DEPARTMENT OF
COMPUTER SCIENCE

NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

INÊS RAQUEL DE ALMEIDA LOPES NUNES

BSc in Computer Science

A CONCEPTUAL MODEL FOR GENDER-INCLUSIVE REQUIREMENTS

MASTER IN COMPUTER SCIENCE AND INFORMATICS ENGINEERING

NOVA University Lisbon
November, 2021

DEPARTMENT OF
COMPUTER SCIENCE

A CONCEPTUAL MODEL FOR GENDER-INCLUSIVE REQUIREMENTS

INÊS RAQUEL DE ALMEIDA LOPES NUNES

BSc in Computer Science

Adviser: Ana Moreira
Associate Professor with Habilitation, NOVA University Lisbon

Co-adviser: João Araujo
Associate Professor, NOVA University Lisbon

Examination Committee

Chair: Dr. Pedro Barahona
Full Professor, NOVA University Lisbon

Rapporteur: Dr. Carla Silva
Associate Professor, Universidade Federal de Pernambuco

Member: Dr. Ana Moreira
Associate Professor with Habilitation, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE AND INFORMATICS ENGINEERING

NOVA University Lisbon
November, 2021

A Conceptual Model for Gender-Inclusive Requirements

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.9.4) [100].

To my grandparents, Alzira Alves de Almeida Lopes and António Lopes, thank you. This dissertation is yours.

First and foremost, I would like to thank my advisors, Professor Ana Moreira and Professor João Araujo, for all the guidance, availability, and dedication that allowed this work to be possible in these different times. I am deeply grateful for your encouragement and kind words that gave me the confidence to pursue this research, the knowledge you shared with me, and your consistent support that made this dissertation a sincerely fulfilling work. It was a true pleasure and privilege to work with you.

I would also like to thank the participants who took their valuable time to contribute with their interest and knowledge in the evaluation of this work.

A special thanks to André Pereira, for your genuine patience, understanding, and optimism, even in the most challenging times. For always being there to listen and discuss ideas, to have fun and laugh, and for doing everything you could to care for me. Thank you for the beautiful moments, your unconditional love, and for always believing in me throughout this journey.

Finally, this work would not have been possible without my family. To my mother, for the open-minded education where I had the freedom to explore and stay curious, for always encouraging me to pursue my studies, and for teaching me to be strong but kind. Thank you for listening to my ideas and discussions about this work, sometimes several times a day, which, without a doubt, were fundamental for this accomplishment. To my father, for showing me the importance of continuous learning and knowledge acquisition, and for always being a firm believer and supporter of my work and success. Last but not the least, thank you to my sister Filipa, for always being there, for your unfailing support and wise advice, for sharing with me your brilliant and thoughtful ideas, and your joy of living that has led us to great adventures and fun together. You inspire me every day.

Abstract

Gender equality is a fundamental human right. Empowering all women and girls means empowering half of the world's population, and therefore it is essential to create a peaceful and sustainable future. However, there is still a great deal to be done to achieve full equality of rights and opportunities for everyone. Gender inequality persists and prevents the development and thriving of individuals and societies. Such is the case of girls' and women's under-representation in information, communication, and technology, where only 3% of the graduates worldwide are women. Consequently, technology development holds serious problems of inclusion and diversity. As technologies rapidly evolve and revolutionize the way we live, missing diverse perspectives during development produces a gender-biased technology that, instead of advancing gender equality, creates new barriers in achieving it.

Technology can play a fundamental role in progressing gender equality and ensuring gender inclusion. Although considered neutral, the software does not equally serve everyone who depends on it. Software systems favor characteristics that are statistically more observed in majorities, ignoring or even attacking certain minorities. Concerning gender, existing systems favor characteristics that are statistically more observed in men over characteristics observed in other genders (e.g., trans, cis women, non-binary). The goal of this dissertation is to create awareness that the problem of non-inclusive software is real and investigate how to develop software that benefits everyone, regardless of their gender. As a first step, we performed a systematic mapping study to gather a comprehensive overview of the state-of-the-art on gender issues in software engineering.

This study served as the groundwork for the development of a conceptual model for gender-inclusive requirements. The model aims to support requirements engineers by providing a representation of gender domain knowledge that can be used as a resource for eliciting gender-inclusive requirements. To integrate the conceptual model into the existing practices of requirements engineers, we propose a framework that offers a set of guidelines with concrete goals to ensure a focus on gender inclusion from the earliest phases and throughout the development process. We applied the framework to develop a gender-inclusive prototype tool that supports the process of using the framework. The developed model was empirically evaluated by a group of 31 participants and the results were positive.

The conceptual model was published in a CORE A conference.

Keywords: gender issues, gender-inclusion, requirements engineering, software engineering, software development

Resumo

A igualdade de género é um direito humano fundamental. Empoderar todas as mulheres e raparigas significa empoderar metade da população do mundo, e como tal, é crucial para a criação de um futuro pacífico e sustentável. No entanto, há ainda um longo percurso a percorrer na conquista da total igualdade de direitos e oportunidades. A desigualdade de género ainda persiste, impedindo o desenvolvimento e prosperação de indivíduos e sociedades. Este é o caso da área da informação, comunicação e tecnologia que tem apenas 3% graduadas em todo o mundo. Consequentemente, a tecnologia apresenta sérios problemas de inclusão e diversidade, desde o seu desenvolvimento à sua adoção e utilização.

A tecnologia, e o software em particular, desempenham um papel cada vez mais central na vida de todos nós. Apesar de considerado neutro, o software não serve da mesma forma todos os que dependem dele. De facto, os sistemas de software privilegiam características estatisticamente mais observadas nas maiorias, ignorando ou mesmo atacando certas minorias. No que diz respeito ao género, os sistemas existentes privilegiam características estatisticamente mais observadas nos homens em detrimento das características observadas noutros géneros (por exemplo, trans, mulheres cis, pessoas não-binárias).

O objetivo desta dissertação e tomar consciência que o problema de software não-inclusivo é real e investigar como desenvolver software que beneficie todos os utilizadores, independentemente do seu género. Como primeiro passo, realizámos um mapeamento sistematico de literatura de modo a construir uma visão integrada sobre o estado da arte em questões de género em engenharia de software.

Os resultados deste estudo serviram de base para o desenvolvimento de um modelo conceptual para requisitos inclusivos de genero. Este modelo visa apoiar a elicitacao de informacao inclusiva providenciando uma representacao do conhecimento dominio de genero como recurso para a formulacao de requisitos inclusivos. Para integrar o modelo conceptual nas praticas da engenharia de requisitos, desenvolvemos uma framework que oferece um conjunto de diretrizes com objetivos concretos para garantir um foco na inclusao de genero desde as fases iniciais e ao longo do processo de desenvolvimento. Aplicamos a framework na construcao de um prototipo de uma ferramenta inclusiva de genero que apoia o seu processo de uso. O modelo desenvolvido foi avaliado empiricamente por um grupo de 31 participantes e os resultados foram positivos.

O modelo conceptual foi publicado numa conferência CORE A.

Palavras-chave: questões de género, inclusão de género, engenharia de requisitos, engenharia de software, desenvolvimento de software

Contents

List of Figures

List of Tables

1 Introduction

The present research work was motivated by the need to develop gender-inclusive software that supports and benefits everyone, regardless of gender. As technologies revolutionize the way we live, overlooking gender perspectives in the development of digital solutions results in gender-biased technology that, instead of advancing gender inclusion, creates new barriers in achieving it. Gender issues should be addressed in the initial stages of software development, namely during the requirements elicitation phase, to avoid perpetuating gender bias throughout the development and to the final product. This introductory chapter provides an overview of the context and motivation of this dissertation, defines the problem statement and the associated challenges, and presents the objectives along with the contributions.

1.1 Context and motivation

Gender issues refers to “all aspects and concerns related to women’s and men’s lives and situation in society, to the way they interrelate, their differences in access to and use of resources, their activities, and how they react to changes, interventions and policies” [39].

Over the last decades, research and measurement of various areas of life by gender made it possible to understand where and when gender issues occur specifically, and its impact on people’s lives. More than ever before, awareness of existing gender issues has grown, contributing to significant progress for equality of rights, responsibilities and opportunities. However, gender inequality persists overall, preventing social and individual development. Systemic issues such as "legal discrimination, unfair social norms and attitudes, decision-making on sexual and reproductive issues and low levels of political participation", remain a reality faced by women around the world [146].

The Universal Declaration of Human Rights sets out that every human being is born free and entitled to dignity and equal rights, without any form of discrimination [175]. Although most countries approach equality in public and private life through laws and policies in their constitutions, intersecting forms of discrimination are being perpetuated, directly and indirectly, by social norms, practices and gender-based stereotypes [91]. Such is the case of girls' and women's under-representation in STEM[1]. While there have been notable improvements in access to education globally, only about 30% of female students choose to pursue STEM related studies and careers. The low interest and participation in STEM is still present mainly because of individual, family, institutional and societal factors, which influence women and girls' perception of these fields and creates obstacles in completing and benefiting from STEM education. The most evident gender differences in STEM disciplines prevail in digital skills and ICT enrollment, where only 3% of the graduates worldwide are female [174].

Consequently, women hold just 24% of all digital sector jobs, making it more likely for inequalities in workplace culture, experience and policies to persist, as well as reinforcing existing gender bias that exclude them [101]. The lack of diversity and constrained decision-making power in the ICT sector also results in serious problems of inclusion and representation in technology production. Overlooking female perspectives, consciously or unconsciously, in the development of digital solutions does not produce gender-neutral technology, but rather gender-biased technology. Spreading harmful beliefs and notions across software services and applications, AI and digital assistants, and the data and algorithms that power these, creates new barriers in achieving equality for women and other consistently neglected groups, such as people of color, persons with disabilities and gender and sexual minorities [151].

