particular system, which can be the basic framework of a society, a regulatory/legislative framework, a framework for something." Medical dictionary [115] defines a framework as "the basic structure about which something is formulated or built."

In software engineering, there are different definitions of a framework. Most popularly a framework can be defined as "a platform" that helps software engineers to develop software programs, systems, and other related artifacts [116].

There are a variety types of frameworks in different fields of study. For instance, if we take engineering and medical fields of study, there are many frameworks. Three examples of engineering frameworks include "A safety engineering framework for open adaptive systems" [117], a social engineering attack framework [118], a security RE framework [119], and so on. A few examples of medical frameworks encompass the Meta organic framework for biological and medical applications [120, p. 6260], huge medical data-driven cloud-based systems for home diagnosis services [121], frameworks for learner assessment in medicine [122], etc. A framework's purpose, aim, and meaning differ in each field of study.

In the field of RE, several frameworks have been proposed. Schon *et al.* [123] proposed a framework in order to enhance agile RE. Arayici *et al.* [124] developed a RE framework to integrate systems development for the construction industry. Edward and Geoffrey [125] proposed a RE framework within the aim of process

improvements in software industries that are small to medium-sized. Ashok and Deraman [126] propose requirements vagueness evading framework with the intention of helping requirements engineers to write unambiguous requirements. Bates [127] developed an ethical framework for engineering professionals and students to discourse ethics from twofold standpoints. These are to enhance the knowledge of engineering graduates regarding their professional code of ethics responsibilities and to assist experts who confront different problems and encounters related to acting unethically and to protect them from doing unethical activities. Rehman and Volker [128] suggested a security RE framework for cyber-physical systems. The chief aim of the proposed framework was to address the concern of security requirements elicitation for different cyber-physical system components. It helps the elicitation of security requirements accompanied by software facets.

Van de Poel [129] suggested an ethical framework for emerging technologies' acceptability in social experiments. The framework is developed according to "bioethical principles for experiments with human subjects: non-maleficence, beneficence, respect for autonomy, and justice." Similarly, Lurie and Shlomo develop a framework for ethics that combines the professional expertise of software engineers with their ethical obligations [10].

In RE, commonly frameworks are used to integrate methods, techniques, standards, and research problems [130]. A "framework" is typically utilized to explain a fundamental structure or set of guidelines, rules, or ideas that serve as a basis for comprehending, creating, or resolving problems [131]. For example, a framework in software development is a pre-made, reusable code structure that programmers can utilize as a starting point when creating particular kinds of applications [132].

A framework is an organized method of thinking or doing things. It offers a collection of guidelines that deal with problems in a particular field. It mainly serves as a basis for people or organizations to confront issues or activities methodically and in an orderly way. Frameworks are suggestions, not strict laws, that support people in a particular field of study in making morally sound decisions in various work-related contexts [133].

A framework in ethical discourse is a collection of principles or rules that direct moral judgments [133]. The IEEE and ACM, for instance, have codes of ethics that provide guidelines for moral behavior in computing.

Within the scope of this research, the framework is considered a set of pieces of advice as clauses that guide software engineers to ethical decision-making by considering the suggested bits of advice to address the stated fundamental ethical concerns during RE. The meaning of "clause" varies based on the context in which it is used. For example, a clause in computer programming could relate to a particular segment of the syntax of the language or a logical condition inside a statement [134]. When referring to particular sections of the IEEE-CS/ACM JTFSECEPP that narratively describe different principles, the term "clause" is used [68]. In the context of this research, clauses are used to indicate the suggested bits of advice that are adapted from IEEE-CS/ACM JTFSECEPP clauses to deal with the primary identified RE ethical issues.

2.3 Ethics

2.3.1 Introduction

Ethics is one of the core branches of philosophy which covers a wide range of aspects starting from principal questions about the nature of humanity and autonomy to very real-world inquiries around the morality of physician-assisted suicides and experimentations on creatures [135].

Greek word “ethos,” meaning “good,” is where the word ethics originates and indicates “character, habit, customs, ways of behavior, etc.” [135]. Additionally, some scholars define ethics as “moral philosophy” that reflects behaviors, customs, characters, and so on [136]. It is the thoughtful examination and identification of the good things and/or bad things regarding human behavior and individual responsibility. To put it differently, ethics indicates the “good” things and means in order to find those things, and the bad things and the way which helps us to avoid that bad thing(s). Lillie [137] stated that ethics denotes “what to be done to achieve what is good and what ought not to be done to avoid what is evil.”

Ethics is a science at the same time a philosophical discipline. As a science, ethics deals with a set of organized bodies of knowledge of what humans should be through justification of what one is. Similarly, as a field of philosophy, ethics examines norms and procedures through which we live and comprises the reasonings of those values and guidelines. Ethics is a theoretical and practical study since it offers important ideologies up on the foundation of ethical conclusions that are reached regarding the final goal being achieved, and how to go about achieving it [138].

Ethics is not simply a collection of codes. Rather, ethics shows the means to find "what is good and how to get it" as an advisory service, not as an obligatory way of manner. Ethics is not mainly to oblige or restrict somebody to do something, but it is a voluntary action [138].

2.3.2 History of Ethics

It is difficult to trace the origin and moral perception. Nevertheless, from the literature, we can give an overview of the beginning of ethics [139]. As scholars stated, the first ethical thoughts were transmitted by the word of elders and fathers [137]. But as the communities learned, to be able to read and write, they began to put their ethical beliefs in a written form. These documents worked as primary proof of the advent of ethics. According to Western Philosophy, the history of ethics began with the arrival of Socrates around the fifth century [139]. The main goal of Socrates was to imitate his companion individuals towards the necessity of rational criticism of their beliefs and practices [137]. To search for reasonable realms for ethical encouragement, Socrates focused on the logical interrelationship among values and facts in which he constructed ethical philosophy.

Plato introduced objectivism to moral truth later. He first proposed an organized perspective on God, nature, and humans before identifying driveable ethical principles. The major intention of Plato regarding ethical philosophy was to show an approach concerning a dream of the good. Plato originated religion and ethical ideals. The successor philosopher after Plato, Aristotle, focuses on the necessity of exercising the knowledge of good by demarking practically applicable guides for persons working to live virtuously [139]. Contrasting to his forerunner, Aristotle banned an

all-embracing “Form of Good. Instead, he thought that every item, practice, action, or individual has dissimilar ultimate goodness. The distinction between morality-weak and morality-robust individual depends alone on their behavior, not on their desires or wishes.

In general, Socrates was the beginner to appreciate the necessity to state ethical notions and endeavour to launch a worldwide standard. Plato established his standard in unchangeable, common constructs and goodness is measured by his ideal Form. Aristotle converted into practical implementation through which pleasure is realized over the correct act. Following his explanation, societal and individual righteousness are connected. The former philosophers depicted assumptions near moral responsibility. On behalf of Plato, ethically erroneous conclusions are carried out in mistake due to a deficiency of knowledge. Furthermore, Aristotle supplemented the likelihood of selecting to do wrong as additional errors [139].

Nowadays, ethics is recognized not only by philosophers or social scientists but also in all areas and fields of study. Especially in the software engineering industry, we have come to a time when ethics is very important and relevant to people’s daily activities [140]. That is why ethics is considered a professional code of conduct in almost all fields of study.

2.4 Software Engineering and Ethics

2.4.1 Software Engineering Ethics

The most profound realization is found within ethics, which discloses how people are constantly determined to live [53]. Intentional action that has an impact on people’s lives, whether positively or badly, is referred to as ethics [55, 66]. Ethics in engineering plays a vital role in offering harmless and valuable technological services and products [8]. Software engineering ethics might be characterized by software engineers’ practical and moral decision-making processes [56, 9].

Different scholars define ethics according to their perspectives. Among them, the prominent individuals whose definitions of ethics align most closely with the "IEEE-CS/ACM JTFSECEPP" are discussed below.

The famous German philosopher, Immanuel Kant, defined ethics in terms of his deontological moral theory. According to Kant, the idea of obligation and the intrinsic value of every person serve as the foundation for ethics [33]. Instead of concentrating just on the results of actions, his definition highlights the significance of obligation, logic, and commitment to moral principles in ethical decision-making. Similar to the IEEE-CS/ACM JTFSECEPP, Kant emphasizes the public good and upholds professional integrity.

Stuart [34] stated that moral judgments should seek to optimize overall pleasure and minimize overall pain. He states that a course of behavior is ethically justified if it results in the utmost happiness in support of the vast population. Aristotle, a well-known Greek philosopher, describes ethics as a quality that is demonstrated by habitual behavior found between excess and deficiency and developed via practical wisdom to lead a fulfilling life for people [35].

A well-known person in software engineering ethics, Donald Gotterbarn, stated ethics as "the systematic study of our general moral concepts and our moral judgments, including the moral judgment that we ought to engage in such a study. His definition mainly implies that in addition to upholding honesty, equity, and respect for everyone, software engineers should act and perform in the public interest and ensure that their work reduces harm while benefiting society [36]. He said that “software engineering involves more than adherence to rules; it includes the responsibility to consider the broader impact of software on society.”

Michael Davis, a famous philosopher on business ethics, defines ethics as “… the discipline that examines one's moral standards or the moral standards of a society and asks how these standards apply to our lives and whether these standards are reasonable or unreasonable, whether they are supported by good reasons or bad” [37]. According to Davis, ethics refers to the methodical study of morality, with an emphasis on establishing what is good or bad and how these conclusions relate to decisions made by people in different situations [37]. Davis contributes to software engineering ethics. He emphasizes the need for honesty, transparency, and respect for stakeholder rights for engineering professional ethics.

Base [38] known for her work in computer science and ethics, defines ethics as "a set of moral principles that govern human behaviors." Basse's definition shows ethics as a framework of ethical guidelines that direct and control human conduct. It implies that ethics serve as a foundation for evaluating actions and decisions in terms of their alignment with moral principles. She discussed the ethical and social implications of computing, emphasizing concerns about security, privacy, and professional accountability, which are essential elements of the IEEE-CS/ACM JTFSECEPP.

According to the IEEE-CS/ACM JTFSECEPP ethics can be defined as adhering to principles and practices that prioritize the interests of the public, clients, and employers, as well as high professional standards, justice, integrity, lifelong learning, and the profession's progress. This assures trustworthy and accountable behavior in all professional settings [39].

In one or another way, these individuals have provided important concepts and principles to computing and software engineering professional codes of ethics developed by different organizations, including the IEEE-CS/ACM JTFSECEPP. All the above definitions of ethics share common concepts such as transparency, professional integrity, public accountability, and considering the wider effects of software on society. They provide a foundation for ethical decision-making and professional conduct within the software engineering field of study.

After reviewing ethics definitions given by the above-stated scholars and organizations, we tailored contextual definitions that will be fit for our research scope. Specifically, our definition of ethics aligns with the IEEE-CS/ACM JTFSECEPP.

So, in the context of this research, ethics is considered to follow the specific advice stated in the proposed framework for ethical decision-making during RE. This includes ensuring the requirements prioritize public safety and well-being communicating with customers while maintaining the confidentiality and privacy; upholding ethical principles for achieving high quality, complete, unambiguous, transparent, exclusive requirements gathering and documenting requirements that align with the client's real goals and interests, and ensuring they are ethical and feasible; engaging and understanding stakeholders; be transparent about any potential

ethical issues and conflicts of interest; updating knowledge and skills and contribute to the professional community by sharing insights and advancements; ensuring that all activities are well-documented, traceable, and reliable, thereby maintaining stakeholder trust throughout the RE life; overlook ethical concerns and providing honest, evidence-based recommendations; and commit to ongoing professional development and self-reflection on ethical issues.

There is a significant association between ethics and software engineering. Software engineers create software artifacts, ranging from easy, like application programs, to complicated ones, for instance, software for rockets, aircraft, and automobiles, among other safety-critical systems. These types of systems affect society in either a beneficial or harmful way on the public and the environment [141].

Principally, Software engineering, which builds software services and products, views itself as a technical career. Nevertheless, in today's world, software systems play either a direct or indirect role in the daily activities of human beings. Software engineers will make key decisions on the side of a software system throughout software development processes. These decisions have an excessive effect on society and clients. The implications of this demonstrate that software engineers should take into account for their decisions to protect the public and customers from the unforeseen hazards of those software products and services [55, 66]. So, during their decision, ethical issues related to that software product and/or service should be critically considered and analyzed [142].

2.4.2 Ethics in Software Development

Nowadays, software is found ubiquitously. The uses of software are observed in many of our daily works. Among them, some are children's electronic toys, cellophanes, televisions refrigerators, and a computer's onboard devices within cars, etc. Software services and products offer a vital contribution in many dimensions. Some of its contributions include providing business information solutions for enterprise resource planning, and facilitating, structuring, and safeguarding operations of small, medium, and large firms. Those who develop software systems have a remarkable chance to do good or evil, to enable or influence others to do good or harm. This suggests that

software engineers ought to have a strong commitment to ensuring that software artifacts are advantageous for all parties involved, as well as for society at large [10].

Features, functions, and infrastructures in software packages often extend beyond what end users want and understand. Nevertheless, whether the software satisfies the goals and desires of stakeholders relies upon software artifacts quality and issues considered and incorporated during the software development process [143]. Every day, vast amounts of software are used, and when they adversely affect customers, the situation becomes urgent. Throughout the software development process, developers might have to make morally difficult choices to ensure that users and stakeholders suffer as little as harm as possible. There are many reasons for the inclusion of ethical practices in software development project decision-making [69]. To prevent software from hurting people, software engineers should carefully evaluate how to incorporate ethical principles into their decisions. Software engineers have a duty to the public and their profession. When they develop software, they have to show their willingness to uphold their professional code of ethics as an element of their contributions to society. Ammar [6] stated that "Professional codes of ethics do not only require doing the right things for the client, but it also includes doing the right thing for the society."

The public's faith in software artifacts, their image, inner and outside communication amongst several collaborators, and their degree of contentment intends to be better off when ethical issues are taken into account during software development initiatives [144]. To highlight frequent ethical problems in software development, Berenbach [69] presents nine main categories for ethical dilemmas. Mea culpa, rush jobs, not my problem, red lies, fiction ware versus vapourware, lack of diligence, and canceled vacations are some examples of dilemmas. There should be a clear blueprint that tells software developers what needs to be considered and integrated with regard to ethical issues in the course of software creation. Software engineers will also be better able to understand and follow ethical principles when they develop software development projects [6].

2.5 Software Engineering Code of Ethics

Within the discipline of software engineering, software codes of ethics require high attention [6, 10, 145]. Taking ethical considerations into account during software development processes will enhance stakeholder satisfaction, internal and external communication, public trust, and the professional image of software artifacts [55]. There should be a clear set of guidelines or frameworks that take these ideas into account and show what parts of ethical issues in software development projects should be taken into account through software development projects. Moreover, identifying ethical issues in software development projects clearly will give software engineers a better understanding and ability to follow codes of ethics in their discipline [6] to give clear, real, appropriate, and precise requirements. Nevertheless, in actual software development, ethical considerations are not considered [6, 69, 55, 11].

One of the topics of software engineering that requires more research attention is software engineering codes of ethics (SWECODE) [6]. It immensely affects people's daily lives and might take up a large portion of the practice. Software engineers' ability to adhere to established codes of ethics significantly impacts the degree of harm that malfunctioning software technology can cause humans [11].

Many organizations, such as IEEE, ACM, the British Computer Society, and others, have developed codes of ethics for computing professionals [129, 146, 147, 148, 149]. Nevertheless, those easily accessible codes of ethics provide software engineering experts with a comprehensive road map. These codes of ethics are deficient in extensiveness and deepness. The codes of ethics won't evaluate and proceed onward with ethical concerns regarding software artifacts and development phases in a detailed manner [48]. The majority of those are complex and encompass a wide range of aspects, they are difficult to follow on a daily basis [11, 148].

Chapter Three – Related Work

3.1 Introduction

In order to find peer-reviewed journal articles, conference papers, and other studies regarding requirements engineering and ethics, an intensive search was carried out across different databases, including Google Scholar, ACM Digital Library, IEEE Xplore, ResearchGate, PubMed, and SpringerLink. To limit the search, particular keywords were used such as "Ethics in Software Engineering or Requirements Engineering", "Ethical concerns of Requirements Engineering", "Ethical Considerations in Software Development" and "Ethics in Software Requirements Engineering", "Ethics in Requirements Engineering", “Ethics and Requirements Engineering", "Requirements Engineering Ethics" or “RE Ethics”, "Ethical Framework for Requirements Engineering" or “Ethical Framework for RE ”, “Codes of Ethics in Software Engineering” or “SE Codes of Ethics ”, etc. Using such kinds of keywords, we retrieved scholarly articles.

After gathering the available sources, we followed Keshav's [150] three-pass reading approach. In the first pass, to get the general aim of the papers, we read the abstracts, introduction, section headings, headings, and references at a glance. During this time we were able to identify the type of papers, their relevance to our research, their correctness, and their clarity to understand and review them easily. After this stage, among many downloaded papers, we chose about eighteen that are related to the concept of RE ethics, issues in RE, software engineering codes of ethics, and ethics in software development. In the second pass, we read each section of the papers carefully to grasp the papers' content and aim. Then, among the eighteen, fourteen papers that are more related to the research objectives were chosen. Each of these papers was reviewed critically. These papers are most likely related to software engineering codes of ethics and ethics in software development and RE, papers that have a resemblance with RE from different perspectives.

The nominated papers were intensively reviewed in line with the gaps that each paper raised, the methodologies used, the contributions of the paper, and the limitations and

recommendations for further investigation. In the following section, we discussed some of those reviewed papers.

Various scholars have examined the ethical considerations underlying software systems from different perspectives. Many researchers examine ethical problems concerning software quality, human elements of software artifacts, socio-political issues, and concerns associated with reliability, privacy, accessibility, and accuracy. Researchers have also looked into software engineering codes of ethics. Here, we review some of the major research works, including the IEEE-CS/ACM JTFSECEPP.

3.2 IEEE-CS/ACM Joint Task Force Code of Ethics

The well-known organizations, IEEE-CS, and ACM together proposed codes of Ethics and Professional Practice with eight principles (IEEE-CS/ACM JTFSECEPP) [9, 148]. Table 3.1 shows the full version of the IEEE-CS/ACM JTFSECEPP.

Table 3.1 The Eight Principles

IEEE-CS/ACM JTFSECEPP – Principles and Clauses
Principle 1 Public: Software engineers shall act consistently with the public interest. In particular, software engineers shall, as appropriate:
1.01 Accept full responsibility for their own work.
1.02 Moderate the interests of the software engineer, the employer, the client, and the users with the public good.
1.03 Approve software only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment. The ultimate effect of the work should be to the public good.
1.04 Disclose to appropriate persons or authorities any actual or potential danger to the user, the public, or the environment, that they reasonably believe to be associated with software or related documents.
1.05 Cooperate in efforts to address matters of grave public concern caused by software, its installation, maintenance, support or documentation.
1.06 Be fair and avoid deception in all statements, particularly public ones,

IEEE-CS/ACM JTFSECEPP – Principles and Clauses
concerning software or related documents, methods and tools.
1.07 Consider issues of physical disabilities, allocation of resources, economic disadvantage and other factors that can diminish access to the benefits of software.
1.08 Be encouraged to volunteer professional skills to good causes and to contribute to public education concerning the discipline.
Principle 2 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. In particular, software engineers shall, as appropriate:
2.01 Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.
2.02 Not knowingly use software that is obtained or retained either illegally or unethically.
2.03 Use the property of a client or employer only in ways properly authorized and with the client's or employer's knowledge and consent.
2.04 Ensure that any document upon which they rely has been approved, when required, by someone authorized to approve it.
2.05 Keep private any confidential information gained in their professional work, where such confidentiality is consistent with the public interest and consistent with the law.
2.06 Identify, document, collect evidence and report to the client or the employer promptly if, in their opinion, a project is likely to fail, to prove too expensive, to violate intellectual property law, or otherwise to be problematic.
2.07 Identify, document, and report significant issues of social concern, of which they are aware, in software or related documents, to the employer or the client.
2.08 Accept no outside work detrimental to the work they perform for their primary employer.
2.09 Promote no interest adverse to their employer or client, unless a higher ethical concern is being compromised; in that case, inform the employer or another appropriate authority of the ethical concern.

IEEE-CS/ACM JTFSECEPP – Principles and Clauses
Principle 3 Product: Software engineers shall ensure that their products and related modifications meet the highest professional standards possible. In particular, software engineers shall, as appropriate:
3.01 Strive for high quality, acceptable cost, and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.
3.02 Ensure proper and achievable goals and objectives for any project on which they work or propose.
3.03 Identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects.
3.04 Ensure that they are qualified for any project on which they work or propose to work, by an appropriate combination of education, training, and experience,.
3.05 Ensure that an appropriate method is used for any project on which they work or propose to work.
3.06 Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified.
3.07 Strive to fully understand the specifications for software on which they work.
3.08 Ensure that specifications for software on which they work have been well documented, satisfy the users' requirements and have the appropriate approvals.
3.09 Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates.
3.10 Ensure adequate testing, debugging, and review of software and related documents on which they work.
3.11 Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.
3.12 Work to develop software and related documents that respect the privacy of those who will be affected by that software.
3.13 Be careful to use only accurate data derived by ethical and lawful means, and use

IEEE-CS/ACM JTFSECEPP – Principles and Clauses
it only in ways properly authorized.
3.14 Maintain the integrity of data, being sensitive to outdated or flawed occurrences.
3.15 Treat all forms of software maintenance with the same professionalism as new development.
Principle 4 Judgment: Software engineers shall maintain integrity and independence in their professional judgment. In particular, software engineers shall, as appropriate:
4.01 Temper all technical judgments by the need to support and maintain human values.
4.02 Only endorse documents either prepared under their supervision or within their areas of competence and with which they are in agreement.
4.03 Maintain professional objectivity with respect to any software or related documents they are asked to evaluate.
4.04 Not engage in deceptive financial practices such as bribery, double billing, or other improper financial practices.
4.05 Disclose to all concerned parties those conflicts of interest that cannot reasonably be avoided or escaped.
4.06 Refuse to participate, as members or advisors, in a private, governmental or professional body concerned with software related issues, in which they, their employers or their clients have undisclosed potential conflicts of interest.
Principle 5 Management: Software engineering managers and leaders shall subscribe to and promote an ethical approach to the management of software development and maintenance. In particular, those managing or leading software engineers shall, as appropriate:
5.01 Ensure good management for any project on which they work, including effective procedures for promotion of quality and reduction of risk.
5.02 Ensure that software engineers are informed of standards before being held to them.
5.03 Ensure that software engineers know the employer's policies and procedures for

IEEE-CS/ACM JTFSECEPP – Principles and Clauses	
	protecting passwords, files and information that is confidential to the employer or confidential to others.
5.04	Assign work only after taking into account appropriate contributions of education and experience tempered with a desire to further that education and experience.
5.05	Ensure realistic quantitative estimates of cost, scheduling, personnel, quality and outcomes on any project on which they work or propose to work, and provide an uncertainty assessment of these estimates.
5.06	Attract potential software engineers only by full and accurate description of the conditions of employment.
5.07	Offer fair and just remuneration.
5.08	Not unjustly prevent someone from taking a position for which that person is suitably qualified.
5.09	Ensure that there is a fair agreement concerning ownership of any software, processes, research, writing, or other intellectual property to which a software engineer has contributed.
5.10	Provide for due process in hearing charges of violation of an employer's policy or of this Code.
5.11	Not ask a software engineer to do anything inconsistent with this Code.
5.12	Not punish anyone for expressing ethical concerns about a project.
Principle 6 Profession: Software engineers shall advance the integrity and reputation of the profession consistent with the public interest. In particular, software engineers shall, as appropriate:	
6.01	Help develop an organizational environment favorable to acting ethically.
6.02	Promote public knowledge of software engineering.
6.03	Extend software engineering knowledge by appropriate participation in professional organizations, meetings and publications.
6.04	Support, as members of a profession, other software engineers striving to follow this Code.
6.05	Not promote their own interest at the expense of the profession, client or

IEEE-CS/ACM JTFSECEPP – Principles and Clauses
employer.
6.06 Obey all laws governing their work, unless, in exceptional circumstances, such compliance is inconsistent with the public interest.
6.07 Be accurate in stating the characteristics of software on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.
6.08 Take responsibility for detecting, correcting, and reporting errors in software and associated documents on which they work.
6.09 Ensure that clients, employers, and supervisors know of the software engineer's commitment to this Code of ethics, and the subsequent ramifications of such commitment.
6.10 Avoid associations with businesses and organizations which are in conflict with this code.
6.11 Recognize that violations of this Code are inconsistent with being a professional software engineer.
6.12 Express concerns to the people involved when significant violations of this Code are detected unless this is impossible, counter-productive, or dangerous.
6.13 Report significant violations of this Code to appropriate authorities when it is clear that consultation with people involved in these significant violations is impossible, counter-productive or dangerous.
Principle 7 Colleagues: Software engineers shall be fair to and supportive of their colleagues. In particular, software engineers shall, as appropriate:
7.01 Encourage colleagues to adhere to this Code.
7.02 Assist colleagues in professional development.
7.03 Credit fully the work of others and refrain from taking undue credit.
7.04 Review the work of others in an objective, candid, and properly documented way.
7.05 Give a fair hearing to the opinions, concerns, or complaints of a colleague.
7.06 Assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other

IEEE-CS/ACM JTFSECEPP – Principles and Clauses
confidential information, and security measures in general.
7.07 Not unfairly intervene in the career of any colleague; however, concern for the employer, the client or public interest may compel software engineers, in good faith, to question the competence of a colleague.
7.08 In situations outside of their own areas of competence, call upon the opinions of other professionals who have competence in that area.
Principle 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. In particular, software engineers shall continually endeavor to:
8.01 Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process.
8.02 Improve their ability to create safe, reliable, and useful quality software at reasonable cost and within a reasonable time.
8.03 Improve their ability to produce accurate, informative, and well-written documentation.
8.04 Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.
8.05 Improve their knowledge of relevant standards and the law governing the software and related documents on which they work.
8.06 Improve their knowledge of this Code, its interpretation, and its application to their work.
8.07 Not give unfair treatment to anyone because of any irrelevant prejudices.
8.08 Not influence others to undertake any action that involves a breach of this Code.
8.09 Recognize that personal violations of this Code are inconsistent with being a professional software engineer.

As shown above, in Table 3.1, the IEEE – CS/ACM jointly revised their codes of ethics and suggested eight principles that cover many relevant issues for software engineers. These include the public, clients, employers, products, judgments, the

profession itself, co-workers, and oneself. However, their joint task force code of ethics has become difficult to apply and uphold daily. While this code of ethics can provide valuable guidelines, it is challenging to apply it in specific situations. This is because it considers a wide range of perspectives that a software engineer must take into account [10, 11].