The Beijing Declaration and Platform for Action of 1995 was a major landmark for the global agenda for gender equality by asserting that women's rights are human rights [33]. It sets concrete strategies and objectives that ensure respect for those rights and presents specific and significant actions to be taken for women's empowerment. In 2015, 17 ?? were established in the 2030 Agenda for Sustainable Development, a plan of action adopted by all United Nations Member States, that seeks universal peace and prosperity for people and the planet. Following with the commitment of the Beijing Conference on Women, Goal 5 of the 17 ?? aims to "Achieve gender equality and empower all women and girls", and defines 6 different, but interrelated targets to end all forms of discrimination, violence and harmful practices against women and girls everywhere, and to simultaneously recognize the care and domestic work, and ensure equal access to quality education, sexual and reproductive health, economic resources and political participation.

Furthermore, it highlights the pivotal role digital technology plays in supporting and advancing gender equality with the target "Enhance the use of enabling technology, in particular information and communications technology, to promote the empowerment of women" [172]. Rightly used, technology holds the potential to empower women and girls by providing a network for safe communication, facilitating socio-economic and political participation, and increasing their opportunities of accessing valuable education and careers. Equality in digital access and skills means girls and women having an active and meaningful participation in society, as digital transformation continues to reach every

[1] Science, technology, engineering and mathematics

sector of it.

However, the consequences of the existent gender imbalance and exclusion in the development process for the technology created and for its users is still an emerging topic. Recent research has shown that specific individual differences in the use of technology can be clustered by gender and that technology is privileging characteristics statistically associated with men. Overlooking diverse perspective during development creates biased and noninclusive technology that prevents women and girls from being equipped with the tools to identify and solve the most critical problems of our time, even if they possess the ideas to solve them. Given the impact and importance of technology for societies' successful development and its potential for advancing gender equality, software technologies must integrate different perspectives and accommodate diverse experiences to grant equal enjoyment of its benefits to all users. **Technology plays a fundamental role in shaping cultural and social beliefs and attitudes. Hence, there is a need for the development of approaches that help and guide the creation of inclusive technology.** This thesis is motivated by this statement. In particular, we focus on the problem of reducing gender issues from the initial stages of software development (i.e., Requirements Engineering) to overcome the gender gap and improve gender inclusiveness in software.

1.2 Problem statement

Our problem statement and objective definition result from the main findings of a systematic mapping study. We defined the problem statement as the **lack of approaches to address gender issues** in the early stages of software development, particularly in **Requirements Engineering**, and the wrongful assumption that if gender is not discussed, then the **software will be gender-neutral**. The discussion that follows identifies and justifies the two main problems that comprise the problem statement, the challenges that arise from these problems, and a brief description of what we can do to solve them. Fig.1.1 depicts the relationships between the problem statement, its two major problems, and the challenges that need to be tackled to solve them. Finally, we present the objectives of this dissertation.

Problem 1: Lack of approaches to address gender issues in RE

Research showing gender differences in software use that are not accounted for has emerged recently, highlighting the need for software to be gender-inclusive. Based on the findings of a systematic mapping study, the topic is gaining significant attention: there is already a well known set of gender issues in software systems that need to be addressed to ensure all users can use it equally. Although some approaches have been proposed to address gender issues, such as GenderMag [26], GIF [77] or RULES [173], these deal with them in the design phase of development. Indeed, the existing work on gender in Requirements Engineering is still limited. We did not find an approach for integrating gender in

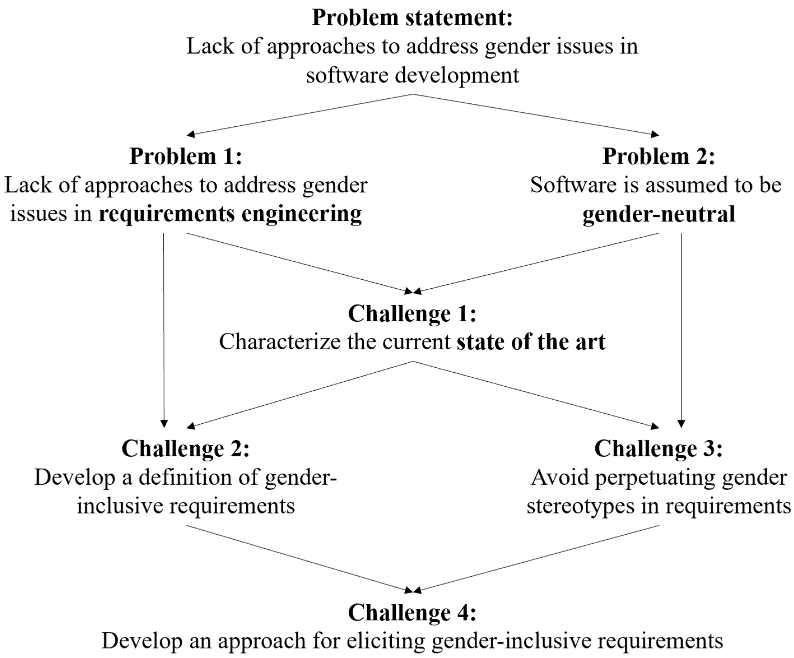


Figure 1.1: Problem statement, two main sub-problems, and respective challenges

requirements nor common definitions for gender and related concepts that can be used in already established practices of requirements engineers. The lack of accepted and shared definitions for such complex concept may result in misconceptions that can turn to non-inclusive, or even discriminating, requirements. Consequently, software design decisions will be based on gender-biased requirements, which negatively impacts users. Therefore, gender issues should be addressed in the early stages of software development to avoid propagating them to the next phases of development and ensure the system satisfies requirements that are gender-inclusive while following the software requirements process. This statement justifies the definition for our first problem.

Problem 2: Software is assumed to be gender-neutral

When developing software that is not gender-specific, it is assumed to be gender-neutral by the stakeholders [185]. Therefore, gender-based opinions are disregarded and potential gender differences are not considered during development. Because software is the result of decisions and interactions that incorporate specific social values and objectives in specific social contextual settings by those involved in the development process, it is neither neutral nor exclusively technical. Since stakeholders and individuals with decision-making power in the process are typically predominantly men, they integrate their own perspectives at the expense of others that are overlooked. This results in gender-biased software with no support for diversity. The identification and acknowledgment of gender issues is the first step towards overcoming them. Therefore, our second problem refers to the necessary clarification that a gender-aware approach should be adopted rather than a gender-neutral one to ensure equality of outcome. That is, by addressing the differences in the development process, the final software supports everyone, regardless of whether they have the same perspectives of the stakeholders or not.

Challenge 1: Characterize the state of the art

The first step to solve the above-stated problems is to survey the current state of the art on gender issues in software development. This challenge consists of answering what gender issues currently exist in software systems and what solutions have been proposed to solve them during the software development process with focus on the early stages. Additionally, it aims to survey how is the concept of gender understood in the software engineering field. Thus, we can identify the most relevant research works of this particular topic to understand what concepts or methodologies can help solve the following challenges. To ensure that we gather a comprehensive overview of the state of the art, we performed a systematic mapping study, presented in Chapter 3.

Challenge 2: Develop a definition of gender-inclusive requirements

The second challenge involves understanding what makes software gender-inclusive. Software is assumed to be mostly gender-neutral and the software engineering field and gender studies are seen as unrelated, and thus, gender is typically ignored or used as a categorical variable, resulting in gender issues. To ensure a gender perspective is integrated from the beginning and throughout the process, we need to understand what requirements must be met by the software system to satisfy gender-inclusiveness. Thus, we require a common definition of what entails the concept of gender-inclusive requirements, in what context and domains they can be applied, and what information is necessary for their formulation. First, it is crucial to acknowledge the existence of gender differences and then identify which differences impact the adoption, preferences, and use of the software. Additionally, we must investigate gender bias within the software itself and how it creates disadvantages for individuals who identify with the female and non-binary perspectives as they correspond to the underrepresented and marginalized groups in the software development process. The main focus is to understand what gender differences are not being addressed in software development that result in gender issues in software with the objective of integrating these differences in the definition of the gender-inclusive requirements. This definition must be sufficiently complete so everyone involved in the process can understand and discuss it, while being general enough so that it can be instantiated according to the domain of application of the system. To achieve this, we resorted to the findings of the systematic mapping study.

Challenge 3: Avoid perpetuating gender stereotypes in requirements

Gender issues can emerge during software development as a result of gender stereotypical assumptions by those present during development about prospective users. If these assumptions are not identified and discussed in the earliest stages of development, the final software could be unfairly discriminating against groups of individuals that are underrepresented in the process. As mentioned previously, the impact is disproportionately negative towards women. Therefore, an approach that aims to address gender issues in software development, specifically in Requirements Engineering, must allow stakeholders, developers, owners, and every individual with decision-making power to become aware of their gender-stereotyped ideas during elicitation. Simultaneously, it must provide methods to mitigate and avoid integrating them in the information that will be used for the formulation of the system's requirements. Hence, to address this challenge we need to gather the gender issues that currently exist, the potential issues that can occur, and develop a clear distinction between gendered and gender-inclusive requirements.

Challenge 4: Develop an approach for eliciting gender-inclusive requirements

The final challenge, then, is the development of an approach for eliciting gender-inclusive requirements. As discussed in the above challenges, its development considers the existing knowledge on gender issues as groundwork for the definition of gender-inclusive requirements. This definition must capture fundamental gender related concepts and their relations that impact software. After knowing what gender-inclusive requirements are, an approach is necessary for their elicitation. This approach must minimize the possibility of unclear, ill-defined requirements that may lead to gender stereotypes or discrimination against any gender. Then, it must ensure the elicited requirements include all gender perspectives and benefit all potential users, regardless of their gender. Furthermore, because gender is a social concept, this approach should also consider the social context of its application, prioritizing, if possible, requirements that promote inclusiveness and empowerment. Finally, it must be intuitive for everyone in the development team to use. In this dissertation, we fulfilled the first part of this challenge and took a first towards the second one.

1.3 Objectives and proposed solution

In this dissertation, we worked towards addressing the above stated problems by proposing a **Conceptual Model for Gender-Inclusive Requirements**. The first challenge was to understand what gender issues currently exist, what possible impact they can have in software development, and what are the consequences for the software itself and its users if they are not addressed. To achieve this, we conducted a systematic mapping study on gender issues in software engineering. The results presented in Chapter 3 indicated the existence of gender issues in software engineering. Yet, there were few approaches for mitigating them and mostly focused on the design of the system. Overall, research on gender in the context of software development and its impact on systems and users is well documented. In addition, gender issues arise early during development, and thus, should be addressed in the requirements engineering process to ensure gender-inclusive design decisions that will lead to software that supports all users equally. However, this knowledge is scattered across distinct resources and there is no common structured vocabulary for analyzing and addressing gender and related concepts in requirements. This gap motivated the collection of key gender-related concepts that would be both useful and practical for Requirements Engineering and could provide a starting point for analyzing requirements in light of gender-inclusive goals.

Then, we used the collected knowledge as the groundwork for developing a common definition for gender-inclusive requirements, with the aim of addressing both the second and third challenges. To this end, a conceptual model appears as the most suitable solution, as it provides a structured and concise description of a particular domain knowledge, usually in a standardized graphical notation, that supports the requirements engineering process [45]. Additionally, conceptual models enable the discussion and analysis of the knowledge they express by providing a common terminology and a shared understanding among individuals involved in the development process [120]. Thus, in this context, a conceptual model can express what and how gender concepts and their relationships should be considered and integrated in the requirements process. Furthermore, they facilitate the reuse of information and help clarify assumptions, goals, and decisions [120]. The objective is to provide a documented means of communication between all stakeholders that facilitates an informed and purposeful discussion and aligns the decisions made with the goal of developing gender-inclusive software. For these reasons, we proposed a well-defined conceptual model that captures key gender concepts and relations, presented in Chapter 4. As far as we know, this model is the first of its kind.

Finally, to assist with the integration of the conceptual model in the requirements elicitation process, we proposed a first version of the Gender-Inclusive Requirements Elicitation (GIRE) Framework, presented in Chapter 5. The GIRE framework intends help software and requirements engineers understanding the taxonomy of the conceptual model and includes a set of general guidelines and question-based checklists to elicit information for formulating gender-inclusive requirements. This work represents a first step towards addressing the fourth challenge.

1.4 Contributions

This dissertation contributed to the RE field with the following contributions:

- A paper accepted for publication and presentation at the ER 2021 conference, 40th International Conference on Conceptual Modeling (CORE A).
- A systematic mapping study on gender issues in software engineering with a comprehensive analysis of the state-of-the-art on this topic, including what gender issues exist in the context of software development and use, what approaches have been proposed to solve them, and a discussion of existent gaps in this research area.

- A conceptual model for gender gender-inclusive requirements based on the systematic mapping study. This is the first proposed conceptual model on this topic, as far as we know.

- A framework for integrating the conceptual model in requirements elicitation practices, namely semi-structured interviews, questionnaires, or creativity techniques. In addition, we developed a prototype tool to support the process of the framework.

- An evaluation of the conceptual model that discusses the overall perspectives of novice and expert participants on the model, including its utility and ease of understanding.

1.5 Document structure

This document is organized into seven chapters, and it is structured as follows:

- Chapter 1, *Introduction*, introduces the context and motivations for this thesis, identifies the problem statement and the associated challenges, and presents the objectives and expected contributions.

- Chapter 2, *Background*, provides a brief description of the Requirements Engineering field and the topics on which this dissertation is based. It provides an introduction to systematic mapping studies and to gender issues.

- Chapter 3, *State of the art*, presents the conducted systematic mapping study and a discussion of its results. Further, provides a comprehensive overview of the state of the art on gender issues in software engineering.

- Chapter 4, *Conceptual modeling of gender-inclusive requirements*, proposes a conceptual model for gender-inclusive requirements based on the systematic mapping study. It describes the development and conceptual foundations of the model.

- Chapter 5, *A framework for eliciting gender-inclusive requirements*, presents a framework for integrating the conceptual model in requirements elicitation. It also describes the development of a gender-inclusive prototype tool through the process of the framework.

- Chapter 6, *Evaluation*, reports the evaluation of the proposed conceptual model, including the planning and discussion of results.

- Finally, Chapter 7, *Conclusions and future work*, concludes the dissertation and presents opportunities for future work.

2

Background

The present research work aims at proposing an approach for addressing gender issues during the requirements phase of software development. The purpose of this chapter is to briefly introduce the main concepts in the field of requirements engineering that serve as the basis for this thesis, namely requirements tools, techniques and approaches for requirements representation. We also describe the process of performing systematic mapping studies, a method we conducted in Chapter 3. Finally, we present the concepts related to gender and gender issues that form the background of this work.

2.1 Requirements Engineering

Requirements engineering concerns the elicitation, analysis, specification, and validation of the needs and constraints for a given software system, as well as the handling of the resulted requirements throughout the software development life cycle. In this section, we provide an overview of the requirements engineering activities and approaches, with focus on requirements elicitation.

2.1.1 Requirements Engineering activities

Requirements engineering is the process that "covers all of the activities involved in discovering, documenting, and maintaining a set of requirements for a computer-based system" [88]. It plays a fundamental role in software and systems engineering by understanding, defining, and validating what is required from a system, identifying the system's constraints, and highlighting the motivation behind its development [192]. The activities involved in requirements engineering assure that the features and functionalities requested by stakeholders are analyzed and documented correctly, so the development team can accurately design and implement the system that meets the stakeholders' and users' expectations. [121]. The five main activities in requirements engineering are [153]:

- **Requirements elicitation** is the first activity in requirements engineering, and it is considered the process of discovering and researching system requirements and constraints by understanding the stakeholders' goals and the costumers' and users' needs. The development team is responsible for identifying and setting the scope of the system based on communication with costumers and stakeholders. The main goal of this activity is to gather the requirements for the system to be developed.

- **Requirements analysis and negotiation** is the process of reviewing all the elicited requirements to analyze and identify potential inconsistencies in their representation, such as ambiguous or conflicting statements, and adjust them to achieve an agreed view of the system's goals. This is accomplished by discussing and negotiating the conflicts with the stakeholders individually. The aim of this activity is to resolve all conflicts in requirements and meet the stakeholders' needs.

- **Requirements documentation** involves documenting the requirements established in the previous phase in an organized and consistent way. The resulting document gathers three types of requirements: *user requirements*, representing the user's needs and expectations, *system requirements*, formally describing what the system should do to meet the user's needs, and *domain requirements*, reflecting the system environment. The document produced during this phase will serve as the basis for the next phases and should be written in natural language with diagrams as support for comprehension by the involved parties in the requirements engineering process.

- **Requirements validation** is the process of discovering and correcting the conflicts, omissions, and inconsistencies in the requirements document, so the final document is of quality and represents the stakeholders' needs.

- **Requirements management** is the last phase of the requirements engineering process. As the system evolves, changes to requirements emerge that may impact or compromise established requirements. Therefore, changes must be tracked and analyzed to ensure the requirements document is updated, consistent and supports the system evolution.

This five activities constitute the general set of main activities in the requirements engineering process, although they can be adapted to meet certain organization' needs or changed depending on the system that is being developed [88]. In this dissertation, we focus on requirements elicitation.

2.1.2 Requirements Engineering approaches

Requirements engineering approaches are a set of methods and techniques used for documenting and analyzing the system requirements [88] during the activities mentioned above. Using natural language to write and express software requirements is an intuitive and flexible approach that has been used since the beginning of requirements engineering [88]. However, it is also difficult to capture relationships between requirements and it is prone to misunderstandings: the same concept can be written in many different ways, leading to ambiguities that rely on the interpretation of the reader [161]. Therefore, alternative representations such as visual models have been proposed to complement the textual representation of requirements during elicitation and analysis [36]. Visual models, such as diagrams or tables, provide a consistent and standardized structure, and can be supported by textual representations for effective communication with a variety of stakeholders. They act as a common understanding of the software system being implemented and provide meaning to the elicited requirements in a specific context by establishing the level of abstraction needed in each phase. Their selection must be a well-considered decision as it will influence and possibly restrict what can be detailed and represented [121]. Various approaches and different techniques have been proposed to support distinct modeling objectives in requirements specification and analysis such as the following:

- **Aspect-oriented**, is based on the identification, modularisation and composition of crosscutting concerns of a system, such as availability or usability. The separation of crosscutting concerns into different modules, known as aspects, enables reuse and modification independently of other concerns and highlights the modularity of the system [133].

- **Agent-oriented**, uses agents, independent entities with goals, to model social dependencies and interactions between multiple agents in the system. It enables the modelling of non-functional requirements [181].

- **Object-oriented**, uses objects to define the system requirements. Objects have a state and an information set regarding their attributes, functionalities, and interfaces to interact with other objects [88].

- **Scenario-based**, uses scenarios, a step-by-step description of existing behaviors, for representing requirements. Scenarios are based on real-life examples of stakeholders' interactions with the system, and therefore easy to understand [161].