3.3 Related Work on Software Development and Code of Ethics

To overcome the challenges observed in the IEEE-CS/ACM JTFSECEPP, Ammar *et al.* [6, 11] conducted a research study on SWECOE by taking one software project development company known as SADA as a case study in Saudi Arabia. The researchers analyzed SWECOE and proposed new classifications for it according to the software development life cycles. The researchers took the four principles of the IEEE-CS/ACM JTFSECEPP (public, client and employer, product, and judgment) and classified them based on the RE, design, code, test, and maintenance phase, and proposed a new SWECOE classification.

The research showed how to adapt the IEEE-CS/ACM JTFSECEPP 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 JTFSECEPP. They directly took the IEEE-CS/ACM JTFSECEPP and classified them according to lifecycle phases. Besides, they did not give any justification as to why they did select only the four principles. Furthermore, the research was carried out by taking a single company. To enhance the validation of the research results, more cases should be considered. Generally, the work of Ammar *et al.* [6, 11] showed that further investigation should be carried out to analyze ethical concerns in SDLC phases in a more detailed manner.

Lurie and Shlomo [10] proposed an ethical framework that amalgamates software engineers' professional knowledge with their ethical responsibilities [10]. The main purpose of the research was to associate the software development lifecycles with the professional responsibilities of software engineers. The proposed work has both strong and weak points. Though the research demonstrates the possibilities of integrating both technical knowledge and the professional code of ethics, the proposed

work does not suggest basic ethical issues that should be answered at each level. For example, we can see the “yes” or “no” questions that are prepared for the RE phase. These include questions concerning requirements decision-makers, functional requirements, customer resistance, the existence of special negative consequences and contradictory issues, and requirements prioritization. By answering only these questions, it is difficult to address the main ethical concerns. In general, when we see their proposed framework, it relies on a set of “yes/no” ethical questions for each phase of development that may lead to subjective decision-making, and what is considered ethical can vary among individuals. The framework may not capture the full complexity of ethical decision-making, and stakeholders may interpret questions differently. In addition, the “yes/no” format of the ethical questions may oversimplify ethical considerations and lead to a checkbox mentality, where stakeholders simply go through the motions of answering questions without deeply reflecting on the ethical implications. This could result in a shallow consideration of ethics in the development process.

Van de Poel [129] suggested a framework to evaluate experimental technologies, more particularly those technologies that can pose ethical, societal, and safety risks. The proposed framework is intended to serve as a guide for decision-making when technologies are still being developed and their long-term effects are still unknown. One of the contributions of this research is that it addresses ethical responsibility in technological innovation, specifically on how to anticipate and prevent potential risks before widespread use. One of the strengths of Van de Poel's proposed work it emphasizes ethical responsibility, which is vital and aligns with RE's ethical decision-making. However, there are gaps in its direct application to RE ethics. While the framework does take into account ethical responsibility in software development, it does not explicitly cover the specific RE issues. In addition, it fails to provide tangible guidelines for RE practitioners on how to formally include ethical considerations in RE.

An ethics-aware software engineering approach was presented by Aydemir and Fabiano [48], which records, examines, and represents stakeholders' values about software artifacts and software engineering procedures. By examining the ethical issues with software assets and software engineering processes, their relevant value,

and in-danger entities like developers and users, the suggested effort seeks to assist stakeholders in software development initiatives. Furthermore, the approach is suggested as a plugin for all models of the development process. It also consists of five distinct stages: articulation, specification, implementation, validation, and verification. The approach is also suggested with four distinct enablers: transparency, ethics knowledge, conscious valuing, and awareness. The proposed method is stimulating since it presents fresh ideas for software development methodologies that consider ethics. Nonetheless, the covers a very wide issues. A broad aspects and every stage of the software development life cycle were covered. Target groups and development stages tailored to ethical issues should be taken into account in order to attain depth. Since the RE phase is the cornerstone of the others, it is very important to include a list of specific ethical issues along with suitable solutions.

Salehnia and Hassan [151] assessed ethical issues of software engineering and ethical aspects of software and system development. These scholars reviewed different codes of ethics including the ACM, Data Processing Management Association, Institute for Certification of Computing Professionals, IEEE, and IEEE-CS/ACM JTFSECEPP. The review shows that in one way or another, all of them are playing a role for computing professionals to diminish the advert impacts of computing systems and trying to safeguard the safety, health, and welfare of society. Moreover, they reviewed other scholars' papers from a technical point of view, social, and ethical point of view. From their review, they conclude that like technical knowledge of software development, recognizing more about the ethical impacts of software artifacts, in which it is to be used, is very important.

3.4 **Complementary Works on Ethics in RE**

Rashid *et al.* [152] assessed the ethics of software engineering in a digital world. The authors stated that software engineering exercises and practices should explicitly integrate ethical concerns. As stated by the authors, we are living in a globally growing digital world, in which various technologies are pervasively accessible. This enhances the 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 speeds up numerous unethical activities. The authors also explained this situation, i.e., the problem of a single technology having the ability to bring both good and bad, as a dual-use dilemma. To demonstrate the concept of the dual-use dilemma in the context of the digital world, the authors used three sample applications: Google Maps, chat rooms, peer-to-peer networks, and Facebook. They stated that the dual-use dilemma emerges because the same single technology can be used for harm or for good. As discussed, instead of a simple comparison of the benefits and drawbacks of a technology, software engineers should critically investigate the damaging ethical concerns of individual software technology. Moreover, they should search for alternative solutions, approaches, and/or methods, including examining the algorithmic design of those technologies, for the dual-use dilemma for the corresponding technology.

Thew and Sutcliffe [153] investigated three socio-political issues of requirements engineering, specifically values, motivations, and emotions. As for the elicited requirements techniques, researchers employed interviews, workshops, prototyping, and telephone conferences. To evaluate the proposed method, researchers used case studies and expert evaluations. The researchers proposed a method for analysing 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 about the RE 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 RE process. However, Thew and Sutcliffe's [153] research only focuses on values, which are relatively constant personal beliefs, and motivations, which are psychological concepts associated with personal characters that may be observed as stakeholders' long-term goals in RE. The professional ethical responsibilities of software engineers (technical persons) for RE were not discussed yet.

Whittle *et al.* [154] discovered the means of integrating human values into software engineering and proposed new methods of taking those values into account. To investigate, the researchers used three real-life projects. The researchers used an 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, similar to quality requirements of software systems, consideration should be given to human values like for instance, respect, empowerment, transparency, community responsibility, and others. These things are essential since they relate to the professional responsibility of software engineers during RE processes. Nevertheless, the research work of Whittle *et al.* [33] mainly focuses on human values rather than linking them to ethical issues that should be considered during RE.

Jones [32] identified three deep-rooted troubles of software requirements that desire solutions; these are toxic requirements (that have long-term unidentified requirements which have hazardous impact on direct and indirect stakeholders), incomplete requirements, and surplus requirements. Consequently, software engineers have a professional responsibility to avoid these problems [32]. The researcher places a strong emphasis on the responsibility of software engineers in requirements gathering and analysis to address these concerns.

Similarly, Cemiloglu *et al.* [155] investigated the ethical requirements for addictive technologies in the case of online gambling. Through their study, they identified three categories requirements that should be considered gambling addiction problems. These are 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 gives initial insights regarding issues that must take into account to proactively protect the existence of ethical associated with gampling technologies. Nevertheless, the research only focuses on one addictive technology, gambling. More work needs to be done on ethical guidelines to be followed during RE

Paech and Kurt [156] explored the need to understand users’ views of software. Users could not be discoursed like developers. While users talk regarding the external view of software, developers should have to compare features of the software, identify compliance with the software usability, and differentiate the constraints including its ethical issues. The researchers proposed an intermediary communication language (user view language) as a common ground for discussions among users and

developers. Their research findings emphasize the need for attention to identify ethical problems and to look at different aspects of software, particularly from users' points of view.

In summary, the reviewed related works investigated ethical problems from different viewpoints. In this chapter, we discussed some related works which are more related to the research. This research used different strategies including using different search keywords, “or” and “and” operators, and a pass reading approach for selecting suitable papers to be reviewed. Before anything else, we discussed the IEEE-CS/ACM JTFSECEPP to provide clarifications regarding the adapted codes of ethics for this research. Following that, those works that were carried out related to software engineering codes of ethics and software development are reviewed. In addition, some works that were done on technology and ethics are considered since they are essential to get the ethical implications of those technologies on RE. We categorized those works into three. The first one is the software engineering code of ethics. Secondly, those works that mainly deal with software development and software engineering codes of ethics are elaborated. Thirdly, we explained five reviewed papers that have relations with RE ethics in various ways. As we see from the above discussions, more work is required to address the consequence of software technologies at the initial stages, RE. Therefore, this research is initiated to investigate the ethical concerns of RE and propose a framework that helps software engineers to get a clear guideline concerning the consideration of ethical concerns in RE.

Chapter Four – Research Findings

4.1 Introduction

Before selecting the case companies, we conduct an initial screening using their websites and online platforms. This involves reviewing their annual reports and other related reports available on their website. These include reviewing accessible contact information, the number of employees they have, the area in which they are engaging, years of experience, and other relevant published information. After conducting this preliminary information, we tried to communicate with those companies via phone calls, in person, and through other online platforms. From a total pool of forty companies as population size, seven were purposively selected based on specific criteria such as the availability of required information they have like accessible contact and address information, their transparency/openness, and their active/ongoing engagements in software development. This selection process helps us to ensure the ability of the chosen companies to provide the rich and detailed information necessary for this research.

The research considers a diverse range of company sizes. The participating companies range from having less than ten employees in the development activities to large companies having more than 3000 employees in the development task. Besides, those companies have a variety of years of experience, except for one company (Company E) that has five years of experience; most of them are well-experienced. Four companies (Company A, B, C, and F) have more than sixteen years of experience, and two companies (Company D and G) have sixteen and ten years of experience, respectively. The oldest company, Company B, has twenty-nine years of experience in developing software products and services. The following table, Table 4.1 shows the companies' sizes and their years of experience.

Table 4.1 Software companies' Size and Year of Experience [12, p. 40]

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

Almost all companies are engaged in the provision of affordable IT-based solutions to business enterprises and customers in e-government, enterprise resource planning, human resource management, inventory and stock management, medical software, customer relationship management, mobile applications, etc. The companies are given alphabetical coded names for privacy. The number of participants involved in the research from the seven participating companies is thirty-five. Table 4.2 below displays some of the major areas in which each of the companies engaged in.

Table 4.2: Companies' Engagement

Company Code	Engagements
A	• Provides IT solutions and services in Ethiopia and abroad, with a focus on management information systems consultancy, software development, business process outsourcing, e-commerce, etc.
B	• Mainly, it provides enterprise resource planning system solutions for its clients • Its clients include star-rated hotels, supermarkets, and restaurants
C	• Software developing company for different clients, including hotels, supermarkets, and others
D	• Automating processes, administrative tasks, • Integrating technologies in business and academic institutions, etc.
E	• Software development company • Have a partnership with more than 1000 organizations
F	• Works on security, cybercrimes, and others through software systems development • Works on protection of society
G	• Providing IT-based solutions by developing software systems

Among the thirty-five participants, one is a Ph.D. holder, twenty-four are MSc holders, and ten of them are BSc holders in computer science, software engineering, and information systems. Table 4.3 underneath presents the details of the companies and participants.

Table 4.3: Participants' Demographic Information [12, p. 41]

Variables	Category	No. of Participants Involved	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%
	Ph.D.	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	14	40%

4.2 Data Collection Procedure

Before data collection, the participants were briefed on the aims of the research. They were informed about confidentiality, voluntariness, and their rights to withdraw at any time without facing any consequences. Written informed consent was prepared for each participant. No formal training was provided, but clear explanations were given to all the participants.

We used thirteen interviews and five FGD questions. These questions were validated by twelve computing professionals (ten Ph.D. and two Ph.D. candidates). According to their comments and feedback, we made revisions. The questions are designed to answer the research questionsidentify ethical concerns during RE. All the interviews were conducted face to face and they were audio recorded, with the average length of the interview being 45 minutes. We conducted two FGDs with four members each.

On average, each of the FGDs took an hour. Both FGDs were very open and interactive sessions. During the interview sessions, we used probing questions to discover relevant ethical concerns in a detailed manner. Moreover, before starting the interview and the focus group discussion questions, every participant agreed to and signed the consent form. Annexes II and III depict interview and FGD questions, and the consent form respectively.

4.3 Data Analysis

The thematic analysis is one of the qualitative analysis methods researchers use to organize and analyze qualitative data. It refers to repetitively reviewing and rereading the recorded data to identify codes, categories, and themes [157]. Different scholars recommend using a six-stage analysis of qualitative data to keep the needed rigour in the analysis process [87, 85, 157]. For this research to analyse the collected data, a thematic analysis is employed. Figure 4.1 below shows the processes involved during the analysis of stage of this research.

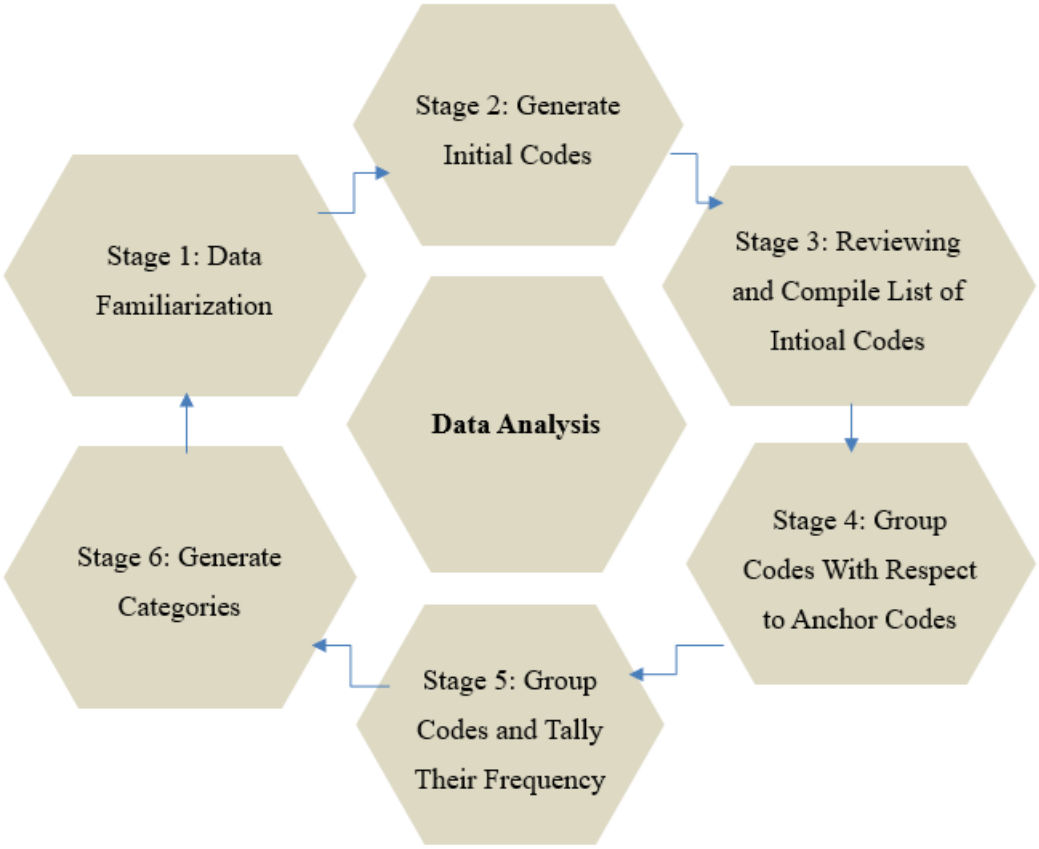

Figure 4.1: Steps in Data Analysis

Here, we discussed the major activities that occurred under each of the processes.

4.3.1 Stage 1: Data Familiarization

The data were organized to be suitable for analysis. Appropriate qualitative analysis techniques were selected and utilized to generate codes, categories, and themes of the research.

At the familiarization with data stage, we immersed ourselves in the data by reading them multiple times. The major processes carried out at this stage are the interview and FGDs, which were listened to and transcribed again and again.

The findings of the interview and focus group discussion questions are grouped according to the research questions. Then, holistic coding is employed to analyse collected data from participants. This method is used since it enables us to grasp the main concepts and patterns within the dataset. The overall concepts of the corpus are identified and coded with groups of data. anchor codes. We thoroughly read and re-

read the collected data, including interview transcripts and FGDs repeatedly to identify the important concepts of the findings. Then, we highlighted key concepts and identified anchor codes to large segments of the data.

Revisions were carried out on them and finally five anchor chodes, corresponding to each research questions were emerged from the entire collected data. These are Importance, Practice, Fundamental Ethical Concerns, Management-related problems, and Methods. These anchor codes are key codes that represent the most important themes that developed at the initial stages of the analysis, and they served as basic reference points to analyse the collected data further by applying initial, in vivo, and attribute coding strategies.

4.3.2 Stage 2: Generating Initial Codes

At this stage, the initial coding strategy is employed on the interview transcripts and FGDs. Each part of the collected data was analysed word by word, phrase by phrase, paragraph by paragraph, and section by section in detail. Every distinct section of the data was carefully reviewed. Following that keyword, and/or word phrases that seem very relevant were highlighted and identified. Then, meaningful labels for each of the data segment were given, and relevant statements were coded, and they are put under their respective anchor codes. In addition, by using in vivo coding, some important actual words of the respondents are taken as they are. Moreover, we use attribute coding to discuss the participating software companies and individual participants' demographic information.

4.3.3 Stages 3 and 4: Review and Compile Initial Codes

In this point, we reviewed, refined, and organized the labeled codes generated in stage 2 to ensure clarity, coherence, and relevance. Finally, at this stage of the coding process, the list of initial codes is compiled. All in all, three hundred forty codes from participants' interviews and FGDs were developed. In the fourth stage, these codes are grouped based on the anchor codes or the research questions.

4.3.4 Stage 5: Group Codes and Tally Their Frequency

During this stage, we organized codes into broader themes and counted their frequency. Revisions were made to make each code distinct and meaningful. Each code is depicted according to its frequency.

4.3.5 Stage 6: Generate Categories and Refine Themes

At this stage, further revisions encompassing refining the initial codes and anchor code revisions were made. The essence of each anchor code, the initial codes, were carefully reviewed. After lots of revision, we obtained one hundred seventy six codes, thirty four categories, and five themes.

In this research, codes, categories, and themes are used based on Saldaña's [87] explanation. In qualitative research, a code is usually a word or word phrase that effectively conveys concise, meaningful, and insightful attributes of a section of textual or visual data. In this research, codes refer to short labels given for specific segments of data/text that express a meaningful concept regarding the texts that they represent. Codes are used to represent small units of data segmentations. Categories are more general classifications used in qualitative research that combine related codes according to common characteristics. By observing patterns and relations among coded segments, categories help to organize the data. We used categories as a transitional phase between coding and the formation of themes. Similarly, themes characterized broad conceptual holistic interpretations of the collected data in line with the research questions. The anchor codes, which were identified at the beginning of the analysis (at stage 1), were reviewed carefully [31]. After multiple revisions, in the end, one hundred seventy six codes, thirty four categories, and five themes emerged.

In the following section, Section 4.4, we discussed the major findings obtained under each of the five themes. Sample codes and categories for each of the themes are also depicted and have been developed using the Xmind mapping tool.

4.4 The Research Findings

This section presents the details of the themes, categories, and codes.

4.4.1 Themes

a. Importance

The first theme, **importance**, deals with the importance of clearly considering ethical concerns during RE processes. Regarding this, all the participants replied that considering ethical concerns during the RE process is essential. Mainly, two categories have been identified under this theme, namely, helping to get quality software and avoiding ethical concern violations. Considering ethical concerns during RE processes helps to get quality software. This means that if we consider ethical concerns when doing RE, it is possible to ensure privacy and security issues, enhance software project success, refine artifacts, enhance ease of use, enable us to get in-depth information, and so on. Considering ethical concerns also helps to avoid ethical violations. This means that it helps us to elude deceptive requirements, protect confidentiality, prioritize social safety, find complete requirements, and so on. A few of the responses are presented as follows:

As one participant stated, “RE is the first phase of system development, and it is a foundation. As a result, if there is a problem in this phase, the problem will be reflected in the design, architecture, and all other phases. Besides, the problem which is not solved in this phase will have a high impact on the other phases” [12].

Likewise, the other participant mentioned that "Of course, considering ethical concerns has many benefits. In requirements-gathering activities, the more we go into detail, the more information we will obtain. Nevertheless, in practice, this is not exercised. Only functional requirements, such as the needs of our customers, have high focus and will be gathered in most situations.”

Similarly, the other participant expressed that “Considering ethical concerns helps to avoid requirements duplication and to give priority and attention to critical ethical issues according to organizational behavior, needs, and wants. In principle,

requirements should be complete, clear, unambiguous, consistent, and accurate. It helps us to respect social values, to empower customers, to protect confidentiality and privacy issues, to protect unauthorized access to information, to avoid unnecessary and unwanted requirements, to avoid deceptive requirements, and so on. However, as far as I know, there are no laws, rules, or regulations that 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 other thing. There should be legal procedures that enforce the inclusion of ethical concerns and the accountability of software developers" [12].

Under this theme, a total of three main categories and eighteen codes were identified from the qualitative data analysis. Figure 4.2, represents these categories and codes within this theme.

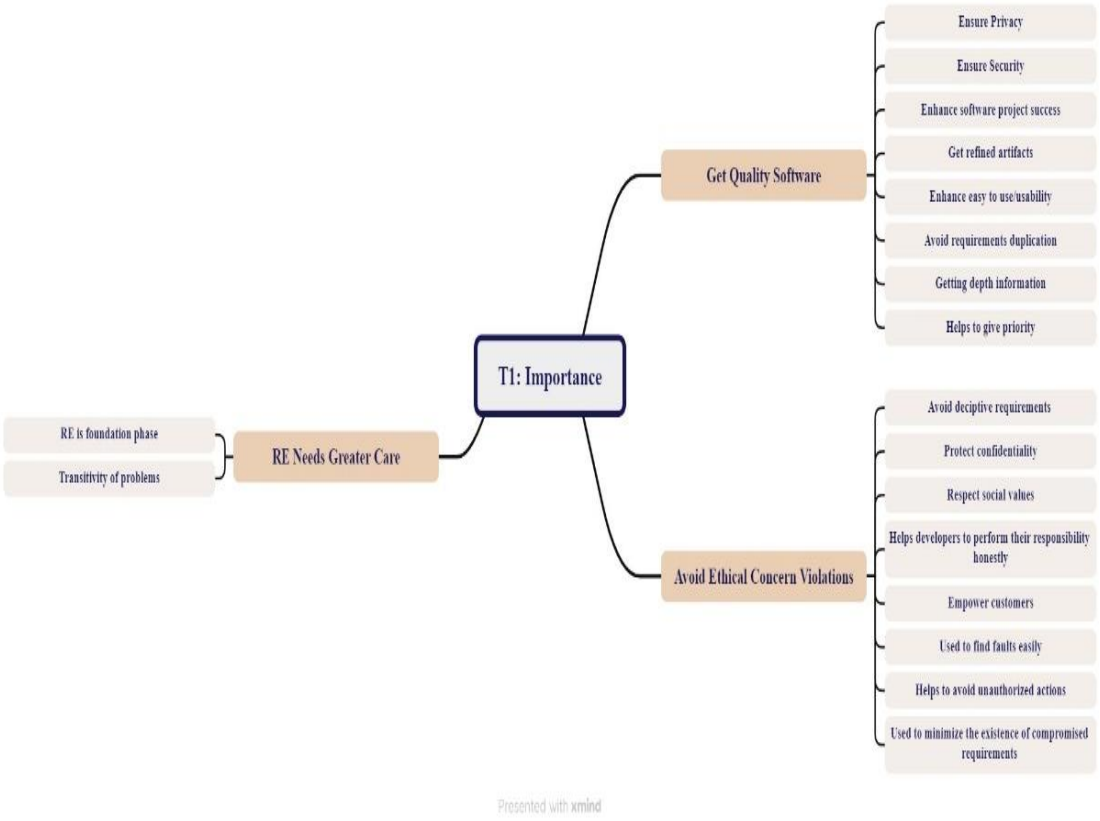

Figure 4.2: Importance – Codes and Categories

b. *Practices*

The second theme, **practices**, assesses the practices of software companies regarding the considerations of ethical concerns during RE processes. In this regard, the practice is assessed concerning industry practices, usage of professional code of conduct standards, and guideline utilization practices.

Regarding industry practices, except one, all the participating companies have a traditional way of doing RE processes. The single company, company "F", is trying to focus on the early identification of ethical concerns, especially security aspects of ethics, during RE processes. Some of the manners in which this company is practicing include ensuring security issues by giving great focus to them. Additionally, the company is using issue-tracking software (like requirements traceability matrices and document matrices) to filter requirements. Further, this company divides departments by specialization, such as business analyst, solution engineer, database designer, developer, architect, etc.

Regarding using professional code of conduct standards, respondents were asked which professional code of conduct standards related to ethical concerns for RE processes they use in their day-to-day activities. The respondents from six companies replied that they do not have any professional code of conduct standards that they follow in their day-to-day work on software development. One company, company "F", uses the IEEE Code of Ethics standard via customization with limited practice. Some of the employees of this company said, "We have not," and others stated only limited practices of IEEE. Similarly, almost all the six companies, except company F, stated that currently, they do not have any guidelines in place for considering ethical concerns throughout the software development project.

Moreover, respondents were asked to mention the methods they use to identify ethical concerns. They were asked what method they use to ensure that requirements don't endanger human safety or involve any deceit, incompleteness, inconsistency, improper requirements, ambiguity, or clarity issues. As most of the respondents said, they do not have any methods or checking mechanisms to perform such things [12].

They are using traditional identification techniques that are used for functional requirements like prototyping, discussions, and workshops.

In general, the findings of this theme refer to the absence of industry practices, professional code of conduct standards, and other guidelines within the company for integrating ethical aspects of software during RE processes. It also indicates that almost all companies do not have any identification methods or checking mechanisms for whether ethical concerns are considered and incorporated during the RE process or not. Generally, within this theme, a total of eight categories and twenty-six codes were developed. These categories and codes are shown below in Figure 4.3.

Figure 4.3: Practice – Codes and Categories

c. Fundamental Ethical Concerns

The third theme, **fundamental ethical concerns**, assesses critical ethical concerns that are obtained from the participants. A few of the responses are presented below.

One participant said, “Non-functional requirements are very vague to capture; they are costly to test." Users could not identify them correctly. Since there are these and other similar problems, taking non-functional requirements as requirements is very challenging.” Additionally, he stated that still functional requirements are unstable in the context of our nation. It is highly challenging to discuss non-functional requirements, and ethical issues with a system whose functional requirements are not satisfied. This is because it necessitates a large amount of funding and a supply of resources, including human and other resources.