- **Viewpoint-oriented**, uses viewpoints of different stakeholders to describe the system. Viewpoints represent the requirements and constraints from a diverse set of sources, which enables the detection of conflicts in requirements [36].

- **Goal-oriented**, uses the goals for "for eliciting, elaborating, structuring, specifying, analyzing, negotiating, documenting, and modifying requirements" [177]. Goals are formulated statements, at varying levels of abstractions, that represent the objectives a system must achieve. They are intuitive to stakeholders and can express both functional and non-functional requirements. Examples of notations are KAOS and i*.

2.2 Systematic mapping studies

A systematic mapping study is an unbiased method for identifying the quantity and type of existing evidence on a specific research topic [85]. The main goal is to provide an overview of the results available and to identify gaps in current research that need to be addressed [128]. In addition, the findings of the systematic mapping study can provide complementary and quantitative information for conducting systematic literature reviews. A systematic literature review is a method for "identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest" [85]. If, before conducting a systematic literature review, very little evidence on the topic is found or the topic is very broad, then a systematic mapping study is more appropriate and should be performed first.

The systematic mapping process has three essential steps: planning, conducting, and reporting. The planning phase aims at defining the research question and the research protocol of the study. The second involves conducting the search of the relevant primary studies, screening the resulting papers, and extracting and synthesizing the data from the selected studies according to the protocol. Finally, the reporting phase discusses the results and produces the systematic map [128]. The outcome of each step is the necessary input for performing the next one.

2.2.1 Planning

The planning phase concerns the specification of the research questions and the establishment of the research methodology to be followed during the conducting phase. It involves the following activities: the formulation of the research question, the definition of the search and study selection strategies, the establishment of the quality assessment criteria, and lastly, the specification of the data collection and extraction strategy.

Definition of research questions

The first step in a systematic mapping study is the definition of the research question that will be driving the research. The goals of the systematic mapping studies, such as mapping the frequencies of publication over time or identifying the publishing forums of certain research area, must be reflected in the research question, which can be broader and divided into multiple sub-research questions [85]. The definition of the research question is the most crucial step of the systematic mapping study. The research questions drive the search process that identifies the relevant studies from where the data will be extracted and analyzed in such a way that leads to relevant findings. By answering the research questions, we can get an overview of the current practices regarding the specific topic and identify current gaps in knowledge [87].

They can be structured according to the ?? method, that is, in terms of population, intervention, comparison, and outcome. The PICOC method provides a simple but focused structure to frame the research questions through these five element criteria [129]. A description of each of the criteria is presented in Table 2.1.

Population	Group of interest for the investigation
Intervention	Technology, tool or procedure that is used to address a particular issue
Comparison	Technology, tool or procedure to which the intervention is compared
Outcome	Significant findings of the investigation
Context	Context of the investigation, in particular it is an extended view of the population and it is where the comparison takes place

Table 2.1: PICOC structure

Once the research questions are defined, the search string that will run on the selected digital libraries can be formulated.

Search strategy

The search strings consist of a set of connected keywords based on the main terms of the research questions and driven by their objectives. The aim of the search is to gather a broad coverage of studies and thus the search strings are unrestricted and more likely to return a high number of results from different databases. To conduct the search, the research sources must also be selected. This involves the selection of the digital libraries covering the most important conferences that publish studies related to the research area of the study. The search for primary studies can be conducted through two search methods: *automatic* by using search strings on scientific digital databases and *manual* by browsing manually through relevant studies from conference proceedings or journal publications.

Furthermore, the snowballing approach can be applied to find additional relevant primary studies that could have been missed [80]. The snowballing approach can of two types, forward or backwards. The first consists of finding citations to the studies found and the second consists of scrolling through their list of references. These two methods can complement each other to gather the most complete set of relevant studies [194]. The outcome of both automatic and manual searches is the set of all relevant papers to be screened in the next step.

Inclusion & exclusion criteria

The inclusion and exclusion criteria are used to select which studies are relevant to answer the research questions [129]. As a first step, the inclusion and exclusion criteria should be based on direct issues such as language, journal, setting, participants, research design, and/or date of publication. Then, duplicate papers are also excluded. The exclusion criteria can determine what questions are not being addressed, and thus, establish the boundaries of the study. Furthermore, the criteria are defined with focus on the research questions to ensure an appropriate and reliable classification. Therefore, articles that do not answer any of them are excluded from the final studies.

Similarly, papers that provide relevant evidence to answer the research questions are included. The reading depth of the studies should be adapted according to the study [128]. Studies resulting from the automatic search can be clearly unrelated to our research questions just by reading their title and abstract; therefore, they can be excluded without further reading. Usually, digital libraries return a large number of irrelevant papers that can be immediately excluded [87]. However, if an abstract lacks information or is possibly misleading, the conclusions or any part of the full paper should be read to clarify how it should be classified. This level of detail requires more effort during this step but increases the validity of the final result. For these studies, the reason for exclusion should be documented for clarity and reliability.

Quality assessment

In addition to the inclusion and exclusion criteria, the studies should also be assessed by a quality criteria. This assessment is critical for ensuring trustworthy results when selecting the primary studies from where the data will be extracted. There is no agreed definition of 'quality' or what a high quality study consists of [87]. Nonetheless, the extent to which a study minimizes potential bias and maximizes external and internal validity are related to its quality. Thus, the quality assessment of a study regards the examination of how likely it prevented potential systemic errors, the internal validity, and the extent to which the effects of the study are applicable outside of the study, the external validity [87].

Quality assessment can be performed through the use of quality checklists. These are checklists of factors that need to be evaluated in a study [87]. The checklist should be constructed according to the context of the study, the empirical methodology, number of citations as well as with regard to the defined research questions. Then, a measurement scale should be created for each factor rather than a yes/no criteria. By attributing a numerical scale to each factor, it is possible to measure and compare studies through their overall score.

Furthermore, quality values can also be established to exclude studies that score below a specific value, while distinguishing medium from high quality studies.

2.2.2 Conducting the search

The conducting phase follows the research protocol defined in the previous phase and it is consists of applying the search process, selecting the relevant studies according to the inclusion and exclusion criteria and the quality assessment, and performing the data extraction and synthesis.

Screening the resulting papers

The retrieved papers from the automatic and manual searches are assessed through the set of inclusion and exclusion criteria that were determined prior to the screening to mitigate bias. These constitute the candidates studies of the study which will be either excluded or included for the next iteration of criteria application. The output of the screening is the complete set of selected studies that will be assessed by their quality and from where the data will be extracted to provide answers to the research questions.

Data extraction

In this step, the studies are sorted according to a defined classification scheme and the information needed for addressing the research questions is collected into a data extraction form. After the execution of the search string, the candidate studies are collected and imported to a bibliographic tool to facilitate the organization.

To reduce potential bias, the form should be previously designed during the planning phase and include standard information regarding each paper, such as the name of the reviewer, the date of extraction and the title, authors, journal and publication details. The goal is to record accurately all the information necessary to perform the quality assessment and answer the research questions. Therefore, the form must include the quality criteria. It also must include all the information relevant to the research questions and the category of the classification scheme. Finally, it should include a section of additional notes.

If possible, the data extraction should be performed by two or more researchers individually to cross check if the data on the extraction form is correct. This comparison analysis can give rise to disagreements that can either be solved through a discussion and ultimately, consensus, or, if viable, through an additional independent researcher [87]. Moreover, if the extraction is performed by two or more researchers where each is reviewing a different set of studies, a method for verification of consistency must be applied. For example, all researchers review the same random sample of the studies and assess the consistency of the extraction [87].

Nevertheless, the data extraction process should be specific enough to classify and categorize the studies that answer the research questions but broad enough that is not a time consuming task [87].

2.2.3 Reporting

In the reporting phase, the main findings are discussed for the research question or each research question, which are answered based on the analysis of the extracted data from the selected studies. The dissemination of the results can consist of a summary of the demographic data of the primary studies and a discussion of the results. The frequencies of publications for each category emphasize the gaps and possibilities for future research, while the extracted data focuses on presenting summarized answers to the research questions. The mapping of studies can be reported visually in a graphical representation, such as bubble plots [129]. These allow the visualization of the focus of past research as the size of the bubbles are proportional to the number of articles of each category. Bubble plots also allow the combining of categories. Finally, the discussion of the results does not involve in depth analysis of the principal findings but a broad one with enough detail to determine how well a certain research area is covered. This also provides a picture of what current gaps exist, and thus, allows the proposal of future research works.

2.3 Gender issues

The European Institute for Gender Equality (EIGE[1]) defines the concept of gender as the "social attributes and opportunities associated with being female and male and to the relationships between women and men and girls and boys, as well as to the relations between women and those between men. These attributes, opportunities and relationships are socially constructed and are learned through socialisation processes. They are context- and time-specific, and changeable" [52].

Gender is a structural construct of society, whose specific constructions vary from one society to another through its intersection with other social categories, such as historical and cultural background, class, religion, race, ethnicity, dis/abilities, and sexuality [52]. The social construction of gender defines what is expected, allowed and valued in the members of a particular society or community at that point in time according to social categories, devising gender roles that are imposed on individuals through educational, political and economic systems, legislation, and culture and traditions [162]. In turn, individuals perpetuate them by internalizing the social expectations for gender norms, and thus, behaving accordingly and expecting it from others [57].

This cycle of gender norms reinforcement affects the development of the individuals' gender identity as well as how they perceive others [57]. Gender identity is a person's individual and socially situated experience of gender that is influenced by the social and culturally constructed norms, opportunities, and expectations associated with the perceived person's biological sex [55]. The term biological sex refers to the set of biological and physiological characteristics that are not mutually exclusive, but are systematically used to differentiate humans as females or males [149]. The individual choice about how a person wants to communicate their gender is defined by their gender expression [54]. The demonstration of a person's gender identity is independent of biological sex, and it can vary freely at any time. It includes body appearance, dress, speech, behavior, personal traits, and actions [54].

[1] https://eige.europa.eu/

As it is being communicated, it is also being perceived and interpreted by others through social expectations and interactions based on the gender norms constructed and established within that particular sociocultural context [57]. The continuous internalization and embodiment of social expectations for gender norms produces the simplistic belief that only two mutually exclusive categories of gender exist, normalizing the gender binary as a natural category [59]. This belief limits the development of individuals at various levels (e.g., personality, education, profession, life opportunities) by ascribing them socially defined notions of femininity and masculinity according to their perceived sex [44, 106].

Furthermore, the institutionalization of such binary gender norms and roles constitute the basis for the gendered systems that create gendered power asymmetries and perpetuate inequality in society [61]. These bring forth gender stereotypes that are used to justify and maintain the "differences and inequalities between women and men in responsibilities assigned, activities undertaken, access to and control over resources, as well as decision-making opportunities" [52]. Preconceived beliefs and stereotyped ideas about individuals both result from and are the cause of discriminatory actions, prejudice, and sexism [60]. Specifically, sexism is directly related to positions of power in society, where those who have it are typically treated favourably and those without it are typically discriminated against [150].

Essentially, all societies' gender norms place women in the latter, and thus, at disadvantage in all domains of life, whether public or individual, regarding their freedom, participation, responsibilities, control of resources, and full enjoyment of rights [52]. The asymmetry of power, access and rights between women and men in institutions and structures prevail in societies, reinforcing and perpetuating gender-based discrimination and inequality [162]. Structural inequality is defined as the "embedding of gender inequalities in social structures, based on institutionalised conceptions of gender differences" [165], which create, legitimize, and sustain the gender discrimination and segregation that create gender issues [165].

Gender issues "include all aspects and concerns related to women's and men's lives and situation in society, to the way they interrelate, their differences in access to and use of resources, their activities, and how they react to changes, interventions and policies" [56]. The identification of gender issues in a specific societal context or domain involves the integration of a gender perspective. Adopting a gender perspective when analyzing any "social phenomenon, policy or process" [58] ensures the "gender-based differences in status and power" [58] are taken into consideration to ensure that all needs, concerns and experiences are represented [58].

By assessing the ways by which "discrimination shapes the immediate needs, as well as the long-term interests, of women and men", strategies that focus on overcoming these issues can be delineated and employed with the aim of achieving gender inclusion, and thus, overcoming inequality and providing equal benefits to all [58].

2.4 Conclusions

In this chapter, we presented briefly the concepts that served as the backbone for this dissertation. We start by presenting the activities of requirements engineering, followed by an introduction to the approaches used during these activities, focusing particularly in the requirements elicitation. Then, we describe the process of conducting a systematic mapping study, a method for identifying the quantity and type of existing evidence on a specific research topic. In chapter 3, we performed a systematic mapping study to gather an overview of the state of the art on gender issues in software development. Finally, we present the main concepts related to gender and gender issues in society.

3

State of the Art

This dissertation focuses on gender issues in software engineering. To identify and classify the research that is currently available on this topic, we performed a systematic mapping study. A systematic mapping study provides an outline on the current knowledge regarding a specific topic and simultaneously allows us to identify the main gaps that need to be addressed in that topic. It starts with the specification of the research questions and the establishment of the research methodology. Then, a search strategy is defined to detect relevant literature, that in turn will be submitted to an inclusion and exclusion criteria and a quality assessment to select the final primary studies. Lastly, data from the selected studies is extracted and analyzed to assess the existing evidence regarding the topics of the previously defined research questions. The outcome of this activity was an overview of the state of the art on gender issues in various technological domains, software development and requirements engineering. In this chapter, we detail each stage needed to conduct the systematic mapping study and discuss the resulting findings.

3.1 Research methodology

The overall approach of the research methodology was to contextualize gender issues in software engineering, specifically *what*, *where* and *why* they occur in software systems, and gain an overview of what methods and techniques have been proposed to address them. Therefore, we performed an systematic mapping study in three stages: planning, conducting and reporting. The activities in the planning stage are establishing the research questions, defining the search and study selection strategies, define a criteria for study selection, design a quality assessment of the studies and, lastly, present a strategy for collecting and extracting data. In the conducting stage, the preliminary results from the execution of the previous stage search are shown, namely, the results for the designated research query. Finally, they are discussed and analysed thoroughly in the reporting stage. The described methodology is illustrated in Fig. 3.1.

In the following subsections we detail each stage of the adopted research methodology.

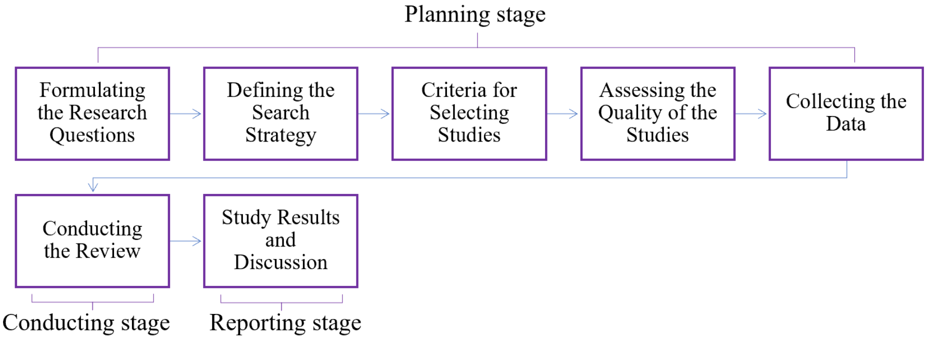


Figure 3.1: Research methodology for the systematic mapping study

3.1.1 Planning stage

The planning stage is the first stage of the systematic mapping study and includes five phases, described in 3.1. The following sections present each phase in detail.

Research question

The objective of this study is to provide a comprehensive overview of the existing approaches for addressing gender issues in software. Thus, we constructed the research question by using the **PICOC** method, which provides a five element criteria to frame a research question, as described in Table 3.1. The main research terms are high-lighted in bold.

Population	Studies that address **gender issues** in **software engineering**.
Intervention	The set of gender issues identified in software engineering and the approaches proposed for addressing them.
Comparison	The proposed approaches are not compared with each other. They are classified based on specific criteria according to our context.
Outcome	A thorough analysis of the state of the art of gender issues in software engineering: Gather the knowledge about how the concept of gender is defined in software engineering, what gender issues have been identified in software systems, and the proposed approaches for addressing them.
Context	Preparation for Master Thesis in Computer Science and Informatics Engineering.

Table 3.1: PICOC analysis with main research terms

This lead to the definition of the following research question:

How have gender issues been addressed in software engineering?

For a more focused and detailed search on the topics we want to address, we decomposed the main research question into four sub-questions, as shown in Table 3.2, along with the motivation behind them. This allows us to understand the quantity and type of research available to analyze and categorize its content regarding our intended context. Hence, we can obtain an overview of the current knowledge of gender issues in software and recognize the existing gaps in this area.

Research sub-question	**Motivation**
RQ1. How are gender and gender issues understood in software engineering?	To discover how gender, gender issues, and related concepts are understood and defined in the context of software engineering
RQ2. What are the software application domains where gender issues were addressed?	To discover the application domains where gender issues were addressed
RQ3. What approaches have been proposed to address gender issues in software development?	To discover what approaches, methodologies or guidelines currently exist for evaluating and addressing gender issues during software development
RQ4. Have the proposed approaches been validated?	To discover whether the proposed approaches for both software development and products have been validated through empirical studies

Table 3.2: Research sub-questions

Search strategy

The search strategy used for identifying the primary studies had two search methods that complemented each other: automatic and manual. The automatic search was conducted by using the set of keywords connected by the logic operators *AND* and *OR* presented in Table 3.3. The search string was created by following the two concepts in Table 3.3 that correspond to the key terms of the main research question and with the aim of representing the sub research questions' motivations (see Section 3.1.1).

Concept	Keywords	
Gender Issues	("gender issues" OR "gender diversity" OR "gender bias" OR "gender stereotype" OR "gender inclusive" OR "gender equality" OR "gender gap" OR "gender difference")	AND
Software	("software engineering" OR "requirements engineering")	

Table 3.3: Search string applied

The search string was run as a query on Google Scholar, a multidisciplinary digital library that provides scholarly literature from various sources such as academic publishers, professional societies, online repositories, universities and other web sites[1]. It includes articles from the most relevant conferences in Engineering & Computer Science, such as

ACM/IEEE International Conference on Software Engineering, and also conferences on more specific areas, such as Computer Human Interaction (CHI) or International Conference on Learning Representations. Given the purpose of this research work, we delimited the scope of our search by excluding studies that address educational software. Thus, we added *"NOT education"* to the end of the search string and narrowed the search results to more focused matches.

An important note regarding the applied search string: the *"AND"*, *"OR"* and *"NOT"* operators were adapted to the Google Scholar syntax by substituting them with a white space character, a vertical bar "|", and a minus signal "-" respectively.

After concluding the automatic search, we performed the manual search by using two different strategies to gather the most complete and consistent set of results. First, we did a manual search on Google Scholar and ACM Digital Library with the keywords from the research questions, scrolling through the resulting articles and selecting the most relevant ones according to the search purpose that did not appear previously. The second strategy we followed is called Snowballing [188], where we examined thoroughly the reference list of each selected study in order to find additional applicable studies that would complement the results of the previous approaches.