According to another respondent's reply, the majority of ethical issues are private. That refers to how an individual behaves. As he said, “The challenge regarding RE is that in most cases, even though there are frameworks and guidelines, practically they are not used. The other challenge is a lack of knowledge on both the customer and business analyst or requirements engineer sides. The ability to identify requirements, hear them, and gather and appropriately document those requirements is a very challenging problem. Besides, from the customer side, they do not know how their business is done.” Similarly, the other respondent replies, “In most cases, I observed knowledge gap problems. Not knowing customers’ requests and needs.”

Another participant said, “The fundamental challenge during RE is the frustration of customers with using automated systems. Because they assumed that automation might cause them to lose their jobs, automation might cause overload (workload), and problems with their attitude concerning automation. Due to that, when they are asked to give requirements, they are not willing to give the real requirements, both functional and non-functional requirements. They hide every piece of information. In addition, unclear business processes, unclear and ambiguous requirements, unfulfilled infrastructure, and using outdated tools and technologies are the other ethical concerns that I observed.”

Similarly, another participant stated, “Customers could not be willing to give all the 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 due to some reasons, such as some of them being afraid to be jobless because of automation, some

of them losing unnecessary benefits associated with corruption, etc. I observed these fundamental ethical problems."

Under the theme of fundamental ethical concerns, six categories have been identified: knowledge gap problems (KGP), requirements identification problems (RIP), requirements quality-related problems (RQP),unwillingness to give requirements (U/DP), doing prohibited activities (PAP), and lack of legal ground or rules for accountability. Under these categories, about seventy five codes have been identified. Regarding the category of requirements identification problems, some of the codes include inadequate RE processes, insufficient requirements, "incomplete requirements, lack of attention, focusing only on functional requirements, not identifying them as requirements, not identifying the real needs and wants of customers, vague requirements, collecting irrelevant requirements, omitting requirements, and so on" [12].

Under the quality-related problems category, the codes include security concerns, privacy concerns, cyber-attacks, confidentiality issues, usability issues, etc. In the doing unpermitted things category, the codes include lack of honesty in job work, carelessness, adding evil requirements intentionally, plagiarism, and so forth [12].

Under the unwillingness to give requirements category, the codes include customers' reluctance to give clear, real, appropriate, and precise requirements, getting impure, and too complex requirements [12]. Regarding the knowledge gap category, lack of knowledge, communication gap, lack of experience, lack of knowledge in identifying and using the appropriate tools and methodologies, and others are included. Besides, these main categories, other categories were identified under this theme. Figure 4.4 below depicts the categories and sample codes within them.

Figure 4.4: Fundamental Ethical Concerns – Sample Codes and Categories

d. Management Related Problems

The theme, **management-related problems**, evaluates issues that tend to be related to software project management. Under this category, various issues are covered. To name a couple of them, a participant said, “Adapting to change can be challenging when it occurs. Not being proactive and not being able to adapt to that change is a bottleneck." The biggest challenge in the software development journey is the frequent change requests that customers raise, stated another participant. “Consider business process reengineering (BPR) as an example. Numerous changes were brought by BPR. We had our entire attention focused on developing BPR throughout that period. We are told to abandon the implementation after expending a great deal of time, energy, and resources on it and resume our regular workdays using the earlier method. We had spent a lot of resources at that point, which made it difficult because the system could not be used for a while.”

The theme has two categories: “frequent change requests and being reluctant to accept changes,” [12] with more than twenty-five codes. The following figure, Figure 4.5, depicts categories and sample codes of this theme.

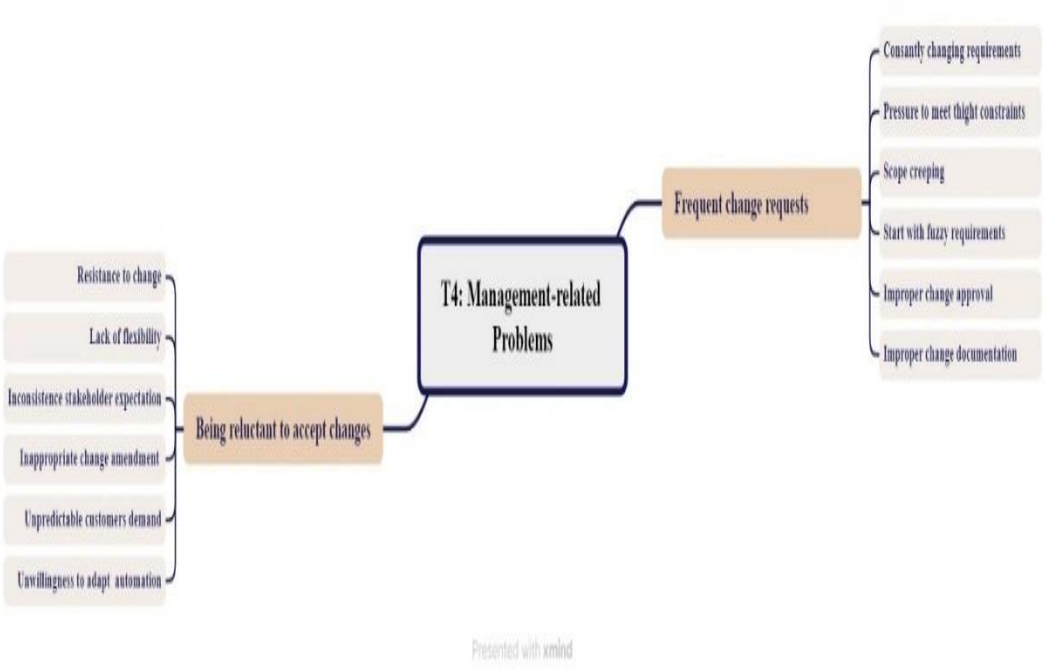

Figure 4.5: Management-Related Problem – Sample Codes and Categories

e. ***Methods***

The fifth theme, **Methods,** examines whether ethical issue identification techniques are present or not. Respondents were asked to mention the methods that are used to identify ethical concerns to know whether ethical concerns are considered or not. They were asked as to how they are determining whether requirements do not endanger human safety or involve any deceit, inconsistency, incompleteness, or inappropriate, imprecise, or ambiguous requirements. The majority of software industries, about six of them, lack defined procedures for identifying ethical problems during RE processes. One company, company F, use communication with customers, such as conversations with customers, meetings, pilot design, user acceptance testing, etc. [12]. The theme has about five key categories and twenty-eight codes. Figure 4.6 below portrays a sample of them.

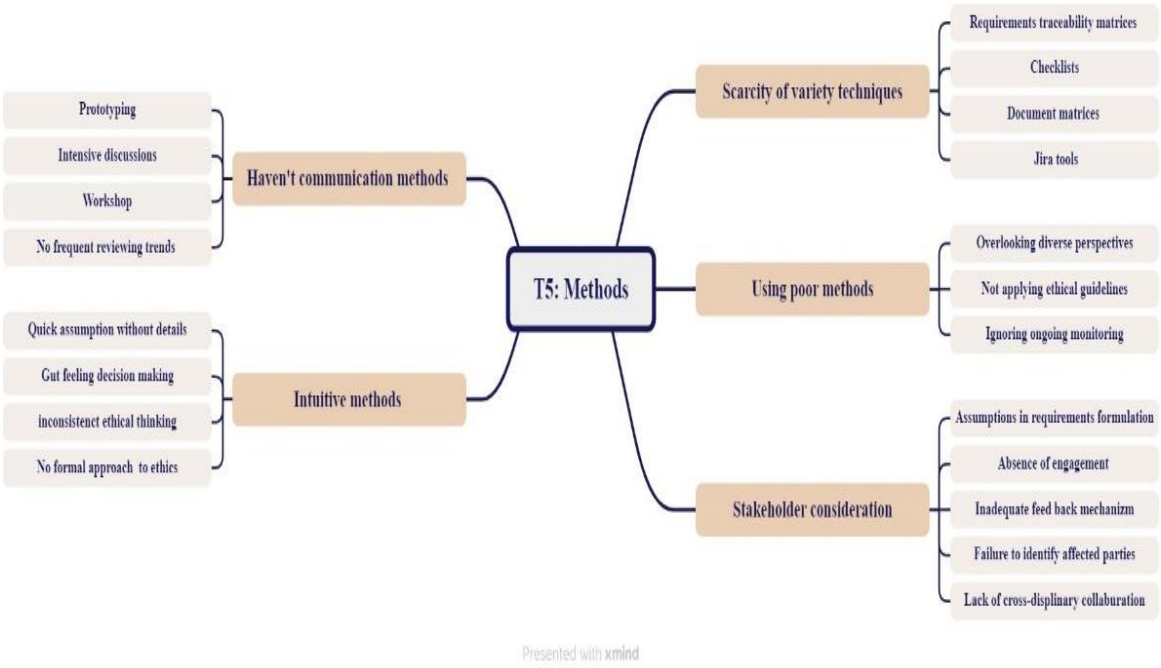

Figure 4.6: Methods Theme – Sample Codes and Categories

To conclude, in this chapter we discussed major research findings. Data were collected from the participants using interviews and FGD questions. This research investigates the industry practices in using methods to identify ethical concerns with three perspectives: industry practice, usages with respect to professional code of conduct standards, and application of any established ethical guidelines to bring ethical issues under consideration. According to the findings, those case companies do not have a well-organized way to consider ethical concerns during RE. Many of them fail to take into account ethical issues in their work throughout RE [72]. Moreover, this research examines the serious problems of RE that need critical attention. Following the analysis results, major themes, categories, and concerns are obtained.

Chapter Five – Framework for Software RE Ethics

5.1 Introduction

This chapter presents the proposed framework for RE ethics. Mainly, the proposed framework will be beneficial for software engineers. It can also be used as an aid for software developers, practitioners, and RE managers in the early identification of ethical concerns. Within the scope of this research, we use the term "software engineers" to represent those computing professionals who mainly work in RE processes, including requirements engineers, system analysts, and other computing professionals. The following subsections discuss the proposed framework and its components.

5.2 The Proposed Framework

After obtaining the analysis results of the research, we revised and screened the themes, categories, and codes based on the research questions. The main aim of this research is to investigate critical ethical issues of RE in real working environments of software industries and propose a solution via framework development. For this purpose, among the five themes, we selected the Fundamental Ethical Concern Theme and the first five categories within it, as discussed under subsection 4.1.1. Each of the categories encompasses codes. These codes represent the critical concerns that should be considered during RE processes. So, hereafter, the codes that were identified and screened during the analysis stage of the research are considered as concerns. Therefore, in this research context, codes and concerns are used interchangeably. Following a rigorous revision, the top twenty-four concerns to be considered during RE are screened. Other concerns can be represented by each of these selected concerns. Details of this are depicted in Annex IV. Table 4.1 displays these screen concerns to be considered in the proposed framework.

Table 5.1: The Top Fundamental Ethical Concerns to be Considered

No.	Categories	Concerns (Codes)
1	**Knowledge Gap**	a Unrealizing the benefit of automation b Lack of experience c Communication gap d Lack of knowledge to identify and utilize tools, and methods e Weak Requirement change management
2	**Requirements Identification**	a Incomplete requirements b Insufficient requirements engineering processes c Presence of ambiguous requirements d Requirements Omitting e Lack of attention for non-functional requirements f Not identifying "to be" requirements g Collecting irrelevant requirements h Lack of proper documentation
3	**Quality Related Problems**	a Security concerns b Privacy concerns c Usability concerns
4	**Unwillingness**	a Unwilling to give clear, real, appropriate, and precise requirements b Getting too complex requirements c Be reluctant to give real/precise requirements
5	**Doing forbidden action**	a Lack of honesty b Carelessness c Adding evil codes on source code d Plagiarism (duplications of works) e Denial of agreements

The above-selected main categories belong to the fundamental ethical concern theme. Among the six categories of fundamental ethical concern theme, we consider the first five, indicated in the Category column of Table 5.1: knowledge gap problems (KGP), requirements identification problems (RIP), requirements quality-related problems (RQP), unwillingness or disclination to give requirements (U/DP), and doing prohibited activities (PAP). To address these problems within each of the five

categories, the corresponding five components of the framework are identified. Every component of the proposed framework suggests pieces of advice for each of the corresponding concerns. Table 5.2 illustrates the identified components of the proposed framework inline with each category.

Table 5.2: Mapping Categories to the Framework Components

No.	Categories	Framework Components (Cn)	Meaning
1	Knowledge Gap Problems (KGP)	Ethics for Knowledge Gap (C1EKG)	Proposes lists of advice as solutions to resolve the identified knowledge gap-related problems
2	Requirements Identification Problems (RIP)	Ethics for Requirements Identification (C2ERI)	Suggests the possible solutions that can be considered to minimize the presence of those problems under RIP during RE
3	Quality-related Problems (RQP)	Ethics for Requirements Quality (C3ERQ)	Gives specific bits of advice that can be used to minimize the mentioned quality-related problems
4	Unwillingness or Disclination (U/D)	Ethics for Unwillingness or Disinclination (C4EU/D)	Recommends sets of advice as clauses that will be useful to get the willingness of customers
5	Doing Prohibited Activities (PAP)	Ethics for Prohibited Actions (C5EPA)	Shows the ways that help to diminish unethical behaviors and practices of software engineers

Following this, the proposed framework is systematically structured using the five components (C1EKG, C2ERI, C3ERQ, C4EU/D, and C5EPA), the major categories, and the screened concerns. Each component incorporates a set of clauses, which are

adapted from IEEE-CS/ACM JTFSECEPP, as resolving mechanisms to address the associated ethical concerns within them.

5.2.1 Notation for the Framework Building Blocks

For clarity, we provide notations for components, concerns, and clauses, which serve as the framework's foundational building blocks. We organize and discuss their labeling using the hierarchical notation system below.

Notation for Components

The components of the framework are labelled as Cn, where C represents the components and n represents the component's number (order of it), ranging from 1 to 5. For example, C1 refers to component 1 of the proposed framework. Hence, using this notation, the five components of the framework are labelled as:

- **Component 1 (C1):** "**Ethics for Knowledge Gap**" **– C1EKG**
- **Component 2 (C2):** "**Ethics for Requirements Identification**" **– C2ERI**
- **Component 3 (C3):** "**Ethics for Requirements Quality**" **– C3ERQ**
- **Component 4 (C4):** "**Ethics for Unwillingness or Disinclination" – C4EU/D**
- **Component 5 (C5):** "**Ethics for doing Prohibited Actions**" **– C5EPA**

Notation for Concerns

The concerns are represented as Con. n.x, where

- Con. stands for the identified requirements engineering ethical concern,
- n represents the component number in which that concern is found (indicates their placement order in the proposed framework components), and
- x shows the concern number within that component.

For instance, Con. 1.2 refers to the second concern of component 1.

Notation for Clauses

The advice provided for each concern under each of the components is organized as Cl n.x.a where:

- Cl denotes clause,
- n represents the framework component number,
- x refers to the concern number within that specific component,
- a indicates the advice number associated with the corresponding concern within that component.

For example:

- Cl 1.2.3 refers to:
  - Cl represents clause,
  - Component 1 of the proposed framework,
  - Concern 2 within component 1,
  - The third piece of advice provided for concern 2 within component 1.

Notation for Checklists

- Con. n.x – concern-related information, as mentioned above.
- SN
  - S refers to the strategies to ensure whether the identified concern is addressed or not (at least addressed to some extent or not).
  - N refers to the strategy number for the concerns to be checked.

In the following section, Section 5.3, each of the proposed framework building blocks is discussed. Also, the high-level proposed framework structure is depicted below in Figure 5.1.

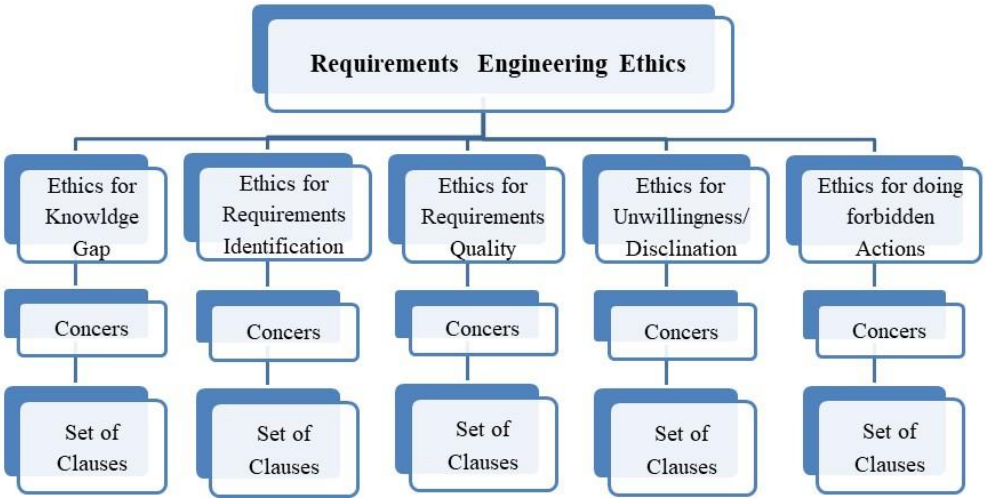

Figure 5.1: High–Level Proposed Framework Structure

5.3 Components of the Framework

5.3.1 Ethics for Knowledge Gap

Pertaining to this research, the knowledge gap indicates the knowledge gap problems that are observed in technical and nontechnical stakeholders of a software project.

Technical persons include software engineers, software developers, or anyone with an appropriate computing profession working and/or engaging in RE processes. In accordance with our research findings, some of these technical persons did not do the right things in their work, they become unwilling to participate professionally. Some of the problems that occurred related to technical persons include lack of experience, lack of identifying tools and methodologies, wrong utilization of those methodologies and tools, using outdated tools and technologies, and improper use of techniques.

Nontechnical persons include customers, managers, sponsors, and other stakeholders of a project who are expected to give requirements during RE processes. In the same way, problems related to nontechnical persons include frustration with automation due to different personal reasons, not realizing the benefits of automation or lack of interest in knowing about it, being reluctant to accept the change of workplace to be an automated environment purposely, and so on. Due to these and other related problems, customers do unethical activities such as not giving the real requirements of a system, becoming reluctant to any cooperation and communication, rejecting the final artifacts, etc.

In RE, knowledge gaps among technical and nontechnical stakeholders of a software project can result in unethical actions, whether intentionally or unintentionally. This lack of knowledge leads to misrepresentation, manipulation, and lack of knowledge in describing systems requirements, which can result in damage and bring hazardous impacts for users, organizations, society, and the environment. The ethical problems arising from knowledge gaps can be categorized under different domains with their impact.

Lack of knowledge in RE may cause misrepresentations of requirements. Nontechnical stakeholders, including business owners, product managers, and users, may define requirements (the needs, goals, and constraints of a system) incorrectly. Using those wrongly defined requirements, software engineers may prepare requirements specifications. This may lead to false expectations of the final product or service. In other cases, individuals or organizations knowingly manipulate requirements to their benefit, taking advantage of less informed stakeholders. For example, an executive in a company who does not have technical know how can misrepresent the functionality of a software product in project reports and thereby deceive investors and customers. A software company can also hide some technical limitatins at the intial stages of RE to secure the contract, and then disclose the necessity for additional expenditures. Such knods of deception result in a breach of trust, financial loss, and loss of reputation, and hence, it becomes a serious ethical problem in RE.

Moreover, a lack of knowledge may lead to collecting, defining, specifying, designing, sorting, and processing sensitive information in an insecure manner. For example, a software development team might neglect the proper encryption of user data due to a lack of knowledge in security best practices. Alternatively, an organization may be unintentionally gathering excessive amounts of users personal data without first gaining explicit consent from users. Such kinds of privacy breaches lead to identity theft, financial fraud, and abuse of user data.

Furthermore, the knowledge gap may lead to different challenges related to accountability and transparency among different stakeholders of a software project. Accountability and transparency are important in RE practices since they allow open

understanding among stakeholders concerning the goals, constraints, and risks of a system. Inadequate documentation, less traceability, and a lack of accountability for decision-making due to the knowledge gap problem create difficulties in seeking accountable persons for particular design decisions.In addition, there are many other cases in which a knowledge gap causes many unethical problems. This study finding shows that the knowledge gap is one of the major ethical problems during RE [12]. Filling the knowledge gap in RE is not only a technical necessity but also an ethical obligation towards being fair, accountable, transparent, and developing software that conforms to the best ethical standards. is, therefore, this component of the framework, Ethics for knowledge gap, discusses possible solutions, specific pieces of advices as set of clauses, for the selected knolwdge gap related problems. The classes that are mentioned under this category help software engineers resolve those problems in advance before proceeding to the other stages of software development. The following table, Table 5.3, shows sample clauses and corresponding concerns under this component of the framework.

Table 5.3: Ethics for Knowledge Gap – Sample Clauses

Concerns to be Addressed in C1EKG	**Clauses as Specific Advice for Consideration**	**IEEE-CS/ACM**
Con. 1.1 Not realizing the benefit of automation	1.1.1 Ensure your descriptions of automation benefits are accurate and based on evidence, taking responsibility for their reliability.	1.01
	1.1.2 Set clear, realistic expectations for what automation can achieve in a software project.	3.02
	1.1.3 Emphasize the human-centric benefits of automation, such as reducing tedious tasks and enhancing job satisfaction.	4.01
	1.1.4 Promote automation solutions that comply with industry standards and legal requirements.	6.06
	1.1.5 Continually improve your skills and knowledge to realize and communicate the full potential of automation in RE.	8.02

Concerns to be Addressed in C1EKG	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
Con. 1.2 Lack of Experience	1.2.1	Ensure that the lack of experience in RE does not compromise the safety, reliability, and quality of requirements. Seek expert guidance and thorough review to mitigate associated risks.	1.06
	1.2.2	Implement management practices that encourage continuous learning and knowledge sharing among team members. Create an environment where inexperienced team members can learn from seasoned professionals.	5.01
	1.2.3	Engage in continuous professional development to enhance skills in RE. Participate in workshops, seminars, and courses to stay updated with industry best practices.	6.03
	1.2.4	Consider best practices and support colleagues in professional development to mitigate the effects of immaturity in experience during RE.	7.02
	1.2.5	Take personal responsibility for your professional growth in the area of RE. Regularly seek feedback and take advantage of training opportunities to improve your expertise.	8.01
Con. 1.3 Communication Gap	1.3.1	Communicate effectively and efficiently with appropriate key stakeholders to identify and mitigate hazardous risks that impact the public.	1.04
	1.3.2	Respect and appreciate the diversity of viewpoints and experiences within the development community and cooperate to produce solutions that are both equitable and inclusive.	2.02
	1.3.3	To dig out ethical concerns early, let users speak about their needs freely in their own words. Try	3.02

Concerns to be Addressed in C1EKG	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
		to understand users better.	
	1.3.4	To provide impartial, unambiguous interpretations of the requirements, communicate objectively.	4.03
	1.3.5	Ensure that ethical standards are clearly communicated and expectations are precisely mentioned for members involved in RE.	5.02
Con. 1.4 Lack of knowledge in identifying and utilizing tools, methodologies	1.4.1	Learning is an endless process. So, upgrade your knowledge of software engineering methods, tools, techniques, principles, concepts, and standards for the development of software through formal and informal education and conversation with colleagues. This encourages to address and prevent issues of public concern.	1.05 and 6.03
	1.4.2	Ensure that software engineers who are working in RE processes are well qualified for their work by an appropriate combination of education, training, and experience.	3.04
	1.4.3	Software engineers should devote themselves to supporting colleagues in RE professional development, including adhering to ethical principles and assisting in technical weaknesses and nontechnical skill developments.	7.02
	1.4.4	Integrate your professional and ethical knowledge and work by following appropriate standards and principles.	8.01
	1.4.5	Demonstrate a commitment to ongoing professional development.	8.02
	1.5.1	Approve requirements only if they have a well-	

Concerns to be Addressed in C1EKG	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
Con. 1.5 Weak requirements change management		founded belief that it is safe, harmless for society, meets specifications, and does not diminish the quality of life and privacy or harm the environment. The ultimate effect of the work should be for the public good.	1.03
	1.5.2	Maintain professional objectivity concerning any change requests or other related tasks you are asked to evaluate.	4.03
	1.5.3	Promoting an ethical approach to the management of requirements ensures the inclusion of holistic needs that enhance ethical practices in RE by respecting the interests and values of stakeholders.	5.04
	1.5.4	During requirements change management, software engineers should not promote their interests at the expense of the profession, clients, or employer.	6.05
	1.5.5	Review change requests from your colleagues, clients, or other stakeholders objectively, truthfully, and properly documented.	7.04

As indicated above in Table 5.3, under C1EKG**,** five top concerns, labeled from Con. 1 to Con. 5, are selected. For each of these concerns, pieces of advice are suggested as some samples shown above in Table 5.3. Other concerns within C1EKG were grouped under these five concerns based on their resemblance in characteristics. Some of them are indicated in Annex IV.

Con. 1.1 has about fifteen clauses that were adapted from principles 1, 2, 3, 4, 5, 6, and 8 of IEEE-CS/ACM JTFSECEPP. The specific adapted clauses in these principles are 1.01, 1.02, 2.01, 2.02, 2.04, 2.06, 2.07, 3.01, 3.02, 4.01, 4.03,

5.03, 6.01, and 8.02. The following table, Table 5.4, shows the total number of clauses within each of the concerns and the corresponding clauses from which they were adapted.

Table 5.4: Ethics for Knowledge Gap – Concerns and Clauses

Concerns in C1EKG	**Number of Clauses**	**Adapted From (IEEE – CS/ACM …)**	
		Principles	**Clauses**
Con. 1.1 Not realizing the benefit of automation	15	1, 2, 3, 4, 5, 6, and 8	1.01, 1.02, 2.01, 2.02, 2.04, 2.06, 2.07, 3.01, 3.02, 4.01, 4.03, 5.03, 6.01, and 8.02
Con. 1.2 Lack of Experience	11	1, 2, 3, 5, 6, 7, and 8	1.06, 2.01, 3.01, 3.05, 5.01, 5.06, 6.03, 7.02, and 8.01
Con. 1.3 Communication Gap	18	1, 2, 3, 5, 6, 7, and 8	1.01, 1.04, 2.02, 2.06, 2.07, 3.02, 3.06, 4.03, 5.02, 5.04, 6.02, 6.05, 7.01, 7.05, 8.01, 8.02, and 8.06
Con. 1.4 Lack of knowledge in identifying and utilizing tools, methodologies	11	1, 3, 4, 7, and 8	1.05, 1.06, 3.04, 4,01, 7.02, 7.04, 7.08, 8.01, and 8.02
Con. 1.5 Weak requirements change management	10	1, 3, 4, 5, 6, 7, and 8	1.03, 3.05, 4.03, 4.05, 5.01, 5.02, 5.04, 6.05, 7.04, and 8.01

As indicated above in Table 5.4 for C1EKG, among eighty clauses of IEEE-ACM, about thirty six of them are adapted with respect to the concerns mentioned under C1EKG.