Study selection strategy

Each study that was found through the automatic and manual searches was evaluated according to an inclusion and exclusion criteria to exclude the ones that were not relevant to answer the main and sub research questions. The evaluation was performed in two iterations. The first iteration consisted of analyzing and reading the title and the abstract of each study, and simultaneously selecting the studies that met all of the inclusion criteria and excluding the studies that met at least one of the exclusion criteria. The inclusion and exclusion criteria applied in the first iteration is shown in Table 3.4.

ID	Study inclusion criteria
IC1.1	The article is available in English
IC1.2	The article is from a conference, workshop or journal
IC1.3	The article is available in full text
IC1.4	The title and/or the abstract of the article is related to the main and/or sub research questions
	Study exclusion criteria
EC1.1	The article is a duplicate
EC1.2	The article is not in English or is unavailable
EC1.3	The article is informal (slides, extended abstracts, blogs)
EC1.4	The article is an introduction for special issues, books, workshops or posters
EC1.5	The title and/or abstract of the article is not directly related to any of the research questions

Table 3.4: Inclusion and exclusion criteria for the first iteration

In the second iteration, the pre-selected candidates from the first iteration were read in full, and only those that satisfied all of the inclusion criteria were selected as final primary studies. Similar to previous iteration, studies that met one or more exclusion criteria were not included. The criteria for the second iteration is shown in Table 3.5.

ID	Study inclusion criteria
IC2.1	The content of the article discusses the main research question and provides answers to RQ1, RQ2, RQ3, or RQ4
	Study exclusion criteria
EC2.1	The content of the article is not focused on the main research question and does not answer any of the sub-research questions

Table 3.5: Inclusion and exclusion criteria for the second iteration

Quality assessment

After the selection of the primary studies through a set of inclusion and exclusion criteria, it is crucial to assess their quality to secure reliable results. We focused on two measures to evaluate each study: the number of citations according to year of publication and its CORE Rank, when available.

As a quality measure, the number of citations presented in the digital library can provide a subjective evaluation of the study's relevance. If the study is cited by a high number of papers, then it is presumably more trustworthy than studies with few or no citations. It is important to note that the year of publication was also taken into account when applying these criteria. The CORE Conference Ranking assigns categories to major conferences in the computing disciplines based on multiple indicators [34]. Conferences are assigned to one of the following categories: Rank C meets the minimum standards, B is good, A is excellent, and A* is a flagship conference.

Data collection and extraction

The purpose of this research is to collect evidence-based gender issues in software engineering. To simultaneously collect all the information needed to answer the research questions and manage the data that was extracted, we designed a data extraction template to record the content of each selected study. Thus, we can ensure that all selected papers are subjected to the same extraction criteria.

The template was designed as a form and it was based on the sub-research questions that had been defined. The form includes the primary information of the paper, such as the title, citations and year, and a section with the relevant information extracted regarding the research questions. The form is presented in Appendix A.

3.1.2 Conducting stage

The search to identify and collect the studies that provide answers to our research questions was conducted by following the steps of the Planning Stage (3.1.1). This procedure yielded a total number of 98 selected research papers for data extraction, and was completed on 11th July, 2020. As mentioned previously, the first step was to perform the automatic search by running the search string on the digital library Google Scholar. This procedure retrieved 729 candidate studies that were collected and saved in the bibliography tool, Google Scholar. Then, we read and analyzed the titles and abstracts of each potential paper to apply the first iteration criteria, as shown in Table 3.4. The papers that were selected according to the first iteration criteria were fully read. For each pre-selected paper, we applied the second iteration criteria, listed in Table 3.5. We selected 46 papers. Finally, we validated and extracted the data that answered the research questions from all the selected papers, and filled in their corresponding extraction forms (see Appendix A). As mentioned in Section 3.1.1, we also did a manual search to complement the automatic search, which added 76 more papers to the total number of studies (of these, 36 were found through keyword manual search and 40 from Snowballing). The summary of the results obtained by following the process described above is shown in Table 3.6.

	1st Iteration Studies
Exclusion Criteria	729
EC1.1	-16
EC1.2	-58
EC1.3	-7
EC1.4	-14
EC1.5	-574
Total	59
	2nd Iteration Studies
EC2.1	-30
Total	29
	Selected Studies
Manual Search	+46
Snowballing	+23
Final Total	**98**

Table 3.6: Results of the conducting stage

3.1.3 Discussion of the results

This section presents and discusses the results obtained in the planning and conducting stages (see 3.1.1 and 3.1.2). In the following subsections, we answer each sub research question as well as the main research question based on the analysis of the extracted data from the selected studies. We also included studies from the first half of 2020.

RQ1. How are gender and gender issues understood in software engineering?

The results of the systematic mapping study show that research on gender issues in software engineering has been growing over the years, not only in the number of studies published, but also in its understanding of the concept of gender, as shown in Fig.3.2. The number of studies published per year has fluctuated. However, there is a noticeable increase in interest in this research topic since the year 2016. Since then, the number of studies published annually has grown, and it is between 2016 and 2020 that most publications were made. Although we only included articles from the first half of 2020, this year already exceeds the number of studies from years before 2017.

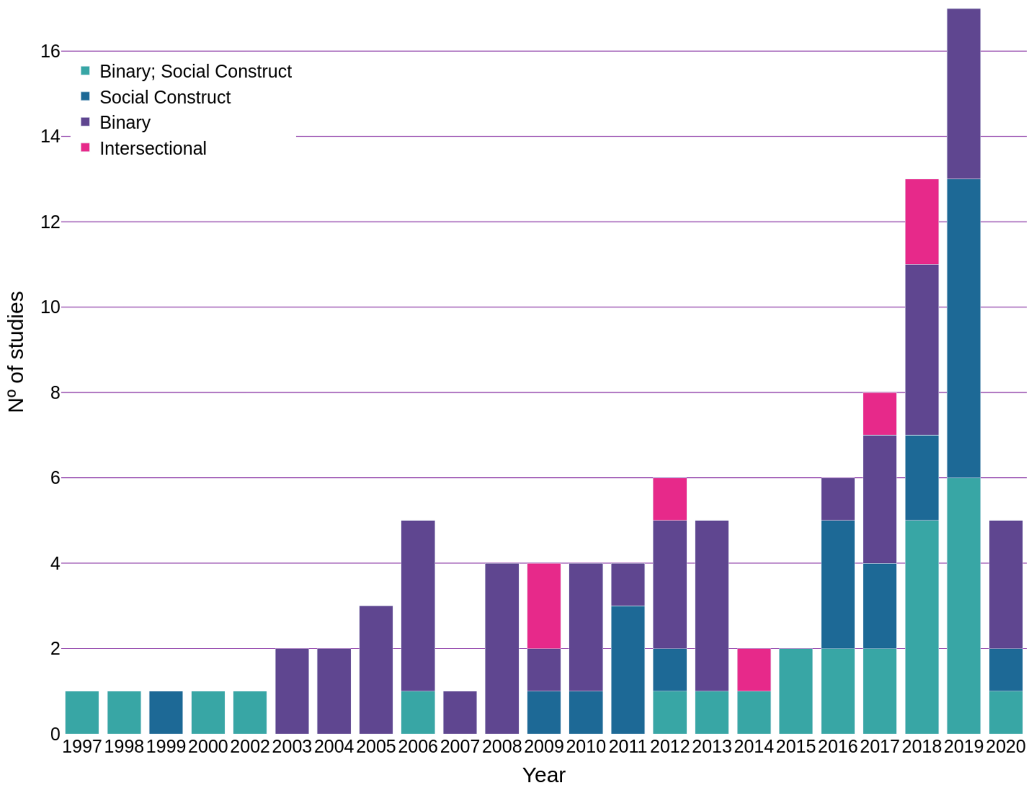


Figure 3.2: Number of studies on gender in software engineering per year and category

To analyze how gender is understood in software engineering, we first extracted information, implicit or explicit, concerning the gender definition from the selected studies. Then, we devised the following four categories of gender conceptualizations and classified the studies according to them: *Binary (categorical view of gender)*, *Binary and Social Context*, *Social Construct*, and *Intersectional*. These are represented in Fig.3.2 with the caption *GenderCode*. The first category includes the papers that do not have an explicit definition of how they represent gender throughout the study and use it as a categorical variable indistinguishable from biological sex (i.e., assume only two genders, female and male, that are aligned with perceived sex, woman and man respectively). In the second category, gender is represented as an individual identity (e.g., attributes and behaviors) and as a social role (e.g., opportunities and responsibilities), but the studies only provide and analyze data on the female and male gender. We included this category because although the papers addressed gender as binary, they recognized that this denotes a simplified understanding of the concept and that future approaches should integrate a more updated notion of gender. Therefore they are distinct from articles that do not recognize the limitations of such perspective. The third category comprises the studies that address gender explicitly as a social construct, including the concepts of gender identity and gender roles, to overcome inequalities and create inclusive software for everyone. Finally, the last category includes the papers that take an intersectional approach, that is, address disparities in software engineering based on the intersections of gender, race, and class. Both the third and fourth category represent gender as a cross-cutting variable. The classification involved a systematic reading of the papers and labelling of their content accordingly. The results are summarized in Table 3.7. In this answer, we also briefly discuss the conceptualization of gender issues for each category.

The majority of the studies, 43 out of 98, assumed a binary view of gender, focusing on individuals who either identify as female or male. Moreover, these studies addressed the concept of gender interchangeably with sex, assuming one's biological sex corresponds to one's gender. Thus, most studies did not discuss their model of gender and used the concept as a statistical variable with only two categories, female and male, to analyze differences between women and men in the use, preferences, adoption, or interaction with software, based on empirical research.