5.3.2 Ethics for Requirements Identification

As of this research finding, there are many requirements identification problems occur during RE processes which include, incomplete requirements, insufficient RE process, presence of ambiguous requirements, requirements omission, concentrating just on the functional requirements, failure to to identify the real “to be” necessities, collecting irrelevant requirements, and lack of proper documentation.

In RE errors in collecting, defining, analysing, and documenting requirements may result in ethical problems that negatively affect users, stakeholders, society, and environments. Incomplete requirements and missing key requirements may result in software failures, safety risks, economic crises, and ineffective solutions that waste time and resources while eroding trust.

Similarly, focusing exclusively on functional requirements or omitting key ones may bring disastrous system failures with potentially hazardous consequences. Ineffective RE processes not only lead to software that fails to meet the requirements of users but also causes cope creeping, increasing costs and schedule deadlines. Additionally, inadequate documentation renders development, maintenance, and upgrading problems with increased risk of misconfigurations, security vulnerabilities, and system crashes.

Every one of these requirements and identification problems, which were understood in the analysis stage of the research, has direct ethical implications. These problems can be considered unethical because they have many direct and indirect consequences, including violating ethical responsibilities, leading to conflict of interest, misleading stakeholders, lack of accountability and transparency, obstacles to information sharing, loss of trust, and so on.

Software engineers have a professional duty to make sure that RE processes are conducted in a responsible, accountable, and transparent way [9]. By resolving these problems in advance, software engineers not only enhance the quality of systems but also maintain their professional responsibility so that software will benefit users instead of harming them.

Therefore, this component of the framework is aimed to address the identified and selected requirements identification related problems. In other words, ethics for requirements identification (C2ERI) deals with the possible solutions that a software engineer can take into consideration to minimize the presence of each of the above-listed problems. For each of the problem domains, more than one clause is indicated. By looking at these clauses, a software engineer can take remedial action to reduce the existence of the problems mentioned above.

Under C2ERI, the topmost eight concerns are considered. To address each of the eight, lists of bits of advice are given. Each of the advices recommended suggestions to tackle the corresponding concerns. Table 5.5 presents sample clauses for each of the eight concerns.

Table 5.5: Ethics for Requirements Identification – Sample Clauses

Concerns to be Addressed in C2ERI	**Clauses as Specific Advice for Consideration**	**IEEE-CS/ACM**
Con. 2.1 Incomplete requirements	2.1.1 Ensure that requirements gather holistic needs to serve the best interests of clients and employees by delivering software systems that meet the objectives and needs of their stakeholders.	1.02
	2.1.2 Ensure significant issues are disclosed. Incomplete requirements can cause significant issues that must be disclosed.	2.03
	2.1.3 Identify, define, and address requirements issues. Incomplete requirements can lead to unaddressed ethical and other related issues.	3.03
	2.1.4 Disclosed conflicts of interest to all concerned parties. Incomplete requirements can bring potential conflicts of interest.	4.05
	2.1.5 Helping colleagues to understand the importance of complete requirements is part of their development.	7.03

Concerns to be Addressed in C2ERI	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
Con. 2.2 Insufficient Requirements Engineering Processes	2.2.1	By any means, the final requirements engineering artefacts and deliverables do not take precedence over the rights, safeties, and interests of clients, employers, and societies.	1.02
	2.2.2	Create a conducive working environment and ensure good leadership for software projects throughout RE processes.	5.01
	2.2.3	Ensure that realistic work is done on requirements engineering processes, including careful consideration of ethical concerns.	5.05
	2.2.4	Help to develop an ethical organizational environment. Inadequate RE processes can lead to unethical practices or outcomes.	6.01
	2.2.5	Improve your knowledge of software engineering code of ethics standards, their interpretation, and their application of them during RE.	8.06
Con. 2.3 Presence of ambiguous requirements	2.3.1	Ensure all requirements are unambiguous to avoid misinterpretations that could lead to unsafe software affecting the public.	1.01
	2.3.2	Clearly communicate any ambiguities found in requirements to clients and employers, seeking clarification and ensuring mutual understanding.	2.01
	2.3.3	Write requirements clearly and precisely, and avoid vague or subjective language usage. Prioritize and validate requirements to avoid redundancy and ambiguity.	3.01
	2.3.4	Ensure clarity and precision in requirements, perform rigorous reviews, validate requirements with stakeholders, and get approval of them.	3.08

Concerns to be Addressed in C2ERI	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
	2.3.5	Make judgments about ambiguous requirements with consideration for their potential impact on users and other stakeholders, ensuring decisions support human values and needs.	4.01
Con. 2.4 Requirements Omitting	2.4.1	Employ rigorous requirement elicitation and analysis methods to ensure that no critical requirements are omitted.	3.05
	2.4.2	Evaluate requirements comprehensively, considering all user needs and human values to ensure no essential requirements are overlooked.	4.01
	2.4.3	Address and correct any instances where requirements may have been omitted, recognizing the ethical implications of such omissions.	6.11
	2.4.4	Mentor, coach, and support colleagues in improving their requirement elicitation and analysis skills to prevent omissions.	7.02
	2.4.5	Continuously improve understanding of the domain and context to ensure all relevant requirements are captured and included.	8.04
Con. 2.5 Lack of attention for non-functional requirements	2.5.1	Be rational to keep social benefits from requirements.	2.07
	2.5.2	Facilitate early detection of ethical concerns by identifying, defining, and addressing ethical issues related to requirements on software development projects.	3.03
	2.5.3	Consider the influence of non-functional requirements on user and systems' usability, ensuring they align with people's safety.	4.01
	2.5.4	Push and promote to consider ethical concerns early,	5.01

Concerns to be Addressed in C2ERI	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
		in the requirements engineering process, to reduce later reworking.	
	2.5.5	Mentor and support colleagues in understanding and implementing non-functional requirements effectively.	7.02
Con. 2.6 Not identifying the real “to be” requirements	2.5.1	Confirm that the "to be" requirements are comprehensive and precise, confirming they lead to effective, safe, and useful RE artifacts and deliverables.	1.03
	2.5.2	Offer extensive information on the identified requirements, spotlighting any holes among the existing state and the required future state, and discuss possible dangers of not addressing these gaps.	2.03
	2.5.3	Ensure that requirements specifications have good documentation, satisfy users’ “to be” requirements, and have the appropriate approval.	3.08
	2.5.4	Communicate best practices and lessons you have learned about the identification of real "to be" requirements in the professional community to improve overall industry standards.	6.02
	2.5.5	Carry out in-depth reviews of requirements documents prepared by colleagues, providing constructive feedback to ensure the real "to be" requirements are accurately identified and documented.	7.04
	2.7.1	Ensure that the requirements-collecting process is carried out with the public interest in mind, focusing on collecting only those requirements that are very	1.02

Concerns to be Addressed in C2ERI	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
Con. 2.7 Collecting Irrelevant and Inappropriate Requirements		useful for the intended purpose of software products, services, and artifacts.	
	2.7.2	Approve requirements only if they have a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish the quality of life, diminish or harm the environment.	4.03
	2.7.3	Take responsibility for detecting, critically reviewing, correcting, and reporting errors on requirements and related activities and documentation on which you work.	6.08
	2.7.4	Encourage colleagues to use software engineering codes of ethics, including this proposed framework guideline, to refine requirements and eliminate irrelevant and inappropriate requirements that have unforeseen impacts and dangers.	7.01
	2.7.5	Improve your ability to create safe, reliable, relevant, and useful requirements for software development projects.	8.02
Con. 2.8 Lack of proper documentation	2.8.1	During RE, disclose any potential danger associated with requirements documentation or related documents.	1.04
	2.8.2	Ensure that specifications for software requirements have been well documented, satisfy the users' requirements, and have the appropriate approvals.	3.08
	2.8.3	Maintain professional objectivity concerning any requirements documentation you are requested to evaluate.	4.03
	2.8.4	Take responsibility for detecting, correcting, and reporting errors in requirements and associated	6.08

Concerns to be Addressed in C2ERI	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
		documents on which you work.	
	2.8.5	Improve your ability to produce accurate, informative, and well-written requirements documentation.	8.03

As shown above in Table 5.5, within C2ERQ**,** eight critical concerns, Con. 1 to Con. 8, are chosen. For each of these concerns, lists of advice are suggested to minimize their existence during RE. There are a total of one hundred-one clauses stated within C2ERQ**.** Con. 2.1 holds eighteen, Con. 2.2 sixteen, Con. 2.3 ten, Con. 2.4 eleven, Con. 2.5 fifteen, Con. 2.6 twelve, Con. 2.7 ten, and Con. 2.8 nine clauses. Table 5.6 below shows a summary of concerns, the number of clauses in each concern, and the source from which each was adapted.

Table 5.6: Ethics for Requirements Identification – Concerns and Clauses

Concerns in C2ERI	Number of Clauses	Adapted From (IEEE – CS/ACM …)	
		Principles	Clauses
Con. 2.1 Incomplete requirements	18	1, 2, 3, 4, 5, 6, 7, and 8	1.01, 1.02, 1.03, 1.05, 2.01, 2.03, 3.01, 3.02, 3.03, 4.01, 4.03, 4.05, 5.01, 6.08, 7.01, 7.03, 8.05, 8.09
Con. 2.2 Insufficient requirements engineering processes	16	1, 2, 3, 4, 5, 6, and 8	1.01, 1.02, 2.09, 3.02, 4.03, 5.01, 5.02, 5.05, 6.01, 6.08, 6.05, 8.04, 8.05, 8.06
Con. 2.3 Presence of ambiguous requirements	10	1, 2, 3, and 4	1.01, 2.01, 2.04, 2.05, 3.01, 3.05, 3.08, 4.01, 4.03
Con. 2.4 Requirements	11	1, 3, 4,6, 7, and 8	1.02, 1.03, 2.03, 2.05, 3.01, 3.05, 4.01, 6.02, 6.11, 7.02, 8.04

Omitting			
Con. 2.5 Lack of attention for non-functional requirements	15	1, 2, 3, 4, 5, 6, 7, and 8	1.03, 2.01, 2.03, 2.07, 3.01, 3.03, 3.05, 4.01, 4.02, 5.01, 6.11, 7.02, 7.04, 8.02, 8.04
Con. 2.6 Not identifying the real "to be" requirements	12	1, 2, 3, 4, 5, 6, 7, and 8	1.02, 1.03, 2.03, 3.01, 3.05, 3.08, 4.01, 5.02, 6.02, 7.04, 8.02, 8.04
Con. 2.7 Collecting Irrelevant requirements	10	1, 2, 3, 4, 5, 6, 7, and 8	1.02, 2.01, 3.01, 4.03, 5.05, 6.08, 6.11, 7.01, 7.02, 8.02
Con. 2.8 Lack of proper documentation	9	1, 2, 3, 4, 5, 6, 7, and 8	1.04, 2.04, 3.08, 3.11, 4.03, 5.03, 6.08, 7.04, 8.03

As depicted in Table 5.6 above, the clauses of C2ERQ are adapted from all the eight principles of IEEE-CS/ACM JTFSECEPP.

5.3.3 Ethics for Requirements Quality

During the research journey, many issues associated with quality are raised. From the research analysis, we found different types of quality concerns as problems of RE. Some of them are security issues, privacy issues, cyber attacks, confidentiality problems, usability problems, and so on [12]. "Ethics for Requirements Quality" can guide software engineers to tackle quality-related problems. Among all the problems identified under quality related problems, the first three, security, privacy, and usability concerns, are selected to be considered. Each of them consists of related serious problems that should be considered during RE. To address these issues, thirty-two clauses are developed via adapting from IEEE-CS/ACM JTFSECEPP clauses. Con. 3.1 consists of thirteen clauses as specific advice to reduce the occurrences of security problems of requirements. Similarly Con. 3.2 and Con. 3.3 incorporates thirteen and twelve clauses, respectively. Sample clauses and a summary of the number of clauses and concers under C3ERQ are indicated in Tables 5.7 and 5.8, respectively.

Table 5.7: Ethics for Requirements Quality – Sample Clauses

Concerns to be Addressed in C3ERQ	Clauses as Specific Advice for Consideration	IEEE-CS/ACM …
Con. 3.1 Security concerns	3.1.1 During RE processes, promote transparency and honesty in all professional interactions and disclose any conflicts of interest that may arise.	1.02
	3.1.2 Assure customers that their data will be protected and used responsibly in RE processes.	2.05
	3.1.3 Ensure that the privacy of the user and confidentiality of their data are respected and maintained throughout RE processes.	3.03
	3.1.4 According to the policies and procedures of an intended company, identify security requirements.	5.03
	3.1.5 Assist colleagues in being fully aware of current standard work practices, including policies and procedures for protecting passwords, files, and other confidential information, and security measures in general.	7.06
Con. 3.2 Privacy concerns	3.2.1 Every precaution must be taken to protect the privacy of clients, stakeholders, and the public.	1.04
	3.2.2 Keep private any confidential information gained during RE processes, where such confidentiality is consistent with the public interest and consistent with the law.	2.05
	3.2.3 Respect and protect the privacy of users	

Concerns to be Addressed in C3ERQ	Clauses as Specific Advice for Consideration		IEEE-CS/ACM …
		by ensuring the confidentiality and security of their data throughout RE processes.	3.01
	3.2.4	Constructively decide to implement appropriate ethical principles that respect stakeholder confidentiality and privacy.	4.01
	3.2.5	Uphold integrity and professionalism in all aspects of RE, including ensuring transparency about data handling.	6.06
Con. 3.3 Usability concerns	3.3.1	Ensure the requirements are easy to understand and effective in meeting users' needs.	1.03
	3.3.2	Identify, document, collect evidence, and promptly report any usability issues to the client or the employer if it is believed that a project may fail, become expensively costly, infringe on intellectual property laws, or pose other significant problems.	2.06
	3.3.3	Ensure that usability is part of the quality parameter of software requirements.	3.01
	3.3.4	Balancing the interests of stakeholders with the public good includes ensuring the reliability of requirements to prevent harm or inconvenience to users.	4.02
	3.3.5	Software engineers shall collaborate with colleagues to identify and address usability problems during RE via fostering a supporting environment.	7.05

Table 5.8: Ethics for Requirements Quality – Concerns and Clauses

Concerns in C3ERQ	Number of Clauses	Adapted From (IEEE – CS/ACM …)	
		Principles	Clauses
Con. 3.1 Security concerns	13	1, 2, 3, 5, 7, and 8	1.02, 1.03, 2.05, 3.01, 3.03, 3.04, 3.12, 3.13, 3.14, 5.03, 7.06, and 8.02
Con. 3.2 Privacy concerns	13	1, 2, 3, 4, and 6	1.02, 1.03, 1.04, 2.03, 2.05, 2.08, 3.01, 3.02, 3.01, 3.02, 3.06, 3.12, 4.01, 4.03, and 6.06
Con. 3.3 Usability concerns	12	1, 2, 3, 4, 6, 7, and 8	1.01, 1.03, 2.06, 3.01, 3.04, 3.07, 4.02, 6.04, 7.05, and 8.02

5.3.4 Ethics for Unwillingness or Disinclination

This research finding demonstrated that most customers and clients are not willing to give real and accurate requirements intentionally [12]. Due to this, the collected requirements will become more complex. Besides, deliberately they become reluctant to give the real requirements. Hence, this component of the proposed framework, "Ethics for Unwillingness or Disinclination," gives guidelines to get solutions for such kind of disinclination from the customers' side. As part of this component, C4EU/D, the three most serious major concerns are selected to be considered during RE. For each of the concerns, possible suggestions are provided. Con. 4.1 comprises fourteen Con. 4.2 eleven, and Con. 4.3 seven clauses as specific bits of advice to lessen their likelihood in the course of RE. Table 5.9 below presents sample clauses from C4EU/D.

Table 5.9: Ethics for Unwillingness – Sample Clauses

Concerns to be Addressed in C4EU/D	Clauses as Specific Advice for Consideration	IEEE-CS/ACM
Con. 4.1 Unwillingness to give clear, real, appropriate , and precise requiremen t	4.1.1 Retain honesty and refrain from deceptive practices that could mislead customers.	2.05
	4.1.2 Participation by individuals must be voluntary.	2.03
	4.1.3 During requirements engineering processes, every potential participant must be adequately informed of the aims of the project, methods used, sources of data, any possible conflicts of interest, the anticipated benefits, their rights and duties of them.	4.05
	4.1.4 Encourage public and clients' knowledge of requirements significance on quality, safety, usability, and reliability of software products and services.	6.02
	4.1.5 Software engineers should work in collaboration fairly with colleagues, customers, and clients, including accepting and providing clear and appropriate information during RE without distorting facts.	7.07 and 7.08
Con. 4.2 Getting too complex requirements	4.2.1 Inform well customers regarding the dangers associated with too complex requirements, focusing on the possible effects on the general public.	1.04
	4.2.2 Put in place management practices that place considerable importance on precise requirements, clarity, and communication to lower project risks.	5.01
	4.2.3 All stakeholders of proposed software projects	6.01

Concerns to be Addressed in C4EU/D	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
		should have ethical obligations.	
	4.2.4	Make sure that cooperative efforts to define requirements are acknowledged and recognized.	7.03
	4.2.5	Improve your capacity to organize and deconstruct complicated requirements in order to improve the results of RE documents, artifacts, and deliverables.	8.02
Con. 4.3 Be reluctant to change intentionally	4.3.1	Even if customers become reluctant, advocate for necessary changes to requirements that ensure the final product is safe and meets quality standards.	1.03
	4.3.2	Uphold confidentiality while promoting necessary reforms and establishing a balance between the client's interests and concerns about public safety and quality.	2.05
	4.3.3	Learn and use the best practices and techniques for handling and negotiating modifications with reluctant clients constantly.	3.06
	4.3.4	Any conflicts of interest or hazards associated with the unwillingness to modify requirements should be openly communicated.	4.05
	4.3.5	Work together with colleagues to address RE challenges, including customers' resistance to automation change amendments.	7.08

All in all, in C4EU/D, there are about thirty-two clauses that were adapted from the eight principles and clauses of IEEE-CS/ACM JTFSECEPP. Table 5.10 provides a summary of the number of clauses and concers under C4EU/D.

Table 5.10: Ethics for Unwillingness – Concerns and Clauses

Concerns in C4EU/D	Number of Clauses	Adapted From (IEEE – CS/ACM …)	
		Principles	Clauses
Con. 4.1 Unwillingness	14	1, 2, 3, 4, 6, 7, and 8	1.03, 1.04, 2.01, 2.03, 2.05, 3.01, 3.02, 4.01, 4.02, 4.05, 4.06, 6.02, 7.07, 7.08, and 8.07
Con. 4.2 Privacy concerns	11	1, 2, 3, 4, 5, 6, 7, and 8	1.04, 2.01, 3.01, 3.02, 4.01, 5.01, 6.01, 7.03, and 8.02
Con. 4.3 Usability concerns	7	1, 2, 3, 4, 5 and 7	1.03, 2.05, 3.06, 4.05, 5.01, 5.11, and 7.08,

5.3.5 Ethics for Unpermitted/Prohibited Activities

According to this finding, many unpermitted concerns are done through software engineers. For example, software engineers may perform their responsibilities carelessly, do their work dishonestly, and add evil or deceptive requirements for some purpose deliberately [12]. Besides, instead of doing the intended task, they simply duplicate the work of others without any permission. Ethics for Unpermitted Activities refers to the methods that help software engineers to avoid such kinds of unethical behavior and activities among them. As advice for concerns within C5EPA, there are about sixty-six clauses. Samples of these clauses are indicated in Table 5.11 below.

Table 5. 11: Ethics for Prohibited Activities – Sample Clauses

Concerns to be Addressed in C5EPA	Clauses as Specific Advice for Consideration	IEEE-CS/ACM
	5.1.1 Try to ensure fair inclusion of all stakeholders' benefits and safety of the society.	1.06
	5.1.2 Be honest and do your professional	2.08

Concerns to be Addressed in C5EPA	Clauses as Specific Advice for Consideration	IEEE-CS/ACM
	responsibility for your employer.	
Con. 5.1 Lack of Honesty	5.1.3 Keep professional neutrality when assessing any requirements specifications or related documents that you are requested to look over.	4.03
	5.1.4 Don't further your personal interests at the expense of your employer, clients, or profession.	6.05
	5.1.5 Not influence others to undertake any action that involves a breach of software engineering codes of ethics.	8.08
Con. 5.2 Carelessness	5.2.1 Try to avoid any harm due to your negligence. Being careful with handling confidential information is essential to maintain trust and avoid data breaches.	2.05 and 2.09
	5.2.2 Careful consideration of technical judgments of requirements confirms they are aligned with human values and ethical standards.	4.01
	5.2.3 Not unjustly prevent software engineers from taking a position for which that person is suitably qualified.	5.08
	5.2.4 Act in the best interest of software engineering profession, clients, employers, and the public.	6.05
	5.2.5 Strive for quality in your work and continuous improvements of your knowledge and ability. This helps software engineers stay updated on best practices and techniques and reducing the likelihood of carelessness.	8.01 and 8.02
	5.3.1 Be fair and avoid deceptions in all requirements, particularly public ones,	1.06

Concerns to be Addressed in C5EPA	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
Con. 5.3 Adding evil requirements for some purpose /Deceptive Requirements		concerning requirements specifications or related documents, methods, and tools.	
	5.3.2	Promptly reporting deceptive requirements prevents potential legal and ethical issues and aligns with professional responsibility.	2.06 and 2.07
	5.3.3	Avoid deceptive practices driven by personal benefits among software engineers and be transparent concerning those conflict of interests.	4.05
	5.3.4	Be accurate in stating the characteristics of requirements on which they work, avoiding not only false claims but also claims that might reasonably be supposed to be speculative, vacuous, deceptive, misleading, or doubtful.	6.07
	5.3.5	Approach colleagues and assist them to well understand the professional codes of ethics well to prevent the occurrences of harmful and deceptive requirements which have hazardous impact on the clients, employers, colleagues themselves, as well as society and the environment too.	7.01 and 7.02
Con. 5.4 Plagiarism (redundancy, duplication of works)	5.4.1	Approve requirements only if they have a well-founded belief that it is original and free from plagiarism. This helps to maintain the quality and reliability of requirements and to ensure they are safe, meets specifications, passes appropriate validations, don’t diminish quality of life, diminish privacy, or harm the environment.	1.03
	5.4.2	Uphold trust with your employers via being	2.01 and

Concerns to be Addressed in C5EPA	Clauses as Specific Advice for Consideration		IEEE-CS/ACM
		honest regarding the originality of your work. During RE appropriately use authorized resources and avoid unauthorized duplications of others work.	2.0
	5.4.3	Carryout adequate reviewing and validations of requirements specifications and other related documents, identify any plagiarism issues, and ensure the truthfulness of them.	3.10
	5.4.4	Ensure that software engineers know the employer's policies regarding all RE processes including requirements elicitation, specification, validation, and managements.	5.03
	5.4.5	Fully acknowledge the work of colleagues, and refrain from taking undue credit.	7.03
Con. 5.5 Denial of agreements	5.5.1	Moderate the interest of your clients, employer, software engineers, and the public. Ensure that requirements are used ethically, and prohibit the denial of agreements.	1.02
	5.5.2	Software engineers should be honest regarding their abilities to achieve agreements and communicate openly with employers and clients to avoid denial of agreements.	2.01
	5.5.3	Not only fair remuneration but also there should be a fair distribution of potential hazards and benefits.	5.07
	5.5.4	Software engineers should not prioritize personal interests over agreements made with stakeholders, ensuring professional integrity.	6.05
	5.5.5	Acquire and maintain professional competency,	8.01 and

Concerns to be Addressed in C5EPA	Clauses as Specific Advice for Consideration	IEEE-CS/ACM
	improve your skills in developing effective communication and negotiation strategies, carefully navigate complex processes of RE, and manage and avert denial of agreements that will exist between clients, employers, sponsors, other key stakeholders, and you (software engineer).	8.02

In this component, C5EPA, the five serious concerns are recognized. For each of them, possible solutions to reduce their existence during RE processes are given. These suggestions are put as a clause. Con. 5.1 has nine, Con. 5.2 tewenty, Con. 5.3 fifteen, Con.5.4 nine, and Con. 5.5 therteen cluases. Table 5.12 presents an overview of clauses and concerns in C5EPA.

Table 5.12: Ethics for Prohibited Activities – Concerns and Clauses

Concerns in C5EPA	Number of Clauses	Adapted From (IEEE – CS/ACM …)	
		Principles	Clauses
Con. 5.1 Lack of Honesty	9	1, 2, 4, 6, 7, and 8	1.06, 2.05, 2.08, 4.03, 6.05, 7.07, 8.07 and 8.08
Con. 5.2 Carelessness	20	1, 2, 3, 4, 5, 6, 7, and 8	1.01, 1.03, 2.01, 2.02, 2.05, 2.09, 3.01, 3.11, 4.01, 4.02, 5.01, 5.02, 5.07, 5.08, 5.09, 6.03, 6.05, 6.06, 7.01, 7.02, 7.03, 8.01, and 8.08
Con. 5.3 Evil requirements …	15	1, 2, 3, 4, 5, 6, 7, and 8	1.03, 1.04, 1.06, 2.01, 2.02, 2.06, and 2.07, 3.03, 3.08, 3.11, 4.01, 4.05, 5.01, 5.02,

			5.10, 6.07, 7.01, 7.02, and 8.03
Con. 5.4 Plagiarism	9	1, 2, 3, 4, 5, 6, 7, and 8	1.01, 2.01, 3.10, 4.01, 4.02, 4.04, 5.03, 6.12, 7.03, and 8.04
Con. 5.5	13 clauses		1.02, 1.03, 2.01, 3.08, 4.05, 5.04, 5.07, 6.02, 6.05, 7.08, 8.01, and 8.02

Generally, the proposed framework integrates these five key components discussed above. Each component of the framework has been carefully chosen to address the concerns within them using the stated clauses that were adapted from IEEE-CS/ACM JTFSECEPP. By combining these elements (the components, concerns, and the IEEE-CS/ACM JTFSECEPP), the proposed framework can be used as an RE guideline for software engineers to proactively identify those concerns and take measures to diminish their presence in equirements. The diagrammatic representation of the proposed framework is depicted below in Figure 5..2.