Further, the studies examined software systems and concluded that it privileges the attributes associated with male users. In this context, gender issues refer to the negligence of female users when developing software systems. We also found studies that investigated the absence of female users in online communities and proposed a set of recommendations for addressing the barriers that prevent women from participating [47, 90, 40, 79, 136, 93, 98, 113, 114]. In this case, gender issues refer to the male-oriented site design and over-representation of male users in the online communities that prevent female users from joining. Finally, the remaining studies identified and offered solutions for the gender bias against female users detected in algorithms' design and datasets [190, 17, 69, 111, 160, 46, 156, 3, 29, 135, 158, 30, 130].

Similar to the first, the papers from the second category presented differences between women and men based on binary female/male empirical studies. However, these papers did not consider gender as an inherent category and approached it from a social perspective, providing a cultural and contextual analysis of the gender-based differences in attitudes towards technology [107, 134, 43, 118, 179, 123, 76, 92, 170, 48, 74, 84, 185, 127, 159, 141, 131, 37, 182, 51, 94, 112, 16, 171]. Some of the papers also included gender-related concepts such as gender norms, gender roles, and gender identity. The studies understood gender as a self-described attribute influenced by social roles and acknowledge that the binary construct of gender is limited. Nevertheless, they still tackled gender as binary. Furthermore, the studies highlighted the need for inclusive technology to prevent gendered software systems. In this category, gender issues resulted from not considering female users' preferences, needs, and perspectives during development and/or introducing gender-based stereotypes against female users in software systems.

The studies classified into the third category explicitly conceptualize gender as a social construct and discuss the limitations of assuming both the binary sex and binary gender perspectives. As a social construct, gender is a spectrum in which individuals align themselves by expressing their gender identity, independent of biological sex. Thus, differences in attitudes towards software are related to differences in gender expressions that result from internalized gendered behavior from a given social context. Gender issues arise when gender stereotypes and traditional views of gender are integrated into software systems [144, 155, 86, 6, 5, 193] or these systems are developed with socio-technical properties preferred by users with male attributes [19, 73, 110, 166, 138, 66, 103, 24, 26, 22, 102, 77, 105].

The category with the fewest papers, only three, corresponded to the *Intersectional* conceptualization, which is proposed for the first time in 2014. The papers from the fourth category present the concept of intersectionality in the context of software engineering [145, 18, 21]. An intersectional approach seeks to explain how gender intersects with other social identities, such as race, culture, and class, and how these intersections form multiple, layered identities. From this perspective, gender can be expressed and understood in diverse ways, and thus social and cultural context are fundamental factors to consider when developing inclusive software. Likewise, gender issues refer to algorithms and software that embed societal gender biases and the consequences of a limited and simplified user identity construction.

Gender Concept	**Description**	**Nº of Papers**
Binary	Gender is used interchangeably with sex and assumes only female or male individuals	43
Binary, Social context	Understands gender as a social construct but only addresses female and male genders	26
Social construct	Understands gender as a social construct and uses concepts such as gender identity and roles	22
Intersectional	Gender and other social categories influence experiences and positions in society	7

Table 3.7: Conceptualization of gender in a total of 98 studies

In summary, most studies have addressed the concept of gender as a binary category inherent to the individual. However, this approach coincides with older studies, and although some recent studies address it from this perspective, the conceptualization of gender in software engineering as a social construction or through intersectional frameworks has been increasing since 2014 (see Fig.3.2). Concerning the gender issues identified in the studies, they are mostly transversal to the gender conceptualization categories. Briefly, gender issues in software engineering include stereotypical masculine culture in the field, not accounting for women's needs, preferences, and perspectives when developing software, and simplifying the representation of complex gender identities in algorithms. In this sub-research question, we answered how gender and gender issues have been addressed and what concepts have been used in software engineering from a broad perspective. In the following sub-research questions, we continue to discuss gender issues in more detail.

RQ2. What are the software application domains where gender issues were addressed?

The results for RQ2 revealed a broad set of software application domains where gender issues were addressed. We found several different and specific application domains, and therefore we aggregated the selected studies according to one of the following criteria: the article was classified as the more general domain from which they are part of, or, in case there was a significant number of articles from a more specific domain, we aggregated them together as a separate domain from the general one. The domains and respective number of studies found are summarized in Table 3.8.

From the selected studies, *Human Computer Interaction* was the domain with more papers with 26 out of 98 selected studies. Of the 28 articles, most are about or related to the GenderMag method [26, 110, 22, 102, 179, 24, 74, 73], a process for detecting gender bias within software, based on a set of personas structured around five underlying gender differences in problem solving skills, and a set of materials for fixing the gender barriers found in the software. Prior to the construction of the GenderMag Method, there was supporting evidence of gender differences in the interaction and performance in computing tasks [71, 11, 12], highlighting the need to develop gender inclusive technology. Following these research findings, studies emerged with concrete focus on the inclusion of gender perspective in technology design [5, 6, 103, 65, 166, 185, 16, 145, 138, 86, 18, 27]. The remaining 5 studies focus on understanding gender differences when designing and developing interfaces [157, 76, 89, 96, 171].

The second application domain with more articles was *Machine Learning* with 17 of the 98 studies. Overall, the studies share the common goal of preventing algorithms from perpetuating gender stereotypes and are motivated by ethical concerns. From the 17 papers, 13 were published between 2015 and 2020, showing that research on this domain is receiving significant attention in recent years. The most covered topics have 5 and 6 out of 17 studies respectively: Natural Language Processing, where different approaches are proposed to mitigate gender bias in natural language data sets and text analysis tasks, [190, 46, 92, 69, 37] and Gender Recognition, Classification and Prediction [3, 144, 160, 130, 21, 155]. The topics with less studies are Information Retrieval and Search Engines with 3 out of 17 [84, 123, 134] and Chatbots [193, 43, 108].

The third application domain with more articles was *End User Applications*. As previously mentioned, gender issues were found in some types of software features and environments, particularly in problem-solving software, with 16 of the 98 selected studies. The studies are intended to empirically investigate gender differences in attitude towards risk, self-efficacy and learning style, and how it impacts the effectiveness of end users in debugging [8, 10, 14, 31, 9, 25, 68, 167], programming [13, 142, 23, 152, 127, 83] and security behaviours and perceptions in a home computing environment [109]. One study proposes an end-user profile formation model based on empirical gender differences in interaction with end-user applications [173].

Twelve out of the 98 articles focus on understanding why gender bias and lack of diversity persist in *Open Source Software*, mainly in participation and contribution to project development [47, 107, 90, 40, 79, 136, 93, 98, 113, 182, 114, 19]), followed by *Web Applications* with 8 out of 98 articles that study gender differences in interaction with web applications [51, 141, 49, 139] and in web interface preferences and designs [115, 35, 112, 159].

Furthermore, the application domains *Social Networking Services* included 5 articles, and *Game Development* 6 out of 98 articles. The studies from the *Social Networking Services* domain focus more on gender differences in the platforms' usage ([116, 97, 50, 169, 124]), whereas *Game Development* has 6 studies that address gender bias in game design ([48, 164, 70, 184, 105, 187]) and 2 that propose gender inclusive approaches for game development and design ([77, 168]). Five out of 98 articles investigate gender differences in perception, acceptance and use of mobile devices ([99]), which includes communication technologies [78, 64], websites [126] and privacy [143], constituting the *Mobile Applications* domain.

The application domains in which gender issues have been least addressed are *Smart Cities, Mobility & IoT* ([118, 94]) and *Requirements Engineering* ([183, 66]) with 2 out of 98. Finally, *Virtual Reality* ([95]) with 1 out of 98 articles. The papers included in the *Mobile Applications*, *Social Networking Services* and *Virtual Reality* domains are focused on finding and documenting gender differences in the user's perception, preferences and behavior, while the papers from the *Smart Cities, Mobility & IoT* and *Requirements Engineering* domains are more focused on detecting and overcoming gender bias.

RQ3. What approaches have been proposed to address gender issues in software development?

As previously mentioned, the set of software application domains where gender issues were addressed was broad, highlighting the need to establish measures to foster inclusion and avoid perpetuating bias in software technologies. Most studies present empirical research on gender differences in perception, preferences, adoption and use of software.

Domain	Nº of studies
Human Computer Interaction	26
Machine Learning	17
End User Applications	14
Open Source Software	12
Web Applications	8
Social Networking Services	5
Game Development	6
Mobile Applications	5
Smart Cities, Mobility & IoT	2
Requirements Engineering	2
Virtual Reality	1
Total	**98**

Table 3.8: Application domains where gender issues were addressed

However, very few existing works propose approaches to address them during the system's development. One of these works is GenderMag [26], a method for evaluating software features and finding gender-inclusiveness issues in software design. The method uses 5 facets of gender differences relating to problem-solving, which are motivation, information processing style, computer self-efficacy, risk aversion and tinkering, and encapsulates them into a set of personas for performing a systematic process of evaluating software so the design is more inclusive for everyone. Another work, IT&me [103], presented gender-sensitive personas as an artefact to be used in a participatory design process of software to ensure the diversity of female perspective. The personas were developed with an agile, iterative approach model that involved potential users of the software platform, preventing the integration of stereotypical gender assumptions in the platform design.

We also found two studies from the *End User Applications*' domain that propose approaches for addressing gender issues in software development. The first is an end-user profile formation approach for behavioral modeling implementations called 'RULES', proposed by [173]. This model consists of five behavioral attributes influenced by the user's gender, namely risk perception, usefulness perception, learning willingness, ease of use perception, and self-efficacy, that were collected from HCI and EUD behavioural studies. The second study we found, described the development of an application, SATIN 2 [127], that would allow any end-user without programming skills to create a smartphone application and thus providing women the tools to participate in the development of software products. To accomplish this, the study presents a set of methods to design and implement features from a gender perspective with the purpose of balancing gender differences in perceived self-efficacy and support female strategies in end-user programming.