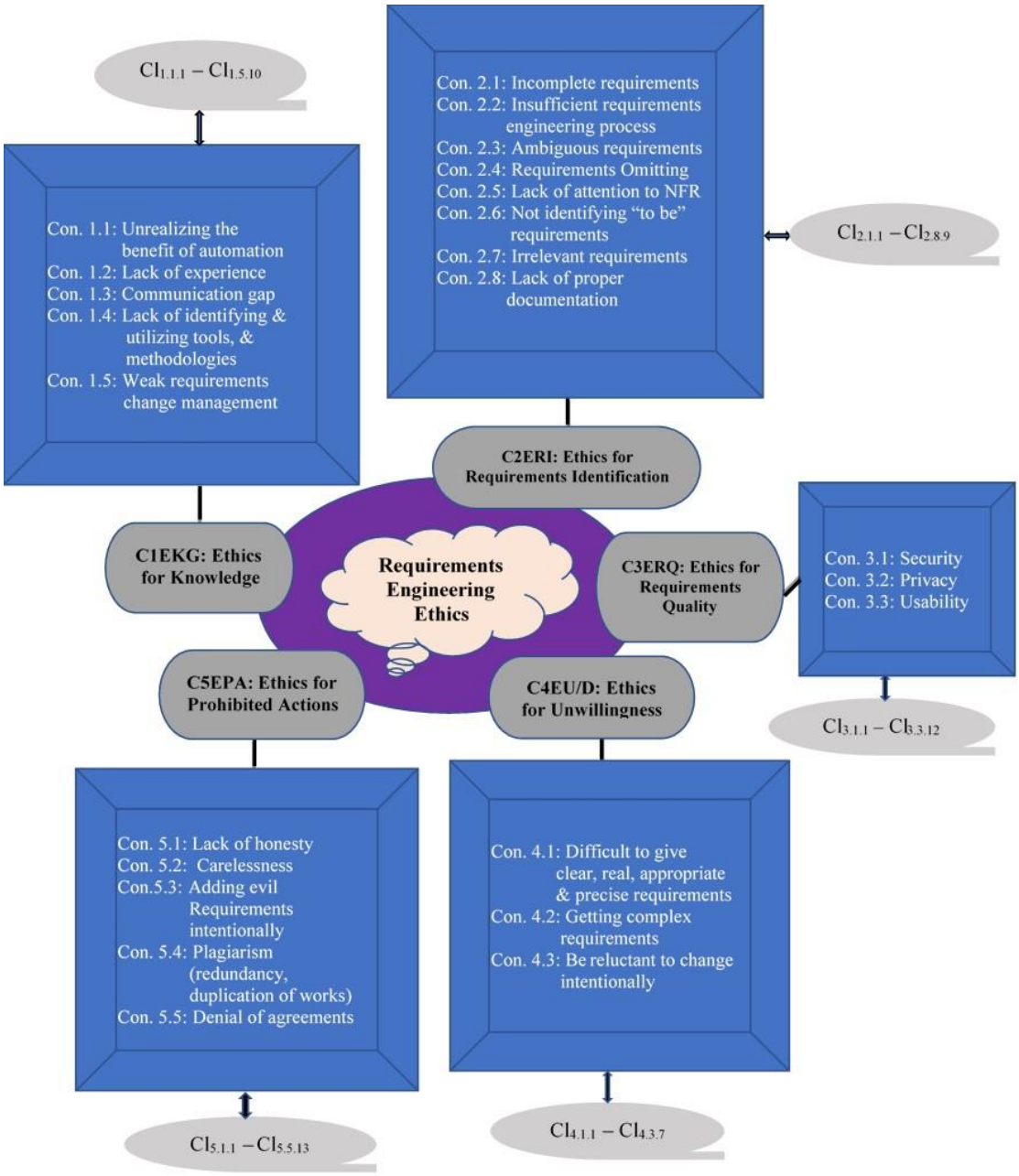

Figure 5.2: Diagrammatic Representation of the Proposed Framework

5.4 Checklists for the Framework Guideline

The term checklist can be used in different forms and different contexts [158]. Checklists are used in many fields of study to minimize errors [158]. Checklists can be used as instruments to unify our day-to-day responsibilities in our profession and to substantiate and prioritize the most important activities [159]. Checklists are formats created to perform repetitive tasks and regulate amenability with a list of requirements in an orderly and systematic manner [159]. Mainly, checklists are

essential to carry out activities in which no important tasks are forgotten and to perform tasks in an established manner.

5.4.1 Types of Checklists

Checklists can be used in a variety of ways to assist performances. Primarily, checklists are a very useful tool for memory aid. According to their usage, there are different types of checklists. Chaparro *et al*. [158] discussed the following types of checklists.

Table 5.13 Taxonomies of Checklists [163, p. 24]

Types of Checklists	Definition	Example
1. Laundry List	A loosely organized list of items where the order is not of primary importance	Shopping list, list of items to take on a trip, medical equipment list
2. Iterative Checklist	A form of sequential checklist that is used multiple times to check or evaluate a system's state or status	Checklist to evaluate the performance of a system after a series of adjustments to fine-tune the performance of the system
3. Diagnostic Checklist	The checklist is used to troubleshoot a system, often using a flowchart organization	Medical protocols used by emergency medical technician
4. Criteria of Merit Checklist	List of evaluative dimensions that users are expected to employ to judge the overall quality of a product	List of evaluative dimensions judges (i.e., color, appearance, taste) are instructed to use when judging food entries in a cooking contest
5. Procedural Checklist	Complex, lengthy, or critical tasks are performed	Assembly instructions for

	intermittently to make sure certain tasks are performed	furniture
6. Preparation Checklist	Multistep procedures certifying that all the steps are performed; order of execution may be less important	Shopping list, collection of necessary supplies for a camping trip
7. Problem – Solving Checklist	Sequential, multistep procedure with decision points used to support diagnoses, troubleshooting, and so on	Medical protocol, selecting an appropriate statistical test
8. Prevention Checklist	Checklists are designed to address known causes of errors or high-risk, high-hazard conditions that may result in injuries or the death of users.	Equipment setup checklist

For this research, a hybrid of iterative, problem-solving, and prevention types of checklists are appropriate and used since the main purpose of our proposed checklist is designed to address iteratively the identified, known ethical problems of requirements engineering, which has a high impact on the requirements. The proposed checklists will be used as a multistep procedure to check iteratively whether those identified concerns are addressed or not.

During RE processes, software engineers perform multiple steps. We are proposing this framework to be used daily for the consideration of ethical concerns for RE. The checklist will be used as a tip to check whether the identified concerns are addressed or not using the given clauses. As discussed above, within each of the five categories, we have identified ethical concerns. Under each concern, some clauses should be considered to minimize the corresponding identified concern. During the implementation of the framework, a software engineer can crosscheck in short by

referring to the checklists regarding each concern's advices. Table 5.14 highlights the suggested checklist for use.

Table 5.14 Checklists for Utilizing the Proposed Framework

No.	Strategies	Go to and Recursively Consult Clauses
1	**Con. 1.1:** Not realizing the benefit of automation **S1:** Ensure the presence of well awareness and realization of automation. • Avoid the frustration of customers concerned about the intended system that will be automated. • Try to ensure that customers are realizing the benefits of automation in a better manner.	1.1.1 to 1.1.15
2	**Con. 1.2:** Lack of experience **S2:** Continuously check your experience and improve it. • Take a rain check on lack of experience on requirements engineering ethics	1.2.1 to 1.2.11
3	**Con. 1.3:** Communication gap **S3:** Be ensuring absence of communication barriers between technical and non-technical parties.	1.3.1 to 1.3.18
4	**Con. 1.4:** Lack of identifying and utilizing tools, methodologies **S4:** Check proper selection and use of tools, methodologies, and techniques	1.4.1 to 1.4.11
5	**Con. 1.5:** Weak requirements change management **S5:** Try to minimize weak requirements change management	1.5.1 to 1.5.10
6	**Con. 2.1:** Incomplete requirements **S6:** Ensure the inclusion of all requirements	2.1.1 to 2.1.18
7	**Con. 2.2:** Insufficient requirements engineering process **S7:** confirm the presence of adequate requirements	2.2.1 to 2.2.16

No.	Strategies	Go to and Recursively Consult Clauses
	engineering processes	
8	**Con. 2.3:** Presence of ambiguous requirements **S8:** check whether all requirements are specified in a clear and simple term or not	2.3.1 to 2.3.10
9	**Con. 2.4:** Requirements Omitting **S9:** Refine all the requirements and add the forgotten one	2.4.1 to 2.4.11
10	**Con. 2.5:** Lack of attention to nonfunctional requirements **S10:** Give attention to nonfunctional requirements	2.5.1 to 2.5.15
11	**Con. 2.6:** Not identifying the real "to be" requirements **S11:** Care towards compulsory "to be" requirements	2.6.1 to 2.6.12
12	**Con. 2.7:** Collecting irrelevant requirements **S12:** Eliminate unnecessary requirements and details	2.7.1 to 2.7.10
13	**Con. 2.8:** Lack of proper documentation **S13:** Verify having well–prepared requirements specification documents.	2.8.1 to 2.8.9
14	**Con. 3.1:** Security concerns **S14:** Specify strong authentication methods.	3.1.1 to 3.1.13
15	**Con. 3.2:** Privacy concerns **S15:** Be ensure the presence of reliable confidentiality requirements.	3.2.1 to 3.2.13
16	**Con. 3.3:** Usability concerns **S16:** Make sure that requirements are usable or not for the intended purpose.	3.3.1 to 3.3.12
17	**Con. 4.1:** Unwilling to give clear, real, appropriate, and precise requirements **S17:** Appreciate intended customers to get the required requirements for a system.	4.1.1 to 4.1.14
18	**Con. 4.2:** Getting too complex requirements	

No.	Strategies	Go to and Recursively Consult Clauses
	S18: Encourage customers to clearly articulate their needs in their own word.	4.2.1 to 4.2.11
19	**Con. 4.3:** Be reluctant with changes intentionally **S20:** Clearly promote the benefits of automation in practical manners.	4.3.1 to 4.3.7
20	**Con. 5.1:** Lack of honesty **S20:** Be obeyed by your personal ethics as well as professional Ethics.	5.1.1 to 5.1.9
21	**Con. 5.2:** Carelessness **S21:** Always remember your professional responsibility.	5.2.1 to 5.2.20
22	**Con. 5.3:** Adding evil requirements for some purpose /Deceptive Requirements **S22:** Keep away from doing any kinds of deception on requirements.	5.3.1 to 5.3.15
23	**Con. 5.4:** Plagiarism (redundancy, duplication of works) **S23:** Do not use the work of others, including requirements, as your own.	5.4.1 to 5.4.9
24	**Con. 5.5:** Denial of agreements **S24:** Ensure the presence of a consent form/keep a specified memorandum of understanding.	5.5.1 to 5.5.13

This checklist ensures whether the identified concerns are addressed or not. During RE, a software engineer can use this checklist to determine whether the identified issues are resolved or not by looking at them forward and backward to consider the listed concerns accordingly.

5.5 IEEE-CS/ACM JTFSECEPP Adaptation

In this research, adapting the IEEE-CS/ACM JTFSECEPP to the proposed framework is crucial for tailoring the famous and mature software engineering codes of ethics to RE ethics. The IEEE-CS/ACM JTFSECEPP is well recognized as a foundation code of ethics for the software engineering discipline [160, 161].

We chose this code because of its numerous benefits. The main reason for adapting this code of ethics is that it is very useful, which was suggested by the two large, well-known, and experienced organizations, IEEE and ACM. The IEEE-CS/ACM JTFSECEPP provides a solid foundation for understanding the broader concepts of ethics in software engineering. It offers a comprehensive and strong framework for professional conduct in software engineering, emphasizing fairness and public safety. This code of ethics covers many dimensions of the software engineering field. However, due to the specific contexts of this study, RE ethics and the IEEE-CS/ACM JTFSECEPP codes of ethics needed to be modified to specific situations. Directly using it for specific situations, like RE, required several modifications to ensure its relevance and efficiency. This adaptation not only improves the quality of software products but also encuarages professionalism among requirements engineers and progresses the practices of RE ethics. Before adapting the IEEE-CS/ACM JTFSECEPP, we undertook a thorough review of relevant literature and examined the code itself. Additionally, we consulted domain experts, including software engineers, software developers, and other practitioners in the field of software engineering. Through these literature reviews and consultations, we gained valuable insights and received recommendations regarding the suitability of the IEEE-CS/ACM JTFSECEPP. Following this, we proceed to adapt this well-established code of ethics.

Key modifications were made on the IEEE-CS/ACM JTFSECEPP to adapt it for RE ethics context, specifically to align the clauses with the identified critical problems of RE. In the IEEE-CS/ACM JTFSECEPP eighty clauses were stated within all the eight principles. Principle one holds eight clauses, principle two consists of nine, principle three nine clauses; principles four to eight encompass six, twelve, thirteen, eight, and nine clauses respectively. To provide specific pieces of advice as sets of clauses, we adapted clauses from each of the IEEE-CS/ACM JTFSECEPP principles.

For the proposed framework, we stated about three hundred two clauses corresponding to each serious concern identified during our research. When we adapted, we did not directly take the clauses. However, by using the clauses of IEEE-CS/ACM JTFSECEPP as a benchmark or starting point, we tailored them to the context of RE, and we are trying to make them detailed, clear, and precise. Additionally, we put each detailed and specific clause per the identified concerns to be addressed. By using a single clause, we even stated various clauses that would be suitable for each problem that were dug out in this research investigation.

For example, here we present two examples to show how IEEE-CS/ACM JTFSECEPP was adapted to address specific concerns in the context of our proposed framework.

Example 1: IEEE-CS/ACM JTFSECEPP Principle 1, Clause 1. It says

"1.01: Accept full responsibility for their own work."

In the context of our work, this clause is tailored to seven clauses that will be suitable to address seven concerns from the first, second, third, and fifth components of the proposed framework. These are

1.1.1 Ensure your descriptions of automation benefits are accurate and based on evidence, taking responsibility for their reliability. (for not realizing the benefit of automation concern)

1.3.1 To take responsibility for accurate interpretation and implementation of requirements, you should ensure the presence of clear communication throughout RE processes (for communication gap)

2.1.1 Engage in stakeholders effectively and considering their holistic needs to contribute to the public good by ensuring that software systems meet the diverse needs of users and stakeholders. (for incomplete requirements)

2.2.1 Ensure the presence of a clear distribution of roles and responsibilities for elicitation, verification, validation, and monitoring of requirements. (for insufficient requirements engineering processes)

2.2.2 Do your best for the public as well as clients' interests, by avoiding inclusions of harms on requirements, to the clients, society besides on the environment. (for insufficient requirements)

3.3.1 Ensure the reliability of requirements to prevent harm or inconvenience to users. Contribute to user satisfaction and well-being by ensuring that software systems are easy to use and effective in meeting users' needs and public well-being. (for usability concerns)

5.2.1 Endorse conscientious behaviour through taking full responsibility for software engineer's work during RE. This ensures that the work in RE processes is thorough and accurate. (for carelessness)

Example 2: IEEE-CS/ACM JTFSECEPP Principle 8, Clause 1. It says

"8.01 Further their knowledge of developments in the analysis, specification, design, development, maintenance and testing of software and related documents, together with the management of the development process."

We adapted this clause into eight clauses as advice for eight concerns from the first and fifth components of the proposed framework. These comprise

1.2.11 Take personal responsibility for your professional growth in the area of RE. Regularly seek feedback and take advantage of training opportunities to improve your expertise. (for lack of experience)

1.3.18 Ensure that ethical standards are clearly communicated and expectations are precisely mentioned for members involved in RE. (for communication gap)

1.4.8 Try yourself to recurrently update your knowledge to select and use the most relevant tools, techniques, and methodologies that will be suitable for your project. (for lack of knowledge in identifying tools, techniques, …)

1.4.9 Try to continually endeavour to further enhance your knowledge of the requirements engineering processes. (for lack of knowledge in identifying tools, techniques, …)

1.4.10 Integrate your professional and ethical knowledge and work by following appropriate standards and principles. (for lack of knowledge in identifying tools, techniques, …)

1.5.10 Maintain professional competence and integrity, stay updated with best practices, and continually improve requirements management skills. (for weak requirements change management)

5.2.20 Strive for quality in your work and continuous improvements of your knowledge and ability. This helps software engineers stay updated on best practices and techniques and reduces the likelihood of carelessness. (for carelessness)

5.5.13 Acquire and maintain professional competency, improve your skills in developing effective communication and negotiation strategies, carefully navigate complex processes of RE, and manage and avert denial of agreements that will exist between clients, employers, sponsors, other key stakeholders, and you (software engineer). (for denial of agreements)

Similarly, all the eighty IEEE-CS/ACM JTFSECEPP clauses were adapted to address specific problems, resulting in three hundred two clauses. The following table, Table 5.15, displays the clauses of IEEE-CS/ACM JTFSECEPP and their adaptations to address (at least to reduce) a particular concern during RE.

Table 5.15 Mapping IEEE-CS/ACM JTFSECEPP Clauses to Concerns

IEEE-CS/ACM…Clauses	Adapted to Proposed Framework Clauses
Principle 1	**PUBLIC**
1.01	1.1.1, 1.3.1, 2.1.1, 2.2.1, 2.2.2, 3.3.1, 5.2.1

1.02	1.1.2, 2.2.2, 2.2.3, 2.4.1, 2.6.1, 2.7.1, 3.1.1, 3.2.1, 5.5.1, 5.5.2
1.03	1.5.1, 2.1.3, 2.4.1, 2.5.1, 2.6.1, 3.1.2, 3.2.2, 3.3.2, 4.1.1, 4.3.1, 5.2.2, 5.3.1, 5.4.1, 5.5.3,
1.04	1.3.2, 2.8.1, 3.2.3, 4.1.1, 4.2.1, 5.3.2
1.05	1.4.1, 2.1.4, 1.4.1, 2.1.4
1.06	1.2.1, 1.4.2, 5.1.1, 5.1.2, 5.3.3
Principle 2	**CLIENT AND EMPLOYER**
2.01	1.1.3, 1.1.4, 1.2.2., 1.2.3, 2.1.5, 2.3.2, 2.5.2, 2.7.2, 4.1.2, 4.2.2, 5.2.3, 5.3.4, 5.4.2, 5.5.4
2.02	1.1.3, 1.3.3, 5.2.4, 5.3.5
2.03	1.3.4, 1.3.5, 1.3.6, 2.1.6, 2.4.1, 2.5.3, 2.6.3, 3.2.4, 4.1.1, 4.1.5
2.04	1.1.5 , 2.3.3 , 2.8.2 , 3.3.3 , 4.2.3 ,
2.05	1.3.17, 2.3.4, 2.4.4 , 3.1.3 , 3.2.5 , 4.1.3 , 4.3.2 , 5.1.3 , 5.2.5 ,
2.06	1.1.6, 1.3.7 , 3.3.4 , 5.3.6,
2.07	1.1.6, 1.3.7, 2.5.4, 5.3.6 ,
2.08	3.2.6, 5.1.4 ,
2.09	2.2.4 2.2.5 5.2.5

IEEE-CS/ACM… Principles and Clauses	Adapted to Proposed Framework Clauses
Principle 3	**PRODUCT**
3.01	1.1.7, 1.2.4, 2.1.7, 2.3.5 , 2..3.6 , 2.4.5, 2.5.5 , 2.6.4 , 2.7.3 , 3.1.4 , 3.2.7, 3.3.5, 3.3.6 , 4.1.6 , 4.1.7 , 4.2.4, 5.2.6
3.02	1.1.8, 1.3.8, 2.1.8 , 2.2.6, 3.2.8 , 4.1.6 , 4.1.7 , 4.2.5,
3.03	2.1.9,, 2.5.6, 3.1.5, 5.3.7
3.04	1.4.3, 1.4.4, 3.1.6, 3.3.7 ,
3.05	1.2.5, 1.5.2,, 2.3.7, 2.4.6, 2.5.7, 2.6.5
3.06	3.2.9, 4.3.3, 1.3.9,
3.07	3.3.8
3.08	5.3.8, 5.5.5 , 2.3.8, 2.6.6 , 2.8.3,
3.10	1.3.10 , 2.3.10, 3.3.10, 5.3.10 , 5.4.3
3.11	2.8.4 , 5.2.6 , 5.3.8,
3.12	3.1.7, 3.2.10,
3.13	3.1.8
3.14	3.1.9
Principle 4	**JUDGMENT**
4.01	1.1.9 , 1.4.2, 2.1.10, 2.3.9, 2.4.7, 2.5.8, 2.6.7, 3.2.11, 4.1.8, 4.2.6 , 5.2.7 , 5.3.9, 5.4.4
4.02	2.5.9, 3.3.9, 4.1.8, 5.2.8, 5.4.4
4.03	2.8.5, 3.2.12, 5.1.5, 1.1.10, 1.3.10, 1.5.3, 2.1.11, 2.2.7,

	2.3.10, 2.7.4
4.04	5.4.5
4.05	1.5.4 , 2.1.12, 4.1.10, 4.1.11, 4.3.4, 5.3.10, 5.5.6, 5.5.7
4.06	4.1.9

IEEE-CS/ACM…Clauses	**Adapted to Proposed Framework Clauses**
Principle 5	**MANAGEMENT**
5.01	1.2.6, 1.5.5, 2.1.13, 2.2.8, 2.5.10, 4.2.7, 4.3.5 , 5.2.9, 5.3.11
5.02	1.3.11, 1.5.6, 2.2.9, 2.6.8, 5.2.10, 5.3.11
5.03	1.1.11, 2.8.6, 3.1.10, 3.1.11, 5.4.6
5.04	1.3.12 , 1.3.13, 1.5.7, 5.5.8
5.05	2.2.10, 2.7.5, 5.3.12
5.06	1.2.7.
5.07	5.2.11, 5.5.9
5.08	5.2.12
5.09	5.2.13
5.11	4.3.6
Principle 6	**PROFESSION**
6.01	1.1.12, 2.2.11, 2.5.11, 4.2.8, 4.2.9,

6.02	1.3.14 , 2.4.8, 2.6.9, 4.1.1, 4.1.12, 5.5.10
6.03	1.2.8, 1.4.1, 5.2.14
6.04	3.3.10
6.05	1.5.8, 2.2.13, 5.1.6, 5.2.15
6.06	1.1.13, 1.3.15, 3.2.13, 5.2.16
6.07	5.3.13
6.08	2.1.14, 2.2.12, 2.7.6, 2.8.7,
6.09	1.3.17
6.11	2.4.9 , 2.7.7, 5.5.11
6.12	5.4.7

IEEE-CS/ACM… Principles and Clauses	**Adapted to Proposed Framework Clauses**
Principle 7	**COLLEAGUES**
7.01	1.2.4, 1.3.17, 2.1.15, 2.7.8, 5.2.17, 5.3.14
7.02	1.2.9, 1.2.10, 1.4.5, 2.4.10, 2.5.12, 2.7.9, 5.2.18, 5.3.14,
7.03	2.1.16, 4.2.10, 5.2.19, 5.4.8
7.04	1.4.6, 1.5.9, 2.5.13 , 2.6.10, 2.8.8
7.05	1.3.16, 1.3.16, 3.3.11
7.06	3.1.12
7.07	4.1.13, 5.1.7

7.08	1.4.6, 4.1.13, 5.5.12
Principle 8	**SELF**
8.01	1.2.11, 1.3.18, 1.4.8, 1.4.9, 1.4.10, 1.5.10, 5.2.20, 5.5.13
8.02	1.1.14, 1.1.15, 1.4.11, 1.3.18 , 2.5.14, 2.6.11, 2.7.10, 3.1.13, 3.3.12, 4.2.10, 5.5.13, 5.2.20
8.03	2.8.9, 5.3.15
8.04	2.2.14, 2.4.11 , 2.5.15 , 2.6.12, 5.4.9
8.05	2.2.15 , 2.1.17,
8.06	1.3.18, 2.2.16,
8.07	4.1.14, 5.1.8
8.08	5.1.9
8.09	2.1.18

Adapting the IEEE-CS/ACM JTFSECEPP in this way adds a new dimension to its application in software development projects, particularly in RE. Moreover, it will open opportunities for future research to explore its adaptability in other perspectives in the area. Furthermore, this adaptation also bargains perceptions into how existing frameworks can be transformed to address specific problems without losing their theoretical essence.

Additionally, adapting the IEEE-CS/ACM JTFSECEPP has practical implications. From a practical standpoint, adapting this code of ethics to the proposed framework improves the framework's real-world applicability. Software engineers and software industries can benefit from the framework that is closely aligned with the realities of their specific environment, rather than applying generic codes of ethics.

Chapter Six – Proposed Framework Evaluations

6.1 Introduction

To evaluate the proposed framework, different methods are employed. In the following subsections, we discuss the tools and methods used to validate the proposed framework and the processes we followed to select and evaluate it.

6.2 Preparation of Validation Tools

Primarily, we derived survey questions from the literature [6, 162] and they are modified to fit the context of this study. The questionnaires were lengthy and contained both closed- and open-ended questions in their initial original stage. Furthermore, there were just four options available for the questions: "high," "medium," "low," and "none." Experts in the field are requested to provide comments and suggestions for possible changes and improvements. First, the questions are given to two experts. These two experts critically looked at the tools and modified them into a shorter and clearer format. In addition, the other three subject-matter experts were handed the questionnaires and asked for their input. Like the initial experts, these provided insightful criticism and improved the questionnaire to be more understandable. The experts recommended using the most popular questionnaire format, the five-point Likert scale, in place of the four options listed above. Furthermore, in between the two positives and the two negatives, they put “neutral” to increase the validity of the answers given by the participants. The addition of the "neutral" option encourages respondents to provide a response that is neither positive nor negative, rather than forcing them to select an option that doesn't align with their viewpoints [95].

Each of the five experts is provided feedback and comments while recursively improving the questionnaires. These questionnaires were updated considering the insightful remarks and suggestions. Finally, questionnaires with a five-point Likert scale are employed to validate the suggested framework. We chose this scale because it makes it easier to measure the opinions, feelings, and feedback that our participants, a group of software developers, experts, and practitioners, had on the suggested RE

framework. On the Likert scale, “Very Good”, “Good”, “Neutral”, “Poor”, and “Very Poor” are the sentiments. Annexes IX and X show the last modified versions of the questionnaires [163].

6.3 Participant Selection

Preliminary and formal evaluations are the two evaluation levels for the suggested framework’s evaluation. During the initial stage of the evaluation, the proposed framework with the five-point Likert scale questionnaires is given to sixteen voluntary participants from whom data were gathered during the data collection stage of the research. These participants were selected from the case software companies using a volunteer sampling technique. The participants went through each framework component and evaluated the proposed framework using the developed questionnaires. Besides, the researchers observed and took feedback from them in person. With the provided comments and feedback, the proposed framework is thoroughly revised [163]. These volunteer experts were not involved in the formal evaluations of the framework. The major purpose of carrying out this initial evaluation is to ensure transparency, accuracy, and validity. This process assists in obtaining feedback, and credibility of the research. and to gain their viewpoints about the representation of concerns, the proposed framework, and its components.

For the formal validations of the proposed framework, participants are selected purposively. Experts with knowledge of software development, specifically in relation to RE, and with practical and theoretical expertise are chosen. During the selection of experts, their willingness is considered. Taking these things into consideration experts, nineteen from local and twenty-one from international software industries, are selected. Only thirty-one of them returned the completed questionnaires. The industry experiences of the participants in software development processes range from one to thirty-five years. The following pie chart shows the experiences of those participants working on software development projects within the software industries.

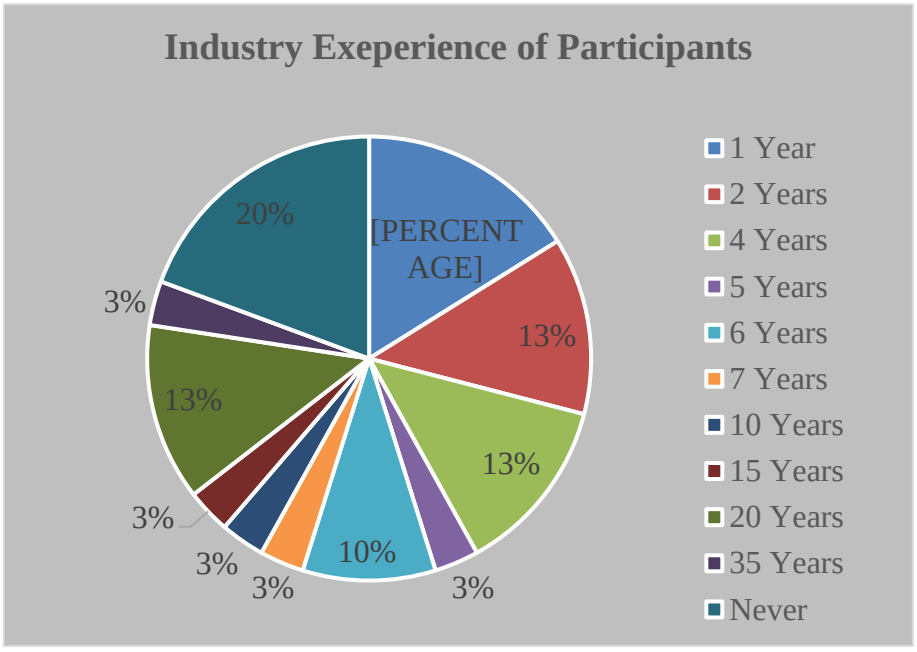

Figure 6.1: Industry Experiences of Participants

Furthermore, different participants with different work profiles who are working in different sections of industries are involved in the validation process [163]. The following pie chart indicates the participants' work profiles.

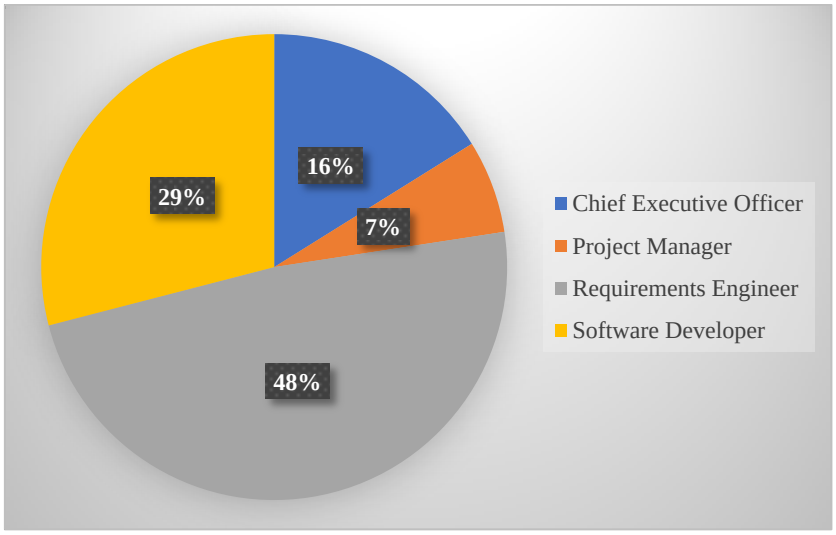

Figure 6.2 Participants Work Profile

6.4 The Validation Processes

Validation is essential to confirm whether the research met its objective [89]. To ensure the participants' understanding regarding the concepts used in the research, such as the three validity types and reliability, we provided clear definitions of them before presenting the questionnaires. Additionally, we designed the questions in a way that assessed conceptual understanding rather than assuming their prior knowledge. A pilot test was also conducted on five experts to identify any ambiguities and refine the questionnaires accordingly. The questionnaires assess the validity of the proposed framework [89]. These include face validity, content and construct validity (plus the content and construct validity of each of its framework's components), and its reliability. Furthermore, the appropriateness of the placement of the clauses for each of the RE's ethical concerns was also assessed. In addition, the parts of the framework that are the most challenging to follow and most importantly perceived are evaluated.

Before starting the evaluation process, the proposed framework, the foundation codes of ethics, IEEE-CS/ACM JTFSECEPP, from which the clauses are adapted as discussed in subsection 5.2, and the contextual definitions of the evaluation parameters, are elaborated to the participants via formal meetings, discussions, and communications media. After that the proposed framework is given to the experts for evaluation using the questionnaires. Side by side, definitions of validity and reliability in the context of this research are given for them. . Next, the questionnaires are filled out in two ways. One is in the form of FGDs ; the other is filled without focus group discussion sessions by using the initial elaboration. Three FGDs, each with seven members, were conducted by dividing the participants into smaller, manageable parts. For all the three groups, discussion sessions were organized on different days, and the proposed framework is presented to them. The research findings, the aim of the framework, its components, the main contributions of the proposed work for software engineers, the contextual definition of validity and reliability, and similar concepts are presented. Free discussions among the participants and researchers were carried out. After getting these discussions and presentation highlights, the participants thoroughly read the framework, which contains the concerns, advice to address those concerns,

and the corresponding IEEE-CS/ACM JTFSECEPP clauses from which the piece of pieces of advice is adapted, and answered the provided questionnaires accordingly. The entire partakers within three FGDs filled out and returned the questionnaires. For the other groups, the participants who filled out the questionnaires without coming and meeting at the physical location, a brief explanation about the proposed framework and the evaluation process is provided via communication channels. The following section discusses the results obtained during this evaluation process.

6.5 Expert Evaluation of the Proposed Framework

As mentioned in Section 6.3, questionnaires were distributed to forty purposively selected experts. However, only thirty-one of them were filled and returned. The returned questionnaires are coded from P1 to P31 to keep the confidentiality of the participants.

6.5.1 Demographic Description of Participants

The proposed framework is validated by twenty-three (74.2%) male and eight (25.8%) female participants. Besides, participants were asked for their ages. The age of about 60% of the participants ranges from 30 (thirty) to 55 (fifty-five) years. The remaining 40% are between 21 (twenty-one) to (twenty-nine) years old. The qualifications of those participants are professors, Ph.D. holders, Ph.D. candidates who are currently engaging in software development projects, and MSC and BSC holders underneath the fields of software engineering, computer science, technologies, and multimedia. The participants are currently engaged as software developers, researchers, and chief technology officers.

6.5.2 Assessments of the Validity and Reliability Results

In research, validity and reliability enhance clarity and minimize the biases of researchers [164]. In the context of this research, validity is considered as the extent to which the proposed framework can express the inclusion of appropriate clauses as sets of advice to address those identified RE issues or not [165]. Among the different types of validity [89], face, content, and construct validity were considered since they were relevant to the research. In this research, face validity refers to whether each of

the proposed framework components, corresponding concerns, and clauses presented in each framework's component looks relevant to its understandability, unambiguousness, and clarity, or not [56]. It also includes whether the proposed framework is feasible, readable, and clear in terms of language, usage, and coherency, or not [165].

Furthermore, content validity indicates the degree wherein the clauses in the framework are representative of addressing the corresponding identified RE concerns [164]. In addition, construct validity refers to whether the concepts mentioned in the proposed framework are easily understandable and easily operational or practicable on a day-to-day basis or not [165]. Moreover, it helps to check whether the proposed framework clauses are appropriately adapted from the IEEE-CS/ACM JTFSECEPP or not.

Additionally, the reliability of the proposed framework was assessed. In this research, reliability is considered to determine whether the indicated specific pieces of advice as clauses, are consistent or not. It also includes internal consistency, which means the extent to which the framework items are intercorrelated and consistent in measuring the corresponding identified issue or not [165]. The major evaluation results are discussed below.

Below, in Table 6.1, the results of the validity and reliability evaluation for the proposed framework are presented.

Table 6.1 The Validity and Reliability Evaluation Results

No.	Parameters	Sentiments									
		Very Good		Good		Neutral		Poor		Very Poor	
		Number of Respondents	Percentage (%)	Number Respondents	Percentage (%)	Number Respondents	Percentage (%)	Number Respondents	Percentage (%)	Number Respondents	Percentage (%)
1.	Face										

	Validity	11	35.5%	20	64.5%						
2.	Overall Content Validity	7	22.6%	23	74.2%	1	3.2%				
2.1.	Content Validity C1EKG	12	38.7%	19	61.3%						
2.2.	Content Validity C2ERI	12	38.7%	19	61.3%						
2.3.	Content Validity C3ERQ	11	35.5%	18	58.1%	2	6.5%				
2.4.	Content Validity C4EU/D	10	32.3%	18	58.1%			3	9.7%		
2.5.	Content Validity C5EPA	9	29%	21	67.7%			1	3.2%		
3.	Overall Construct Validity	11	35.5%	18	58.1%	2	6.5%				
3.1.	Construct Validity C1EKG	13	41.9%	15	48.4%	2	6.5%	1	3.2%		
3.2.	Construct Validity C2ERI	10	32.3%	19	61.3%	2	6.5%				
3.3.	Construct Validity C3ERQ	6	19.4%	22	71%	3	9.7%				

3.4.	Construct Validity C4EU/D	5	16.1%	23	74.2%	2	6.5%	1	3.2%		
3.5.	Construct Validity C5EPA	11	35.5%	17	54.8%	3	9.7%				
4.	Overall Reliability	11	35.5%	18	58.1%	2	6.5%				
4.1.	Internal Consistency	9	29%	21	67.7%	1	3.2%				

The first question that was asked of participants was:

1. Please show the extent to which you believe this framework to have face validity following a thorough perusal of it [162].

As indicated in Table 6.1, 35.5% of the respondents replied that the proposed framework has a very good face validity. This implies that the proposed framework is very relevant to its understandability, simplicity, clarity, and coherency in terms of language and usage. On the other hand, 64.5% of the respondents, responded that the proposed framework has good validity. All the respondents accepted the face validity as good and above.

2. The results of the second question, “Please indicate to what degree you consider this framework has overall content validity," are indicated as follows:

The overall content validity evaluation results suggest that most participants have positive views concerning the proposed framework. Specifically, 96.8% of respondents rated the overall content’s validity result as positive, with 22.6% considering it very good and 74.2% rating it as good. This high percentage of positive responses indicates strong agreement among participants regarding the framework's overall content validity. Likewise, the small proportion (3.2%) of participants selecting the neutral option implies that only a minority of respondents remained

undecided about the framework's content validity. Overall, these findings, among the participants, suggest that the proposed framework is representative of resolving the related RE concerns.

3. “Please indicate to what degree you consider **C1EKG** of this framework to have content validity.”

   The evaluation results for the content validity of C1EKG show that a considerable number of respondents, 38.7%, reflect the content validity of this framework’s component to be very good. This indicates that the clauses discussed in C1EKG strongly aligned with the identified ethical issues. Furthermore, the majority, 61.3% of respondents still believe that the content’s validity is good. This demonstrates that even though a significant percentage of respondents did not evaluate them as very good, they did acknowledge the ability and applicability of the contents in the C1EKG. All things considered, these findings point to a favorable belief on C1EKG's content validity among the participants, with a sizable percentage recognizing good and above-quality.

4. “Please indicate to what degree you consider **C2ERI** of this framework to have content validity.”

   As shown in Table 6.1, a considerable number of respondents, 38.7%, believe that the content validity of C2ERI is very good. This reveals that the clauses included in C2ERI fit the discovered ethical problems. Additionally, many respondents, 61.3%, consider that the content validity of the component is good. This shows that even though a notable number of respondents did not perceive it as very good, they acknowledge the usefulness and capabilities of the contents in the C2ERI. Overall, these results show that respondents have a positive view regarding the content validity of this framework's component, with a significant portion identifying it as good and above.

5. “Please indicate to what degree you consider **C3ERQ** of this framework to have content validity."

The content validity of C3ERQ indicates that a considerable number of respondents perceive it positively. Specifically, 35.5% evaluated it as very good, while a higher percentage, 58.1%, viewed it as good. Only a minority, 6.5%, selected the neutral alternative, suggesting that a large portion of respondents have strong views regarding its content validity. Generally, 93.6% of the respondents agreed that the component has good and above content validity. However, attention should be given for the respondents who stand on neutral sentiment.

6. "Please indicate to what degree you consider **C4EU/D** of this framework to have content validity."

With regard to C4EU/D, the evaluation's findings show that the participants' thoughts toward content validity of it, are promising. About 90.4% of the respondents have positive perceptions, with more than half, 58.1%, evaluating it as good. This suggests that most of the participants thought its content as satisfactory. Nonetheless, it is crucial to remember that 9.7% of participants gave the content validity a poor rating implying that there might be some areas in need of clarification or adjustments to increase its content validity. All in all, the findings point to a good degree of content validity, more than 90% of respondents rated the content validity of C4EU/D as good and above.

7. "Please indicate to what degree you consider **C5EPA** of this framework to have content validity."

The C5EPA evaluation findings indicate a generally positive insight into its content validity among participants. A considerable percentage, 67.7%, expressed that the content validity was good, showing that it adequately represents the intended content. In addition, 29% of participants assessed the content's validity as very good, indicating an even stronger validation of its ability to accurately address the corresponding ethical concerns. Conversely, 3.2% of respondents have a neutral opinion. Overall, the replies from most participants indicate that the results display the intended purpose positively.

8. Please show the extent to which you believe this framework to have overall construct validity following a thorough perusal of it [162].

The evaluation results of the overall construct validity of the proposed framework show that a notable percentage of participants perceive it as good and above. Explicitly, 35.5% of respondents rated it as very good, and a majority of 58.1% evaluated the construct validity as good, indicating a generally positive perception of its effectiveness. However, it is important to take into account that a small number, 6.5%, of them stand on neutral sentiment, revealing some room for improvement with certain aspects of the framework's construct validity. All in all, while many participants rated good and above in the framework's construct validity, addressing the concerns of the minority group, who stand on the neutral sentiment, could enhance its overall effectiveness and acceptance.

9. "Please indicate to what degree you consider **C1EKG** of this framework to have construct validity."

The construct validity evaluation results of C1EKG disclose a good tendency among participants' perceptions. Nearly half of the respondents, comprising 48.4%, expressed a positive viewpoint, characterizing the construct validity of it as good. In addition, a significant proportion, account for 41.9% of participants, rated even further to describe the construct validity as very good. Overall, over 90% of the respondents accepted the construct's validity as good and above. While most respondents viewed it positively, it is important that a smaller number, encompassing 6.5%, accepted a neutral sentiment, and a small number of them, 3.2%, perceived it as having poor construct validity. These results point to the need for further work to improve the construct validity of this framework's component. It is vital to explore further to understand the factors contributing to this neutrality and dissatisfaction and whether any adjustments or clarifications are needed to address potential areas of concern.

10. "Please indicate to what degree you consider **C2ERI** of this framework to have construct validity."

The evaluation results of the C2ERI construct validity suggest a positive observation among participants. Explicitly, 32.3% of respondents perceived construct validity as very good, while 61.3% considered it good. Even though 6.5% of the participants stand on neutral sentiment, the overall findings suggest a generally consistent understanding and agreement among participants regarding the appropriateness of C2ERI to use it as a practical guideline for software engineers.

11. "Please indicate to what degree you consider **C3ERQ** of this framework to have construct validity."

Regarding the construct validity of C3ERQ, the evaluation results show that a large percentage of participants, about 90.4%, perceive it to have satisfactory construct validity. 19.4% of participants evaluated the construct validity as very good; a majority, 71%, assessed the construct validity as good, indicating positive perception of C3ERQ's clauses placement with the targeted ethical concerns. The remaining 9.7% of participants preferred being reserved to give either of the two extreme responses by standing on the neutral sentiment. All things considered, these results suggest that this component reveals promising construct validity, with a noteworthy proportion of participants expressing their belief in its ability to be easily applicable and operational for daily use in real-world scenarios.

12. "Please indicate to what degree you consider **C4EU/D** of this framework to have construct validity."

The evaluation results of C4EU/D indicate that a remarkable number of participants view it to possess good to very good construct validity. Specifically, more than 90% of participants rated the construct validity positively (both the very good and good responses). This shows that the concepts under C4EU/D are easily understandable and practicable. While a small proportion of participants, 6.5%, remained neutral in their assessment, only a minority, 3.2%, stated it was poor. Altogether, these findings suggest a good insight into its construct validity among the participants, with a majority viewing it positively. However, further investigation is required to address concerns raised by the minority who rated it poorly and who expressed neutrality. Participants who stand for neutral sentiment might have reservations or uncertainties

about some aspects of its construct validity, indicating potential areas for improvement and clarification.

13. “Please indicate to what degree you consider **C5EPA** of this framework to have construct validity.”

The evaluation results of C5EPA construct validity specify good insight among the participants. Correspondingly, a significant number of participants, 35.5%, expressed a high level of perception of the C5EPA construct validity; most, consisting of 54.8% of the participants, acknowledged its construct validity as good. A smaller yet notable portion, 9.7%, remained neutral in their evaluation, neither strongly confirming nor opposing its construct validity. Ultimately, these results suggest that C5EPA demonstrates promising construct validity, with a substantial portion of participants suggesting its construct validity as good and above.

14. “Please indicate to what degree you consider this framework to have overall reliability.”

As depicted in Table 6.1, many participants have favorable views regarding the framework's overall reliability. 93.6% of respondents described reliability as good and above. According to the participants' perspectives, this indicates a strong levels of trustworthiness and consistency in the framework’s performance. Additionally, the presence of only 6.5% expressing a neutral sentiment suggests that the framework’s reliability is generally perceived positively among them, with a minimal portion remaining undecided or indifferent. Overall, these results reflect a constructive judgment of the proposed framework’s reliability, with a sizable portion considering it very good and the majority inclined towards a good overall assessment.

15. “Please indicate to what degree you consider this framework to have internal consistency.”

Regarding internal consistency, 29% of the participants mentioned that the proposed framework has a very good internal consistency and 67.7% said it has good internal consistency. This indicates the presence of a strong level of agreement among them regarding the framework’s reliability and coherence. The small proportion, 3.2%, of

respondents who expressed a neutral option may indicate a slight ambiguity or lack of consensus among a minority of participants. Nonetheless, strong internal consistency is implied by most of the positive feedback, assuring the framework's capacity to maintain reliability.

6.5.3 Descriptive Assessment of the Results

The questionnaires are both closed and open-ended questions. The questions are prepared, validated, and given to the experts. These questions are provided to get experts' suggestions, recommendations, and feedback on the proposed framework.

The first open question deals with whether the placement of clauses in line with the identified RE concerns is appropriate or not. It says

1 After reading the proposed framework, please use "√" regarding your observation on the placement of activities with respect to each section?

- ☐ The placement is done appropriately and accurately without any mismatch
- ☐ The placement is done appropriately and accurately with very little modification
- ☐ The placement is done appropriately with some modification
- ☐ The placement is not done appropriately. It should be done again

Please also provide any comments and recommendations.

As a result, 32.3% or 10 of the participants indicated that the placement of clauses concerning each concern is appropriately and accurately done without mismatch. Whereas, 38.7% or 12 of them stated that the placement is done appropriately and accurately with a very little modification. Also, the remaining 29%, or 9 of them stated that the placement is done appropriately, but some modifications should be made, especially in components 3 and 4.

Generally, as observed from the reply, 70% of the respondents suggested that the placement of clauses for each framework component is done appropriately and

accurately. The remaining of them said it was done appropriately with some modifications. This implies that the placement of clauses is good for each stated ethical concern. The second open-ended question asked the participants to identify the framework's component that will become a challenge to follow throughout the regular course of software development. It states, "Which are perceived to be the most challenging to be followed? Among the forty respondents, only eighteen gave a reply to this question. Accordingly, six respondents said that C5EPA is challenging to implement in practice. On the other hand, most of them, more than 55% of them said that each of the proposed framework component clauses can be practiced in real work areas of software development.

The third question deals with identifying the most important component. It asks respondents to state their perspectives towards the relevance of each of the proposed framework components. It says, "From your observation, which are perceived as important?" Like the second question, this question is also been answered by only eighteen respondents. Accordingly, 30% of the experts stated that C2ERI and C5EPA are very important. They said that the problems mentioned under these categories are the basic bottlenecks to developing usable software systems. Besides, the clauses that are mentioned corresponding to each of these components are important. As a result, software engineers can use these components' clauses to minimize the occurrence of the indicated concerns. The other 28% of the participants mentioned that C4EU/D is more important than others. As they justified, to get everything plus to develop a software system, first and foremost, we must get the willingness of customers. As often mentioned, customers are "Kings." Without customers for a system, nothing would be done. Therefore, software engineers must get the real willingness and cooperation of the customers and other stakeholders, consequently, the clauses that are mentioned under C4EU/D are vital. Moreover, 28% of the participants explained that C3ERQ is the most important one. Activities that help to better consider quality parameters, such as privacy, security, and usability of requirements, are the most important ones.

6.6 Discussion: Interpretation of Research Findings

6.6.1 Key Contributions of the Research

This research proposes a framework and evaluates it using empirical data. The main aim of the proposed framework is to improve ethical decision-making in RE that fills gaps in the available software engineering codes, IEEE-CS/ACM JTFSECEPP [11].

The proposed framework is designed to be feasible, clear, and easily practicable on a day-to-day basis so that it can be easily employed by software engineers to consider ethical issues in RE. This research contributes to the field of software engineering, specifically for RE, by systematically organizing a framework based on the empirical research findings.

In addition, the research line up with the ongoing debate on software development ethics. Lurie and Shlomo [10] indorse an ethical-driven software development approach, integrating ethical considerations into software engineering. Similarly, Karim *et al.* [11] emphasize the necessity of ethical principles within the software development life cycles. This research complements these perspectives by ensuring RE practices not only follow methodological requirements but also align with the professional responsibilities of software engineers.

6.6.2 Interpretation of Evaluation Results

The evaluation result shows the proposed framework possesses good validity and reliability, supporting its relevance for practical applications in RE. The evaluation involves the validity, and reliability of the proposed framework and its components, the appropriateness of each adapted clause to address the corresponding concerns, the clauses' relevance to address the issues, and the challenges to use them in the day-to-day work of software engineers during RE. the face validity results indicate a positive perception, with all participants rating as good and above. This suggests that the proposed framework's structure and clarity align with the participants' expectations. Content validity results further demonstrate its effectiveness, with 96.2% of participants rating it as good or very good. However, C4EU/D received a lower rating (9.7% poor), highlighting the need for refinements in that component. Similarly,

construct validity results show that over 90% of participants found the framework understandable and practicable though C4U/D and C1EKG require further improvement based on participants' feedback. Reliability assessment supports these findings, with 95% of respondents confirming the framework's consistency. However, the 4.8% neutral responses suggest that some clauses require further clarity to confirm consistency across different RE environments.

Moreover, the participants' feedback from the FGDs and open-ended questionnaire questions highlights valuable insights into the proposed framework. While the overall framework structure was regarded as good, minor grammatical and formatting problems were recognized, suggesting the necessity of adjustments to improve even more on clarity and professionalism. More importantly, the principal recommendation was the incorporation of requirements validation within the C4U/D framework component. This incorporation would increase the consistency and clarity of software project requirements among stakeholders, customers, and software engineers, and, overall efficiency. Furthermore, the need for revision, modification, and functional alignment with intended use was emphasized, which implies the framework should be practicable for real-world implementations. These findings underscore the essential for iterative improvements based on the provided feedback from the research participants to ensure the framework complies with expectations. Generally, the evaluation results show that the proposed framework is promising to use as a guideline for RE ethics.

These findings echo with Karim *et al.* [11], who emphasize that ethical frameworks must be explicitly integrated into software engineering methodologies. The challenges in framework validation highlight the importance of continuous refinement and ethical considerations to address ever-increasing technological advancements and associated ethical consequences.

6.6.3 Comparison with the Existing Knowledge

Compared to the existing software engineering codes of ethics, IEEE-CS/ACM JTFSECEPP, the proposed framework offers a structured, detailed, and validated guideline that emphasizes practical applicability for specific contexts (for RE) and for

specific issues. While the existing codes of ethics lack specific context, the proposed framework development and evaluation using empirical data for specific situations confirms that the proposed framework aligns with industry needs. Unlike the IEEE-CS/ACM JTFSECEPP, which focuses on broad perspectives of software engineering ethics, this research proposed and validated its framework through empirical evidence, reinforcing its practical applicability for daily use in the real working environments of software industries.

Additionally, this research builds upon IEEE-CS/ACM JTFSECEPP and adapts it to problems found in the real-world working environments of software companies. Unlike some previous works , this research extracts the basic components of the proposed framework from the empirical data and proposes solutions to address the real-world problems of RE that reinforce the practical usability of the proposed framework.

Lurie and Mark [10] and Karim *et al.* [11]stress that ethical considerations should be embedded within software engineering, rather than treated as standalone concerns. This research advances their argument by demonstrating how a well-structured RE framework inherently supports ethical practices by ensuring transparency, clarity, and accountability in RE processes.

The proposed framework facilitates existing RE practices by providing a validated, structured, and context-specific guideline. The main new contributions of this research include

- A RE ethics framework built and validated empirically, ensuring practical feasibility.
- Identification of areas for refinement like C4EU/D and C1EKG, leading to framework improvements.
- Integration of participants' feedback into framework adjustments, improving clarity and applicability.
- Alignment with the well-known codes of ethics, IEEE-CS/ACM JTFSECEPP, reinforcing the importance of professional responsibility in RE.

The research findings contribute to both academic literature and industry practice by bridging the gap observed in software engineering codes of ethics, supporting arguments by both Lurie and Shlomo [10], and Karim *et al*. [11] that software development must incorporate ethical and professional standards as an integral part of software engineers.

6.6.4 Key Takeaways

- The proposed framework is empirically validated and shows strong face, content, and construct validity.
- Some components of the proposed framework (E.g., C4EU/D, C1EKG) require further refinement, emphasizing the need for iterative improvements.
- The research adds practical value by validating an RE framework that fits with industry needs.
- Future work should focus on refining weak areas and expanding validations.

By bringing into line with the ethical and professional standards discussed in Lurie and Mark [10] and Karim *et al.* [11], this research highlights the importance of integrating ethics, validation, and practical usability in RE frameworks, ensuring that they meet both methodological and professional standards.

6.7 Threat to Validity (Limitations of the Research)

While the research follows scientific ways to carry out the major research processes indicated in Figure 1.1, limitations should be recognized. The major limitations of the research concerning research approach, data collection, analysis, and proposed framework evaluation are discussed below.

6.7.1 Research Approach

Even though using the inductive approach provides benefits, some limitations were observed in this research. One of the main problems in using this approach was, this approach needs more time and it was time and resource intensive. Data collection and analysis took lots of time. The other difficulty in this approach is its subjectivity since inductive procedures hinge on themes from the data being discovered and what

preconceived ideas or predispositions the researchers themselves hold will influence theme finding. To resolve this problem, we tried to use different techniques like member checking, expert reviews, and peer reviews.

6.7.2 Data Collection

Employing interviews for data collection assisted us in getting in-depth information from the case companies. However, using interviews may result in interviewer bias, response bias, misinterpretations of questions, and so on. As a complementary method, we used FGDs to understand the problems via dialogues between participants. The use of FGDs helped us to cross-check the data collected through interviews. In addition to that, we used semi-structured interview questions that led to asking probing questions for more clarification regarding the problem under investigation.

The other threat to validity regarding data collection is that we didn't use triangulation using different techniques and different data sources. To fill this, we used techniques including giving the analysis results to the participants for reviewing, peer reviewing, iterative and intensive reviewing, refining the collected data, etc.

The other possible threat to the validity of the research is that the case companies are located in one country. The case companies from whom data were collected are all from Ethiopia. This may bring the question of using the framework only within the country. However, to address this issue and to improve its acceptability and representativeness, the proposed framework is evaluated by both international (about twenty-one) experts and experts from the country. The research findings and the proposed framework were communicated internationally and were accepted.

6.7.3 Analysis

Thematic analysis helps us to understand the study area and make sense of qualitative data that reflects participants' experiences. However, there are some threats to validity associated with this analysis method. One of them is its subjective nature since diverse researchers might understand and interpret the same data in different ways. This leads to possible discrepancies and biases in coding.

To address this potential problem, we followed a six-stage coding process, and we applied multiple coding strategies, including holistic, initial, in vivo, and attribute coding. At each stage of the analysis processes, many recursive revisions and refinements were made.

6.7.4 Evaluation of the Proposed Framework

One of the merits of the proposed framework is that it is derived from thematic analysis results, which provide a clear and organized view of the identified problems and their possible solutions. Expert evaluation increases validity in that the framework is based on known knowledge. Also, using Likert scale questionnaires gives quantitative validation so that expert feedback can be measured objectively. This puts expert opinion into quantifiable terms, providing measurable evidence of the relevance of the framework. Nevertheless, using expert evaluation may result in biased evaluation results. To minimize this, different sets of experts from different places were involved. In addition, FGDs were also employed. Besides, kinds of workshop discussions are organized and conducted on different days to give a clear explanation regarding the evaluation process and associated basic concepts.

Chapter Seven - Conclusions and Future Works

7.1 Conclusion

RE is one of the most crucial steps in developing effective software services and products. Gathering and analyzing requirements is essential to developing a successful software system. However, this stage is challenging and prone to numerous ethical wrongdoings. Considering ethics starting from the beginning is essential to get helpful software artifacts, services, and products and diminish unethical practices in the course of RE.

A wide range of aspects that are important to software engineers, with different dimensions is covered by the IEEE-CS/ACM JTFSECEPP, which is the most popular code of ethics, for instance [9]. This code of ethics provides a wide-ranging standard that states how software engineers should act concerning various aspects including the public, products, the profession itself, co-workers, oneself, and others. Everyday adherence to and application of the IEEE-CS/ACM JTFSECEPP is found to be difficult. To resolve the problem, Karim and others mapped the phases of the systems development lifecycle to the four IEEE-CS/ACM JTFSECEPP [6]. Still, there haven't been any significant modifications to those ethical principles for daily uses in the specific domain.

In this research, we looked into ethical concerns underlying RE. The research assessed the importance of ethical considerations, the procedures employed by the software industries, techniques for determining ethical dilemmas, and basic ethical challenges in RE. The investigation was carried out under actual software industry conditions. The research covered seven software industries [12].

We used two data collection methods. These are focus groups and interviews. Purposive sampling is used to select companies and specific individuals from each company. Qualitative data analysis methods, such as initial, holistic, in vivo, and attribute coding strategies were employed to examine the gathered data [12].

The results of the research show that companies lack professional responsibility codes of conduct, industry norms, and other policies that would allow them to incorporate

ethical principles that aid in minimizing RE ethical concerns. Additionally, almost all companies lacked means to identify and check whether ethical concerns are contemplated and included throughout RE [12].

To address the recognized main ethical problems, we have proposed a framework with five components for RE ethics. This framework aims to minimize the occurrence of major ethical issues during the RE processes [163].

The first component is the C1EKG, aimed at addressing issues encompassing unrealized benefits of automation, lack of experience, communication gaps, insufficient identification and utilization of tools and methodologies, and others. To tackle these problems, we propose sets of clauses as a resolution mechanism.

The second component, C2ERI, states sets of clauses to resolve ethical problems related to incomplete requirements, insufficient RE processes, ambiguous requirements, requirements omission, lack of attention to non-functional requirements, failure to identify "to-be" requirements, irrelevant requirements, and lack of proper documentation, among others. To tackle these concerns, we put forth detailed clauses associated with each of the corresponding concerns as specific guidance for software engineers.

The third component, C3ERQ, addresses major issues such as security, privacy, usability, reliability, consistency, and others. Like other components, here also we provided sets of pieces of advice that should be considered during RE to minimize their occurrences on requirements. The fourth component focuses on C4EU/D and offers guidance through pieces of advice as a solution to address the concerns within it. The fifth component, C5EPA, discusses bits of advice as a resolving mechanism for the issues, including lack of honesty, adding malicious requirements, carelessness, deceptive requirements, plagiarism (redundancy, duplication of works), and denial of agreements.

Results from the evaluation indicate that the proposed framework has the potential to be a practical guideline for software engineers regularly. This framework facilitates the creation belonging to useful software products by providing ethical guidance in

RE. Furthermore, it can be a foundation for further investigations into ethics in RE. However, the limitation of the research lies in obtaining a large number of participants, with only 40 willing participants, of whom nine did not return the questionnaire.

7.2 Contribution of the Research

Through thematic analysis, we identified and filtered out five major problems that should be taken into account during RE. To resolve these problems, the five components are identified. Using these, we proposed a framework for RE ethics. So, the main contribution of this research is the proposed framework, which will be essential for software engineers [163]. The proposed work will assist them by providing clues as to what and how ethical issues should be considered practically in RE activities.

Additionally, by analyzing data collected from interviews and FGDs, the research contributes practical insights into software company practices and challenges.

The research has made a stimulating contribution to academics, as evidenced by the popularity of views and engagements in the published papers. The papers have received enormous attention from scholars as a sign of their relevance and value addition to ongoing debates in the field. The research introduced new perspectives and served as a reference point for subsequent studies.

7.3 Future Works

The proposed framework consists of major components, each of which can be investigated independently. So, in the future, the researchers will assess the impacts of each of the components on the ethical decision-making processes of software engineers.

Besides, to ensure that the framework is relevant, effective, and responsive to the needs and concerns of all stakeholders, diverse experts from different phases of SDLC, including programmers, designers, and architects, will be engaged to gather feedback on the performance of the proposed ethical framework and identify areas for

improvement. This will help us to facilitate collaboration, build trust, enhance the legitimacy of the framework, and ultimately contribute to better ethical decision-making and outcomes.

Furthermore, going forward, we intend to enhance and thoroughly investigate the proposed framework, focusing on each step of the RE processes. This encompasses requirements elicitation, requirements analysis and negotiation, requirements specification, and requirements validation [56, 106].

Moreover, the growth of software technological innovations goes fast in different dimensions. This research sets the foundation for an ethical framework in RE that can be further extended to address ethical issues in various technologies. There are many dimensions in which this research will be expanded to Artificial intelligence (AI) and other emerging technologies. Primarily, we will focus on extending the proposed framework for AI technologies since there are growing ethical challenges and implications in AI [166, 167]. Some future works about this include:

- In the future, we will validate the proposed framework in real AI projects. We will conduct case studies and expert evaluations in real-world AI projects to enhance the applicability and effectiveness of the proposed framework.
- Moreover, bias, transparency, accountability, and fairness are serious ethical issues in AI-driven systems that require tailored ethical guidelines in the RE process [168, 169]. In the future, we will extend the research to integrate these concerns into the proposed framework.
- On the other hand, using AI, like machine learning models, can assist in ethical decision-making processes [170]. Therefore, in the future, we will investigate the role of AI in supporting ethical decision-making practices in RE to identify serious ethical problems of requirements. We will carry out different works, including algorithm design and others to integrate AI with the proposed framework.

References

1 Atkinson R. D. and Luke A., The Economic Benefits of Information and Communications, U.S, Washington, DC: Information Technology and Innovation Foundation (ITIF), 2013..

2 Dutta S., Geiger T., and Lanvin B., " ICTs for Inclusive Growth," Johnson Univ., the Global Information Technology Report, 2015.

3 Thakur D., "What is software? Characteristics and Classification of Software," 2017, June 22. [Online]. Available: https://ecomputernotes.com/software-engineering/characteristics-and-classification-of-software?utm_source=chatgpt.com. [Accessed 23 06 2021].

4 Zakaria N., Ibrahim S., and Mahrin M., "Examining Value-Based Factors in Software Development: A Survey Study in Malaysian Public Sector," *ACM ASWEC, Australia*, vol. 1, no. 15I, p. 13 – 17, October 01, 2015.

5 Biru T., "Reflective Steps: A Collaborative Learning Oriented Approach to Software Development and Process Improvement, Ph.D. Dissertation," Addis Ababa University, Addis Ababa, Ethiopia, October 2008.

6 Ammar F. A., "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, February 2016.

7 Boehm B., Biffl S., Aurum A., Erdogmus H. and Grünbacher P., Value-Based Software Engineering, Berlin Heidelberg: Springer-Verlag, 2006.

8 Richard B. W., Engineering ethics. In Challenges and Opportunities, Springer, 2014.

9 IEEE-CS/ACM Joint Task Force, "Software engineering code of ethics and professional practice," 2014. [Online]. Available: file:///C:/Users/Seble/Downloads/acm-se-code-of-ethics.html. [Accessed 13 July 2021].

10 Lurie Y., and Shlomo Mark, "Professional Ethics of Software Engineers: An Ethical Framework," *Science and Engineering Ethics,* vol. 22, no. 2, pp. 417-434, 2016.

11 Karim Abdul, Fahda Al Ammar; Romana Aziz, “Ethical Software: Integrating Code of Ethics into Software Development Life Cycle,” in *In Proceedings of the Computer and Applications (ICCA), International Conference*, Doha, United Arab Emirates, 2017.

12 Biable, E., Garcia, M., Midekso, D., and Pombo N., “Ethical Issues in Software Requirements Engineering,” *MDPI, Software*, vol. 1, p. 31–52, 2022.

13 Vlasicaug Bill, “Volkswagen Engineer Gets Prison in Diesel Cheating Case,” 25 August 2017. [Online]. Available: https://www.nytimes.com/2017/08/25/business/volkswagen-engineer-prison-diesel-cheating.html. [Accessed 26 August 2021].

14 Aberkane A., “Exploring Ethics in Requirements Engineering,” MSc. Thesis, Dept. of Computing Science, Utrecht Univ., Utrecht, Netherlands, 2018.

15 Pavan M., “Value Based Requirements Engineering: State-of-art and Survey" Master Thesis, Blekinge Institute of Technology, Faculty of Computing, Karlskrona, Swede, 2016.

16 Shah T. and Patel V, “A review of requirement engineering issues and challenges in various software development methods,” *International Journal of Computer* Applications, vol. 99, no. 15, pp. 26-45, 2014.

17 Bhat J., Gupta M., and Murthy S., “Overcoming Requirements Engineering Challenges: Lessons from Offshore Outsourcing,” *IEEE Computer Society*, vol. 23, no. 5, pp. 38-41, September/October 2006.

18 Eric R., Dekai X., Lorenz D., Alain K. and Olivier W., and Johannes J., “Challenges in applying large language models to requirements engineering tasks,” *Design Science*, vol. 10, no. e16., pp. 1-21, 10 (2024): e16.

19 Jain A., “Theory of Value Based Software Engineering," PhD Dissertation, University of Southern California, California, December 2007. https://bit.ly/4iVKSja.

20 Mohd N. and Sahibuddin S., “Critical success factors for software projects: A comparative study,” *Academic Journals*, vol. 6, no. 10, pp. 2174-2186,, 18 May 2011.

21 Hussain A., Emmanuel M. and Fazillah K., "The Role of Requirements in the Success or Failure of Software Projects," *International Review of Management and Marketing,* vol. 6, no. S7, pp. 306-311, 2016.

22 Tamburri D., Palomba F., and Rick K., "Success and Failure in Software Engineering: A Followup Systematic Literature Review," *IEEE Transactions On Engineering Management,* vol. 68, no. 2, pp. 599 - 611, April 2021.

23 Kotowaroo M., and Roopesh K., "Success and Failure Factors Affecting Software Development Projects from IT Professionals' Perspective," in *Soft Computing for Security Applications,* Advances in Intelligent Systems and Computing, 2022, pp. 757-772.

24 International Project Leadership Academy, "Advanced Leadership skills - Developing Project Management Capabilities," CALLEAM CONSULTING LTD, [Online]. Available: http://calleam.com/WTPF/?page_id=1445. [Accessed August 2022].

25 Bloch, Michael, Sven Blumberg, and Jürgen Laartz, "Delivering large-scale IT projects on time, on budget, and on value." Harvard Business Review 5.1, 2-7, October 2012. [Online]. Available: http://bit.ly/42fQPkn. [Accessed 05 August 2017].

26 Geneca, "Why Up to 75% of Software Projects Will Fail," 25 January 2017. [Online]. Available: https://www.geneca.com/blog/software-project-failure-business-development. [Accessed 05 August 2022].

27 Jrgensen, H., Owen L., and Neus A., " Making Change Work, Continuing the Enterprise of the Future Conversation," IBM Institute for Business Value (2008). [Online]. Available: https://bit.ly/4iXDpAd [Accessed 05 August 2022].

28 Gulla J., "Seven Reasons Why Information Technology Projects Fail," IBM Corp, August 11, 2011..

29 Hastie S. and Wojewoda S., "Standish Group 2015 Chaos Report," The Standish Group International, Inc. 2015. [Online]. Available: https://bit.ly/4iV9FDT. [Accessed 10 September 2021].

30 Doheir Y., Elzamly M., and Yaacob N., "Investigating the Role of Critical Success Factors in Achieving the Success of Agile Projects in the

Gaza Strip," *Journal of computing and information technology*, University of Zagreb Faculty of Electrical Engineering and Computing, vol. 30, no. 2, pp. 117-137, 2022.

31 Hoy Z. and Xu M., "Agile Software Requirements Engineering Challenges-Solutions—A Conceptual Framework from Systematic Literature Review," *Information*, MDPI, vol. 14, no. 6, pp. 1-19, 2023.

32 Capers J., "Software Requirements and the Ethics of Software Engineering," July 9, 2012.

33 Udayakumar, L., VV Sunder Singh Babu, and VV Sunder Singh Babu, "Manuel Kant's Deontology Theory," *JRAR-International Journal of Research and Analytical Reviews (*IJRAR*)* , vol. 8, no. 2, pp. 235-243, 2021.

34 Stuart M. Utilitarianism in Seven masterpieces of philosophy, Routledge, Toronto: Ryerson University, pp. 329-375, 2016.

35 Frede D., "Aristotle's Virtue Ethics," *The Routledge companion to virtue ethics. Routledge*, pp. 17-30, 2015.

36 Gotterbarn D., "Informatics and professional responsibility," *Science and* Engineering *Ethics*, vol. 7, no. 2, pp. 249-258, 2001.

37 Michael D., Thinking Like an Engineer: Studies in the Ethics of a Profession. Practical and Professional Ethics Series, USA: Oxford University Press, 1998.

38 Baase S., A gift of fire, Pearson Education Limited, 2012.

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

40 Flores F. and Silvio R., "Ethical software engineering: a critical review about software engineering in face of security requirements in the IoT/IoE society," in *In 2021 IEEE International Systems Conference (SysCon)*, Brazil, 2021.

41 Le Page Michael, "How did Volkswagen rig emissions tests?," *New Scientist*, vol. 11, 2015. vol.. 227, no. 3041, pp. 11, Oct. 2015. [Online]. Available: https://doi.org/10.1016/s0262-4079(15)31296-3. [Accessed 05 January 2022].

42 Boston W., Varnholt H., and Sloat S., “Volkswagen Blames ‘Chain of Mistakes’ for Emissions Scandal,” *The Wall Street Journal,* 2015.

43 Reitze Arnold W, “The Volkswagen air pollution emissions litigation,” 2016.

44 Nazanin M., “A Case Study of Volkswagen Unethical Practice in Diesel Emission Test,” International *Journal of Science and Engineering Applications,* vol. 5, no. 4, pp. 211-216, 2016.

45 Linda G., “Lost radio contact leaves pilots on their own,” *Journal of IEEE Spectrum,* vol. 41, no. 11, pp. 16-17, 2004.

46 Borrás C., “Overexposure of radiation therapy patients in Panama: problem recognition and follow-up measures,” *Revista Panamericana de Salud Pública,* vol. 20, no. 2-3, pp. 73-18, 2006.

47 Fitzgerald B., and Klaas J., “Continuous software engineering: A roadmap and agenda,” *The Journal of Systems and Software,* p. 1–14, 11 August 2015.

48 Aydemir, Fatma Basak, and Fabiano Dalpiaz, “A roadmap for ethics-aware software engineering,” in *In 2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, pp. 15-21. IEEE, 2018.

49 Ahmad K., Mohamed A., Chetan A., Muneera B., and John G., “Requirements practices and gaps when engineering human-centered Artificial Intelligence systems,” *Applied Soft* Computing, vol. 143, no. 110421, pp. 1-30, 25 January 2023.

50 Nguyen P., Babar M., and Verner J., “Trust in software outsourcing relationships: an analysis of Vietnamese practitioners,” in *In 10th International Conference on Evaluation and* Assessment *in Software Engineering (EASE),* Australia, 2006.

51 Hasan N., Hasan M. S., Mahmood A., and Alam J.,“A Model for Value Based Requirements engineering,” *International Journal of Computer Science and Network Security (IJCSNS)* , vol. 10, no. 12, pp. 171-176, December 2010.

52 Systems S. and Stephen M., “Enterprise Architect Requirements Engineering," Sparx Systems Version 1.0,” 2016. https://bit.ly/4hCd61g.

53 Vallor Shannon, Arvind Narayanan, Regnell B., Jones C., and Skipper R. B., “An Introduction to Software Engineering Ethics,” *Markkula Center for Applied Ethics*, pp. 1-60, 2015.

54 Li, Q., “Value-Based, Dependency-Aware Inspection and Test Prioritization, Ph.D.Dissertation California,” University of Southern California, Computer Science Department, California, December 2012.

55 Volkman R., “Computer ethics beyond mere compliance,” *Journal of Information Communication and Ethics in Society,* vol. 13, no. 4, pp. 176-189, 2015.

56 Phillip A., Requirements engineering for software and systems, 3rd ed. ed., Auerbach Publications, 2017.

57 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., 2001.

58 Wong W., Debroy V., Surampudi A., Kim H., and Siok Michael, “Recent Catastrophic Accidents: Investigating How Software Was Responsible,” in *in Proceeding of Fourth International Conference on Secure Software Integration and Reliability Improvement, IEEE*.

59 Berenbach B., “Professional and Ethical Dilemmas in Software Engineering,” J*ournal of Computer*, IEE, vol. 42, no. 1, pp. 74-80, 2009.

60 Fewell J., “The blame game,” *PM Network*, vol. 24, no. 7, p. 23, July 2010.

61 Hussain A., Emmanuel M., and Fazillah M., “The Role of Requirements in the Success or Failure of Software Projects,” *International Review of Management and Marketing*, vol. 6, no. 75, p. 306–311., 2016.

62 CNN, “TSA: Computer Glitch Led to Atlanta Airport Scare,” April 21, 2006.

63 Croomes, S., “Overview of the DART Mishap Investigation Results,” NASA Report, 1-10, May 2006.

64 Krebs B., “Cyber Incident Blamed for Nuclear Power Plant Shutdown,” June 5, 2008.

65 Boehm B., Biffl S., Aurum A., Erdogmus H. and Grünbacher P., “Value-Based Software Engineering: Overview and Agenda,” in *Value-Based*

Software Engineering , Berlin, Heidelberg, SpringerVerlag, 2006, p. 3 – 14.

66 Donald G., "Software engineering ethics," *Encyclopedia of Software Engineering,* vol. 2, pp. 1-13, 2002.

67 Gotterbarn D., and Miller, K. W.,"The public is the priority: Making decisions using the software engineering code of ethics," *Computer,* vol. 6, pp. 66-73, 2009.

68 Gotterbarn, D. W., Bo Brinkman, Catherine Flick, Michael S. Kirkpatrick, Keith Miller, Kate Vazansky, and Marty J. Wolf, "ACM code of ethics and professional conduct," 2018. https://www.acm.org/code-of-ethics?utm_source=chatgpt.com.

69 Berenbach B. and Broy M., "Professional and Ethical Dilemmas in Software Engineering," Journal *of Computer, IEE,* vol. 42, no. 1, pp. 74-80, 2009.

70 Evan C., "Ethics in Requirements Engineering," 20 July 2017.

71 Spiekermann S., "What to Expect From IEEE 7000: The First Standard for Building Ethical Systems. IEEE Technology and Society Magazine," *IEEE Technology and Society Magazine,* pp. 99-100, 2 September 2021.

72 Friedman B., and David G., Value Sensitive Design: Shaping Technology with Moral Imagination, Washington, United States: MIT Press, September 2018.

73 Creswell J., Qualitative Inquiry and Research Design: Choosing Among Five Approaches, 2nd ed., New Delhi: Sage publications, 2016.

74 Bryman A., Social Research Methods, 5th ed., United Kingdom: Oxford University Press, 2016.

75 Braun V., "Using thematic analysis in psychology," *Qualitative research in psychology,* vol. 3, no. 2, pp. 77-10, 2006.

76 Clarke V., Thematic analysis: A practical guide, pp. 1-100. 2021.

77 Hernamdez B., "Naturalistic inquir," in *The handbook of Social Work research method,* SAGE, 2010, pp. 415-425.

78 Sarah J. Qualitative research methods: Collecting evidence, crafting analysis, communicating impact, UK: John Wiley & Sons, 2024.

79 Athens L., “Naturalistic Inquiry in Theory and Practice,” *Journal of Contemporary Ethnography,* vol. 39, no. 1, pp. 87-125, 2010.

80 Patton M., “Integrating Theory and Practice,” in *Qualitative Research and Evaluation Methods*, USA, SAGE Publications, 2014 M10 29 - 832 pages, 2014.

81 Tongco M., “ Purposive sampling as a tool for informant selection,” *Journal of Ethno Botany* Research *and Applications,* vol. 5, pp. 147-158, 2007.

82 Trace C., “Applying content analysis to case study data: A preliminary report,” 2005.

83 Ishak M., and Bakar A., “Developing sampling frame for case study: challenges and conditions,” *World Journal of Education,* vol. 4, no. 3, pp. 29-35, 2014.

84 Aman M., “Qualitative data analysis: An overview of data reduction, data display, and interpretation,” *Research on Humanities and Social Sciences,* vol. 10, no. 21, pp. 15-27, 2020.

85 Yin R., “Qualitative Research from Start to Finish”, 2nd ed. New York, NY, USA: Guilford Press, 2015.

86 Louis Cohen Lawrence Manion, Research methods in education, routledge, 2002.

87 Saldaña J., “The coding Manual for Qualitative Researchers,” Sage, 202, [Online]. Available: https://books.google.com.et/books?hl=en&lr=&id=RwcVEAAAQBAJ&oi=fnd&pg=P

88 Patton M. Q., Qualitative research and evaluation methods: Integrating theory and practice, Sage publications, 2014.

89 Punch, Keith F., and Alis Oancea, Introduction to research methods in education, Sage, 2014.

90 Taherdoost H., Validity and Reliability of the Research Instrument; How to Test the Validation of a Questionnaire/Survey in Research." August 10, 2016.

91 Vveinhardt J., “Structure of Research Design: Expert Evaluation." Chapters (2018).,” *in Management Culture and Corporate Social Responsibility*, Lithuania, April 2018, pp. 155-187.

92 Kusmaryono, I., Dyana W., and HevyRisqi M., “Number of Response Options, Reliability, Validity, and Potential Bias in the Use of the Likert Scale Education and Social Science Research: A Literature Review,” *International Journal of Educational Methodolog*, vol. 8, no. 4, pp. 625-637, 2022.

93 Amidei, J., Paul P., and Alistair W., “The use of rating and Likert Scales in Natural Language Generation human evaluation tasks: A review and some recommendations,” 2019.

94 Mishra, S., and Shashi A., Handbook of research methodology, 2002.

95 Maness M., Parvathy S., Suryaprasanna B., and Abdul P., “When neutral responses on a Likert Scale do not mean opinion neutrality: Accounting for unsure responses in a hybrid choice modeling framework,” *In 7th Innovations in Travel Modeling Conference*, Atlanta, GA, USA, 2018.

96 Lukianova, V., Yuliya S., and Polozova V., “Expert assessment method in socio-economic research and Scales transformations,” *In 2019 7th International Conference on Modeling, Development and Strategic Management of Economic System (MDSMES 2019)*, Atlantis, 2019.

97 Mohagheghi P., An approach for empirical evaluation of model-driven engineering in multiple dimensions, in *Proceedings of the 5th C2M: EEMDD Workshop at ECMFA'10*, 2010, pp. 6–17. 2011. https://www.sintef.no/globalassets/upload/ikt/9012/ecmfa10-c2m-evaluating-mde-final.pdf

98 Anatoly E., Ivan Z., Andrey A., and Alexander M., “Application of expert evaluation method for realization of tasks in construction industry,” *In Web of Conferences*, 2019.

99 Akyildiz, S., and Kwestan H., “An overview of qualitative research and focus group discussion,” *International Journal of Academic Research in Education*, vol. 7, no. 1, pp. 1-15, 2021.

100 Cortini, M., Teresa G., and Stefania F., “Focus group discussion: How many participants in a group?,” *Encyclopaideia* , vol. 23, no. 54, pp. 29-43, 2019.

101 Campbell R., Guide To Focus Group Discussions, Usaid, December, 2008.

102 Ian S., Software Engineering, 9th ed., New York San, Francisco: Addison-Wesley, 2011.

103 Collopy P. D. and Hollingsworth P. M., “Value-Driven Design,” *Journal of Aircraft*, vol. 48, no. 3, p. 749 – 759, May–June 2011.

104 Duboc L., Birgit P., Jari P., Sedef A., Stefanie B., Ruzanna C., Ola L., Norbert S., and Colin C., “Requirements engineering for sustainability: an awareness framework for designing software systems for a better tomorrow,” *Requirements Engineering*, vol. 25, pp. 469-492, 2020.

105 Asghar S., and Mahrukh U., “Requirement engineering challenges in development of software applications and selection of customer-off-the-shelf (COTS) components,” *International Journal of Software Engineering*, vol. 1, no. 1, pp. 32-50., 2010.

106 Darwish R. and Megahed S., “Requirements Engineering in Scrum Framework,” *Journal of Requirements Engineering*, vol. 149, no. 8, pp. 24-29, 2016.

107 Pressman R. S., Software engineering: a practitioner's approach, Palgrave Macmillan, 2005.

108 Babar M., Ghazali M., Jawawi D., and Zaheer K., “StakeMeter: Value-Based stakeholder identification and quantification framework for value-based software systems,” *PloS one*, vol. 10, no. 3, e0121344, pp. 1-33, 2015.

109 Wnuk K., and Pavan M., “Value-Based Requirements Engineering: Challenges and Opportunities,” *Engineering Software Systems: Research and Praxis*, pp. 20-33, 2019.

110 Sharma, V. B., Saurabh T., Susham B., Bharat L., Umakant D., Deepak D., Ashutosh S., and Jae P., “Recent advancements in AI-enabled smart electronics packaging for structural health monitoring,” *Metals 11, no. 10 (2021): 1537*, vol. 11, no. 10, 2021: 1537.

111 Pillai Dhanup S., Veronika S., and Amal K. “A comprehensive review on building integrated photovoltaic systems: Emphasis to technological advancements, outdoor testing, and predictive maintenance,” *Renewable and Sustainable Energy Reviews,* vol. 156, 2022: 111946.

112 Pillai Dhanup S., Veronika S., and Amal K., “Competence advancement supported by social media,” in *Proceedings of the TEN Competence Special Technology Track on Technology Support for Self-Organised Learners,* 2008.

113 Cambridge Advanced Learners Dictionary and Thesaurus, “Framework,” Cambridge University Press, [Online]. Available: https://dictionary.cambridge.org/dictionary/english/framework. [Accessed 12 February 2022].

114 Oxford Advanced Learners Dictionary, “Framework,” Oxford University Press, [Online]. Available: https://www.oxfordlearnersdictionaries.com/definition/english/framework?q=framework. [Accessed 28 November 2020].

115 Allied Health, “Framework, Seventh Edition, Encyclopedia,” Miller Kean, 2023. [Online]. Available: https://medical-dictionary.thefreedictionary.com/framework. [Accessed 28 November 2020].

116 Christensson P., “Framework Definition. Tech. Terms,” 7 March 2013. [Online]. Available: https://techterms.com/definition/framework. [Accessed 28 November 2020].

117 Schneider D. a. M. T., “A safety engineering framework for open adaptive systems,” in *In 2011 IEEE Fifth International Conference on Self-Adaptive and Self-Organizing Systems,* 2011.

118 Francois Mouton, Mercia M. Malan, Louise Leenen, and Hein S. Venter, “Social Engineering Attack Framework,” in *In 2014 Information Security for South Africa,* Pretoria, South Africa, 2014.

119 Mayer Nicolas, André Rifaut, and Eric Dubois, “Towards a Risk-Based Security Requirements Engineering Framework,” in *Proceedings of the 11th International Workshop on Requirements Engineering: Foundation for Software Quality (REFSQ'05),* in conjunction with the 17th

Conference on Advanced Information Systems Engineering (CAiSE'05), Porto, Portugal: Springer, pp. 89–104, 2005.

120 McKinlay A., Russell E., Patricia H., Gérard F., Ruxandra G., Patrick C., and Christian S., "BioMOFs: 2010. Metal–Organic Frameworks For Biological and Medical Applications," in *Angewandte Chemie International Edition*, 2010.

121 Lin W., Wanchun D., Zuojian Z., and Chang L., "A Cloud-Based Framework for Home-Diagnosis Service Over Big Medical Data," *Journal of Systems and Software,* vol. 102, pp. 192-206, 2015.

122 Pangaro Louis, "Frameworks for Learner Assessment in Medicine: AMEE Guide," *Medical Teacher,* vol. 35, pp. e1197- e1210, 2013.

123 Schon, Eva-María, José Escalona, and Jörg Thomaschewski, "A Framework for Modeling and Improving Agile Requirements Engineering," PhD Thesis,," Computer Languages and Systems Department, University of Seville, 2017.

124 Arayici Yusuf, Vian Ahmed, and G. F. Aouad, "A requirements engineering framework for integrated systems development for the construction industry," *Journal of Information Technology in Construction, ITCon,* vol. 11, pp. 35-55, March, 2006.

125 Kabaale E., and Geoffrey M., "A theoretical framework for requirements engineering and process improvement in small and medium software companies," *Journal of Business Process Management,* vol. 21, no. 1, p. 80 – 99, January 2015.

126 Gupta A., and Aziz D., "A framework for software requirement ambiguity avoidance," *International Journal of Electrical & Computer Engineering,* vol. 9, no. 6, pp. 5436-5445, December 2019.

127 Bates R., "An ethical framework for engineering faculty: motivation, examples & discussion," in *In 2014 ASEE North Midwest Section Conference*, 2014.

128 Rehman, Shafiq Ur, and Volker Gruhn, "An effective security requirements engineering framework for cyber-physical systems," *Technologies,* vol. 6, no. 3, July 2018.

129 Van de Poel, “An ethical framework for evaluating experimental technology,” *Science and engineering ethics,* vol. 22, no. 3, pp. 667-686, June 2016.

130 Gayatri V., “Review of Frameworks in Requirements Engineering (RE) p. 101,” in *The International Conference on Software Engineering, Mobile Computing and Media Informatics (SEMCMI2015),* Mobile Computing and Media Informatics, 2015.

131 Partelow S., “What is a framework? Understanding their purpose, value,,” *Journal of Environmental Studies and Sciences,* vol. 13, p. 510–519, 2023.

132 Douglas C. Schmidt, “Frameworks: Why They Are Important and How to Apply Them Effectively,” *ACM Queue magazine,* vol. 2, no. 5, pp. 1-7, 2004.

133 Ernest M., “About software engineering frameworks and methodologies,” *In AFRICON, IEEE,* pp. 1-5., 2009.

134 David Thomas and Andrew, The Pragmatic Programmer: your journey to mastery, Addison-Wesley Professional, 2019.

135 Russ S., The Moral Status of Technical Artifacts: Fundamentals of Ethics, 6th ed., New York: Oxford University Press, 2012.

136 Hardy, Jörg, and George Rudebusch, Ancient ethics, Vandenhoeck & Ruprecht, 2014.

137 Lillie W., Introduction to Ethics, Allied Publishers, 2011.

138 Mastrian, K. G., Dee McGonigle, and Nedra Farcus, “Ethical applications of informatics,” in *Nursing informatics and the foundation of knowledge,* Jones & Bartlett Learning, LLC., 2012, pp. 69-88.

139 Jennifer D., A brief history of ethics. Ethics in Forensic Science, 2012, pp. 1-26.

140 Ozkaya I., “Ethics is a software design concern,” *IEEE Software,* vol. 36, no. 3, pp. 4-8, 2019.

141 Lynch W and Ronald K., “Engineering practice and engineering ethics,” *Journal of Science, technology, & human,* vol. 25, no. 2, pp. 195-225, 2000.

142 Génova G., Rosario M., and Anabel F., “Ethical Responsibility of the Software Engineer,” in *In PhiSE.*, 2006.

143 Gary P., Ethics and software development, IBM Corporation, 2006.

144 Harell G. and Daim T., “Virtual teams and the importance of building trust,” *IT professional,* vol. 11, no. 6, pp. 46-49, 2009.

145 Terrell B., Computer and information ethics, John Wiley and Sons, 2008.

146 Gotterbarn, D. W., Bo Brinkman, Catherine Flick, Michael S. Kirkpatrick, Keith Miller, Kate Vazansky, and Marty J. Wolf, “ACM code of ethics and professional conduct,” ACM**,** 2018.

147 Thomson Alan J., and Daniel L. Schmoldt, “Ethics In Computer Software Design and Development,” in *In the International Workshop on Software Fairness, Workshop of Computers and Electronics in Agriculture*, Sweden, 2001.

148 Gotterbarn Don, Miller Keith, and Rogerson Simon, “Software engineering code of ethics is approved,” *Communications of the ACM,* vol. 42, no. 10, pp. 1-9, 2003.

149 Gotterbarn D., Software engineering code of ethics and professional practice Computing Handbook, Third Edition: Computer Science and Software Engineering, 74-1, 3rd ed.

150 S. Keshav, “How to Read a Paper,” David R School of Computer Science, Canada, 2013.

151 Salehnia Ali, and Pournaghshband Hassan, “Ethical Issues in Software Engineering Revisited,” in *In Proceedings of Ethical Issues of Information Systems* , 2002.

152 Rashid Awais, Weckert John, and Lucas Richard, “Software engineering ethics in a digital world,” *Computer, IEEE,* vol. 42, no. 6, pp. 34-41, June 2009.

153 Thew S. and Sutcliffe A., “Value-based requirements engineering: method and experience,” *Requirements Engineering, spring,* vol. 23, no. 4, pp. 443-464, June 2017.

154 Whittle, Jon, Maria Angela Ferrario, William Simm, and Waqar Hussain,, “A case for human values in software engineering,” *IEEE Software,* vol. 38, no. 1, pp. 106-113, 28 November 2019.

155 Cemiloglu, Deniz, Emily Arden-Close, Sarah Hodge, Theodoros Kostoulas, Raian Ali, and Maris Catania, “Towards Ethical Requirements for Addictive Technology: The Case of Online Gambling and Practice (REthics),” *In 2020 1st Workshop on Ethics in Requirements Engineering Research*, 2

156 Paech Barbara, and Kurt Schneider, “How Do Users Talk About Software? Searching for Common Ground,” *In 2020 1st Workshop on Ethics in Requirements Engineering Research and Practice (REthics)*, 2020.

157 Dawadi S., “Thematic analysis approach: A step by step guide for ELT research practitioners,” *Journal of NELTA*, vol. 25, no. no. 1-2, pp. 62-71, December 2020.

158 Chaparro A., Joseph R., Elizabeth H., and Anastasia D., “Checklists: A review of their origins, benefits, and current uses as a cognitive aid in medicine,” *Ergonomics in design*, vol. 27, no. 2, pp. 21-26, 2019.

159 Host M., and Per R., “Checklists for software engineering case study research,” in *First international symposium on empirical software engineering and measurement (ESEM 2007)*, 2007.

160 Ardis M., David B., Gregory W., Jeff O., Mark S., and Willem V., “Software Engineering 2014-Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering a Volume of the Computing Curricula Series." Joint Task Force on Computing Curricula I,” in *Proceedings 16th Conference on Software Engineering Education and Training*, 2014.

161 Keppler P., “Guide to the software engineering body of knowledge (SWEBOK): Version 3.0,” IEEE - Computer Society, 2014.

162 Shepherd I., “A conceptual framework for simulation in healthcare education,” Doctoral dissertation, Victoria University, Melbourne, Australia, https://vuir.vu.edu.au/35047/. 2017.

163 Biable E., Garcia M., and Midekso D. “Proposed ethical framework for software requirements engineering., June 2023. DOI:,” *Journal of IET Software*, 2023.. https://doi.org/10.1049/sfw2.12136.

164 Mohajan Ha., “Two criteria for good measurements in research: Validity and reliability 17, no. 4, pp. 2017,” *Economic Series,* pp. 59-82, 2017.

165 Taherdoost H., “Validity and reliability of the research instrument; how to test the validation of a questionnaire/survey in research questionnaire/survey in research," *Int. J. Acad. Res. Manag.*, vol. 5, no. 3, pp. 28–36, August 12016.

166 Du S., and Chunyan X., “Paradoxes of artificial intelligence in consumer markets: Ethical challenges and opportunities,” *Journal of Business Research,* vol. 129, no. 6, p. 961–974, May 2021.

167 Singhal A., Nikita N., Hasnaat T. and Vijay M.“Toward fairness, accountability, transparency, and ethics in AI for social media and health care: scoping review,” *JMIR Medical Informatics,* vol. 12, no. 1, p. e50048, 2024.

168 Casas R., Joan J, and Santi C., “Education ethical dilemmas and AI: from ethical design to artificial morality,” in *International Conference on Human- Computer Interaction,* Cham, 2021.

169 Alabi M. The authors are Manoj Boopathi Raj, “Ethical Implications of AI: Bias, Fairness, and Transparency,” *Journal of Engineering & Technology Advancements. Technol. Advancements,* vol. 4, no. 4, p. 13–25, 2024.

170 Sorathiya A. and Gouri G., “Towards extracting ethical concerns-related software requirements from app reviews,” in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering,* Sacramento, CA, USA, 2024.

171 The Royal Academy of Engineering, “The Challenges of Complex IT Projects, The report of a working group from The Royal Academy of Engineering and The British Computer Society,” British Computer Society, British, April 2004.

172 Zemont G., “Towards Value-Based Requirements Traceability", Master Thesis, University of DePaul, Chicago Illinois, March 2005.

173 Sutharshan A., “Human factors and cultural influences in implementing agile philosophy and agility in global software development Ph.D. Dissertation,” University of Edith Cowan Computing, Health and

Sciences Faculty, School of Computer and Security Services, Perth and Bunbury, Western Australia, July 2013.

174 Khan M. and Khan M., “A Review of Value-Based Software Engineering and its Impacts,” *International Journal of Advanced Science and Technology*, vol. 75, pp. 33-42, 2015.

175 Zhang X., Auriol G., Eres H., and Baron C., “A prescriptive approach to Qualify and Quantify Customer Value for Value Based Requirements Engineering,” *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 4, pp. 327-345, 2013.

176 Detweiler C. and Maaike H., “Value Stories: Putting Human Values into Requirements Engineering,” in *in REFSQ Workshops*, 2014.

177 Gotterbarn, D., Miller, K. and Rogerson, S. “Computer Society and ACM Approve Software Engineering Code of Ethics,” *Computer* , vol. 32, no. 10, pp. 84-88, 1999.

178 Runeson P., M. Host, A. Rainer, and B. Regnell, Case Study Research in Software Engineering., Hoboken, New Jersey: John Wiley & Sons, Inc., 2012.

179 Yin R., Case Study Methods, in "Complemetnary for Research in Education", 3rd ed. Camilli, editor. Washington, DC, USA: American Educational Research Association, January 2004.

180 Bent F., “Five Misunderstandings about Case-Study Research,” *Qualitative Inquiry*, vol. 12, no. 2, pp. 219-245, April 2006.

181 Thew S., Value based requirements engineering, United Kingdom: The University of Manchester, 2014.

182 Quigley M., Encyclopedia of information ethics and security, IGI global, Hershey, New York: Monash University, 2007.

183 Christel, M. and Kyo C., “Issues in requirements elicitation.,” Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, CMU/SEI-92-TR-012, January 1992.

184 Ali R. Nan J., Keith P., Sarah M., and John M., “The emerging requirement for digital addiction labels,” in *International working conference on requirements engineering, REFSQ 2015: Foundation for*

software quality, Essen, Germany, March 23-26, Proceedings 21, pp. 198-213. Springer International Publishing, Cham, 2015.

185 K. P. a. R. A. Alrobai Amen, "Digital addiction: a requirements engineering perspective," in *International working conference on requirements engineering: Foundation for software quality,* Cham, 2014.

186 Alrobai A., Keith P., and Raian A., "Reinforcement learning as a framework for ethical decision making," in *Workshops at the thirtieth AAAI conference on artificial intelligence,* 2016.

187 Sambala E., Sara C., and Lenore M., "Ubuntu as a framework for ethical decision making in Africa: Responding to epidemics," *Ethics & Behavior,* vol. 30, no. 1, pp. 1-13, 2020.

188 Zhang B., Veijalainen J., and 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),* vol. 1, p. 176–187, 2016.

189 Thakurta R., "A Value-Based Approach to Priorities Non-Functional Requirements during Software Project Development," *International Journal of Business Information Systems,* vol. 12, no. 4, pp. 363-382, 2013.

190 Timo G., Alexander P., and Julian N., "Ethics in the Software Development Process: from Codes," *Philosophy & Technology,* vol. 34, p. 1085–1108, 2021.

191 Laplante P. and Kassab M., "Kassab, Requirements Engineering for Software and Systems", 4th ed. Boca Raton, FL, USA: Auerbach Publication, Taylor & Francis Group, LLC, June 2022.

Annexes

Annex – I: Publications

1 Biable S.E., Garcia N.M., and Midekso D. “Proposed ethical framework for software requirements engineering,” *Journal of IET Software, John Wiley and Sons Ltd*, vol. 17, no. 4, 526 – 537. 2023. https://doi.org/10.1049/sfw2.12136.

2 Biable S.E., Garcia N.M., Midekso D., and Pombo, N. “Ethical Issues in Software Requirements Engineering,” *Journal of Software in MDPI*, vol. 1, no. 1, pp. 31–52, 2022. https://doi.org/10.3390/software1010003.

Annex – II: Data Gathering Instruments [12]

I Interview Questions

A. Script Prior to the Interview Session

Dear Respondents/Participants

We would like to thank you for being willing to participate in the interview sessions of our research. This interview is part of Ph.D. research for the partial fulfilment of Doctor of Science in Software Engineering, entitled "**Framework for Software RE Ethics**" The aim of this research is to investigate the ethical concerns that need to be considered explicitly during the RE processes of the software development lifecycle. As such, the research will develop a RE framework aimed towards enhancing integration of ethical concerns in software artifacts. In addition, the proposed framework will function as a guideline to help software developers ensure the explicit inclusion of ethical concerns for RE processes.

To obtain the valuable information needed to explore the facts and have the full picture of the study problem; obtaining the experiences, practices, skills, perceptions and insight of people who are currently working in software development is of paramount importance. Thus, we kindly request your assistance in responding to the following interview questions directly and honestly. Please be informed that all information provided will be kept confidential and no individual data will be reported, but your reply will help immensely in advancing our research in this area, and it will be used accordingly.

Once again, we thank you in advance for your cooperation.

With regards,

Researchers

B. Review of Consent Form Aspects and Background Notes

We want to ask you about your experience with RE processes. The research involves interviews with people doing software engineering and software development in different roles within different software companies.

As mentioned above the purpose here is to try to understand how software engineers consider ethical issues in different RE processes so that we can try to identify and propose solutions for the problems in this regard.

As you fill in the consent form of this research, we want to ask you the following questions and tape record your answers.

Are you still ok with me recording (or not) our conversation today? Yes/No

Can I switch on the recorder?

Participant's Code:

_____

C. Interview Questions

1 Can you please tell me, from your perspective, the importance of clearly considering ethical concerns during software RE processes?
2 In your company, in what manner is ethics being practiced in carrying out RE processes?
3 Which professional code of conduct are you following to identify ethical concerns for your day-to-day activities of RE processes?
4 Do you currently follow any other guidelines in place for considering ethical concerns throughout software development lifecycles?
5 During RE processes which fundamental ethical concerns are you considering?
6 What are the main factors that hinder you to identify ethical problems of requirements?
7 Would you please describe the methods you use to identify ethical concerns?

8 In your company who is involved in RE processes?

9 What kinds of conflicts of interest occur among different stakeholders of a software system during RE processes?

   a. How do you manage conflicts of interest that occur among different stakeholders of a software system during RE processes?

10 Would you please describe a time that you were challenged ethically during the journey of software RE processes?

11 Does your organization use any method to identify ethical concerns of RE? If yes, please provide the methods.

12 How do you check (make sure) whether requirements are not being held?

   a threat to human well-being (any negative adverts? or not?)

   b incomplete, inconsistent, inappropriate, imprecise, ambiguous/unclear requirements?

   c deceptive requirements and toxic requirements?

13 Would you please provide any general suggestions on how to best practice the integration of ethical concerns during RE processes?

About You

Would you tell me, please?

1 Your name _____

2 Name of your industry/company _____

3 What is your role in the industry? _____

4 What formal qualification (educational level) do you have, if any? _____

5 What is your field of study? _____

6 How long have you been working as Software Engineer/Developer? ____

II Focus Group Discussion Questions

1 What do you think about the concept of ethical concerns in RE processes?

2 In your company, in what manner is ethics being practiced eliciting, analyse, validate, and manage requirements?

  2.1 In your view point, how have the main ethical violations occurred during RE processes?

  2.2 What are those violations?

  2.3 From your perspective, how do we make sure that requirements are complete, consistent, clear (unambiguous), precise, and free from any deceptive and toxic requirements?

3 What methods have you used to identify ethical concerns for RE subprocesses?

4 What do you feel about the necessary ethical concerns, which should be clearly considered in each of the above RE subprocesses?

5 Which professional code of conduct standards will be more suitable to identify and consider ethical issues during software RE processes? Why?

Annex – III: Consent Form for the Research Interviews

Ethical Framework for Software Requirements Engineering

Principal Researcher: Name: *Seblewongel Esseynew, PhD candidate, Addis Ababa University.*

We appreciate your perusal of the information sheet regarding the sub-study on interviews. If you would like to take part, kindly fill out and sign the form below. To indicate that you concur with each of the following statements, please initial the boxes:

I have recognized purposes this research and the confidentiality conditions. I approve to be interviewed by Seblewongel Esseynew for the purpose of this research. I give my consent to the collection and use of my opinions, perceptions, information, and experience during this research.

Please Initial box:

I confirm that I have read and understood the information sheet ☐

I am aware that participation in this research is completely optional and that I can end it whenever I want, for any reason, and without facing any repercussions. ☐

I know that the information I provide will kept strictly private. I recognized that my identity will not be connected to the research materials, nor will it be appear in any report or reports that from the study. ☐

I accept a tape recording of this interview. I acknowledge that the audio recording of this interview will only be used for analysis and that excerpts from the conversation, from which I would remain anonymous, may be used in any research-based conference presentation, report, or journal article. I am aware that the original recording will not be accessible to anyone other than the study team, and that no further use of the recording will be done without my express written consent. ☐

I approve the use of my anonymized data for ongoing research, including publications linked to this research once it is finished. ☐

I agree to take part in this interview. ☐

_____	_____	_____
Name of participant	Date	Signature
_____	_____	
Email Address of participant	Phone Number of participant	
_____	_____	_____
Principal Investigator	Date	Signature

Annex – IV: Sample - Theming the Data -Fundamental Ethical Concerns

No.	Categories	Concerns (Codes)	Remark
1	**Knowledge Gap**	a . Unrealizing the benefit of automation b Frustration of automation c Lack of experience d Communication gap e The lack of knowledge to identify and utilize tools and methods f Weak Requirement change management	b → a (b is considered/included under concern a. The concerns are coded and put according to their frequency.
2	**Requirements Identification**	a Incomplete requirements b Insufficient requirements engineering processes c Presence of ambiguous requirements d Requirements Omitting e Lack of attention f Not identifying "to be" requirements g Collecting irrelevant requirements h Lack of proper documentation i Not identifying the real needs, and wants of customers (not identifying the real problems) j Unclear requirements k Vague requirements l Collecting Inappropriate requirements	The first eight are indicated in the proposed framework. Others encompassed within them. So, when we suggest pieces of advice, we consider all. Directly or indirectly, most of them are covered by others. For instance: i → f, a and g; j and k → c; l and n → g; o → e; and p → a. (**Notice:** x → y implies concern x is included under concern y)

No.	Categories	Concerns (Codes)	Remark
		mRequirements redundancy n Focusing only on functional requirements o Insufficient requirements p Unable to get holistic needs and wants	
3	**Quality Related Problems**	d Security concerns e Privacy concerns f Usability concerns g Performance issues h Cyber Attack i Lack of reliable services of systems j lack of consistency services of systems k Confidentiality issues	The first three were selected. Most of them are covered by the selected one. Example: cyber-attack → (a, it is part of Security); confidentiality → (b privacy) privacy, and lack of reliability and consistency→c) usability)
4	**Unwillingness**	d Unwilling to give clear, real, appropriate, and precise requirements e Getting too complex requirements f Be reluctant to give real/precise requirements (2 g Unwilling to accept the artifact (1 h Be reactant with changes (1) i Getting impure requirements (1)	The first three are explicitly put. Others included within them. For instance: d → a; e → c; and f → f → a and b.
5	**Doing forbidden action**	f Lack of honesty g Carelessness h Adding evil codes on source code i Plagiarism (duplications of works) j Denial of agreements	Five are considered and f is included within e (denial of agreements).

No.	Categories	Concerns (Codes)	Remark
		k Conflicts due to differences of interest	

Annex – V: *Sample Questionnaires used for Framework Validation*

What is your Gender?

- O Male
- O Female

How old are you?

_____

In which institution/organization are you working?

_____

What is your current position?

_____

How long have you worked in the software industry?

_____

After reading the rationale provided to you regarding this framework, please indicate to what degree you consider this framework to have the indicated validity and reliability.

1 Face validity of the Framework
   - O Very Good
   - O Good
   - O Neutral
   - O Poor
   - O Very Poor

2 Overall content validity of the framework
   - O Very Good
   - O Good

- O Neutral
- O Poor
- O Very Poor

3 Content validity of Component 1

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

4 Content validity of Component 2

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

5 Content validity of Component 3

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

6 Overall construct validity of the framework

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

7 Construct validity of Component 1

- O Very Good

- O Good
- O Neutral
- O Poor
- O Very Poor

8 Construct validity of Component 2

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

9 Reliability of the framework

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

10 Internal consistency

- O Very Good
- O Good
- O Neutral
- O Poor
- O Very Poor

Annex – VI: *Sample Questions Asked During Framework Evaluation*

1 Placement of Activities. After reading the proposed framework, please use "√" regarding your observation on the placements of clauses with respect to each concern.

- ☐ The placement is done appropriately and accurately without any mismatch.
- ☐ The placement is done appropriately and accurately with very little modification.
- ☐ The placement is done appropriately with some modification.
- ☐ The placement is not done appropriately. It should be done again.

2 Which are perceived to be the most challenging to be followed?

3 From your observation, which is perceived as important?

4 Please also provide any comments and recommendations.

_____

_____

_____

_____

_____

_____

Annex – VII: Declaration Sheet

The undersigned confirms that this dissertation work is my original work. During the journey of this research, I have followed the scientific, ethical, and technical conduct of research. The sources which are used in this dissertation are properly acknowledged, cited, and listed under the reference section. Moreover, I declare that I have not submitted this dissertation to any other academic institution for the purpose of receiving a degree or for any other mission.

_____

Seblewongel Esseynew Biable

As an advisor and co-advisor, we confirm to the best of our knowledge

_____ _____

Professor Nuno M. Garcia Dr. Dida Midekso (Associate Professor)