From the list of studies in the *Game Development* domain, we found a research work that proposed an integrative approach to understand and evaluate gender inclusiveness in game development through a framework, *Gender Inclusivity Framework (GIF)*, that guides the design of gender inclusion in games [77]. Moreover, we found a study from the *Smart Cities, Mobility & IoT* domain where a conceptual model that relates gender differences in daily mobility patterns and social roles with user acceptance of smart mobility technologies is proposed [94]. The model offers an theoretical and analytical structure to understand gender specific travel behavior by emphasizing the differences in everyday-life situations between women and men that influence the acceptance and use of 'smart' mobility options (e.g., autonomous driving and sharing) to consider the diverse requirements and needs of different social groups and the goal of providing mobility for all.

Lastly, we found an article that addressed gender issues in requirements engineering [183]. During the requirements elicitation process, the users inform the analyst about their expectations for the system to be developed. However, gender differences in communication patterns can result in a misinterpretation of information by the analyst, leading to inaccuracies and errors in system design. The study uses the *Modified Coherence Method (MCM)* to overcome these differences by adding a structure to the requirements gathering process and allowing a clear information transfer in interviews between mixed gender users and analysts.

Gender Issues	**Proposed Approach**
Gender-inclusiveness issues in problem solving software and lack of approaches for software practitioners, such as User Experience (UX) professionals or software developers, for addressing them	*GenderMag* - a systematic evaluation method for practitioners to find the gender-inclusiveness issues through five facets of gender differences that impact use and usability of software
Lack of diversity in female perspectives in the design process of software lead to personas that promote gender roles and stereotypical perceptions	*IT&me* - set of four gender-sensitive personas, developed with an agile, iterative approach model that involves potential users
Gender differences in performance and in correlations between performance and a set of behavioral attributes	*RULES* - a user profile formation approach consisting of five attributes influenced by gender
The gender imbalance in the ICT sector, lacking women and being gender-segregated results in gender differences in end-users' programming skills	*SATIN 2* - a system to provide end user development environments, where people w/ no programming skills can develop a mobile app
Lack of a cohesive understanding of gender inclusiveness in games and the relationships that exist between the different dimensions and components to support game designers, researchers, and educators in the design of gender inclusive games	*GIF* - a framework that enables the deconstruction of the concept of gender inclusiveness in games into smaller, conceptually distinct and manageable components to design gender inclusiveness in games
Gender differences in mobility patterns and acceptance of new mobility concepts and automated driving that are currently not being taken into account in development	*Conceptual model* - integrates a theoretical framework on gender specific mobility patterns and technology acceptance elements
Gender differences in discourse may create a communication barrier between the interviewer and the interviewee during the requirements elicitation process	*MCM* - a method for adding a structure to the requirements gathering process in interviews that allows clear information transfer between mixed gender users and analysts

Table 3.9: Definitions of gender issues and their respective proposed solutions

RQ4. Have the proposed approaches been validated?

All the studies presented in RQ3 were validated with the exception of the project IT&me [103] and [183]. Therefore, we present the studies and their validation methods to answer RQ5 in the same order as in RQ3. The GenderMag method has been empirically validated through a research report with 10 user experience practitioners [26], a study with 20 participants to investigate whether the initial prototype was more gender inclusive after the GenderMag analysis [179] and two researches, one with 10 professional software teams, and other with Microsoft, that attempted to integrate the method in their software projects [73, 24]. Overall, the results showed that applying the GenderMag method improved the software's inclusiveness and eliminated the gender gap in the its design.

The Game Inclusivity Framework [77] was also evaluated by five experts including academic researchers and professional designers to validate its gender-inclusive components. The SATIN 2 project [127] was evaluated through an observation study where 11 participants used the software to build mobile applications. All of the participants managed to finish the applications and most were satisfied with their accomplishments. Furthermore, the conceptual model proposed in [94] was revised according to the findings of an empirical research that gathered quantitative studies of gender-specific mobility patterns and qualitative studies on the acceptance of new mobility concepts and automated driving.

Finally, the 'RULES' model was evaluated through a field test on a population of 30 end-users that built their own web database-driven application and then answered an online survey about their perceived behavioral attitudes when interacting with the developing tool. The results confirmed that the attributes of the model influence performance differently for every gender [173].

Main RQ. How have gender issues been addressed in software engineering?

Overall, gender issues in software engineering is a research topic that has received growing attention in recent years. Initially, the concept of gender was treated as a binary variable to analyze gender differences in preferences, needs, and use of software systems. The studies highlighted the importance of acknowledging these gender-based differences to develop successful software for female and male users. Otherwise, presumably neutral software systems would privilege male users.

In recent years, more research emerged addressing the concept of gender in a non-essentialist way, that is, acknowledging the social context of gender and introducing concepts such as gender identity and gender roles. These papers provided updated definitions of gender in the context of software engineering and focused on developing software that is inclusive for everyone. Additionally, we found five studies [145, 21, 18, 67, 82] that stated that intersectional practices applied to software engineering would be the most complete when developing inclusive software. Understanding the complexity of identity, with gender as one layer of that identity, provides a more realistic representation of users.

As discussed in RQ2, gender issues have been addressed in various domains of software, and as such, diverse approaches have been proposed and discussed according to the intended application context. In the *Machine Learning* domain, gender issues referred to gender bias in datasets and in the development of algorithms and models. The conceptualization of gender issues as significant gender differences in perception, preferences, adoption and use of software where the female attitudes and perspectives are neglected was presented across software domains.

From these studies, we were able to extract an empirically validated set of individual differences in the interaction with software that cluster by gender. Statistically, users with female attributes tend to see technology as a tool to accomplish a goal, exhibit a comprehensive processing style, have low self-efficacy, show aversion to risk and are process-oriented learners [26]. Users with male attributes see technology as a source of entertainment, are more selective, exhibit high self-efficacy, are risk-tolerant and learn by tinkering [26]. Software features are designed to be more supportive of problem-solving attributes typically associated with males rather than females, and thus, these differences in attitude impact software use and feature acceptance [26].

Additionally, these attributes are interrelated and mutually influenced. Self-efficacy is positively correlated to *Perceived Ease of Use* for female users [83] and is related to *Willingness to Learn* [23] and *Tinkering* [14]. *Motivation* for using the system is also related to the willingness to learn different features [83], which may be lower for female users [68], and to the *interaction environment*, where female users are more motivated by a collaborative environment, rather than a competitive one [8, 77]. The perceived cost-benefit of learning may also be higher than a male user's perceived cost to learn the same feature [173]. Moreover, perceived ease of use is more important for female users, while male users are more influenced by perceived usefulness [94].

Regarding *visual design*, a study on web interfaces identified *sense of belonging* as an issue for software adoption and use [112]. Women's perceived sense of belonging and consequent motivation for using the software system are impacted negatively when user interfaces are driven by gender-biased design choices in aesthetics, images and language [112]. Although female users were more affected by the stereotypical images, male users were also affected, and both benefited from the gender-inclusive design [112]. Visual design is also more important and impacts levels of credibility for female users [126]. In addition, they are more accurate in assimilating and decoding verbal and nonverbal cues [159, 112]. If the language or communication style of an interface or online community is not gender-inclusive (e.g., masculine gender-exclusive language, "boy's club", sexist language) their perceived sense of belonging, willingness, and motivation to engage decreases [112, 47].

Furthermore, users with female characteristics are more concerned about privacy [143] and are more socially oriented [99]. Therefore, they prefer technology that enables them to be available and connect to others, specially with people they already know [116]. They also have less access to digital resources [94], less previous technological experience

[47], are more sensitive on cost of purchasing [99], and have less time available to interact with the system and can be discouraged if it compromises too much time to learn [48, 187]. Thus, the compatibility of the system with the everyday routine of these users is essential for its use [94].

We also found studies that aimed at understanding why women engage less in online programming communities, such as StackOverflow [47, 107, 19, 182], and GitHub [79], to propose measures for improving female participation in open source development and reduce the gender participation gap. These studies identified gender issues in how StackOverflow are designed that affect women's participation, based on gender differences: women have more doubts in the level of expertise needed to contribute, feel that they have limited knowledge of site features, are intimidated by the community size, and have stranger discomfort [47].

Users also report an unwelcoming and hostile environment in StackOverflow that furthers criticism and sexist language [47, 19]. Additionally, StackOverflow rewards its users through reputation points that are attributed based on the activity on the site, which gives twice as much points for answering questions than asking them. Because women ask more questions whilst men answer more, women have lower reputation points [182]. These studies also propose a design solution for each of the detected issues to make the site more fair and inclusive.

Finally, various studies presented theoretical discussions that introduced gender and gender-related concepts in the context of software development, describing the issues with specific notions of gender in the field, and proposing more adequate approaches for integrating gender in the software development process [7, 5, 6, 82, 18, 132]. The studies argue that gender inequality in the software development field creates biased systems and discuss how gender-neutral development not only does not account for diverse users with different needs and preferences but could also introduce unintended harmful assumptions and embed gender stereotypes into the software. Therefore, they highlight the importance of diversity in the social context of software development to create software whose values are aligned with society's.

In summary, there is already a set of well-defined gender differences in how software is used and perceived in literature, as discussed here and in the answers to the sub-research questions. However, the number of studies that approached these gender differences to reduce or remove them was relatively low compared to the total number of articles found. Mostly, studies highlight the need to include diverse perspectives to foster inclusion in software but do not propose concrete solutions for solving them or tackle the task of avoiding gender issues during development. Furthermore, the range of approaches where gender issues were addressed is limited. As mentioned in RQ2, the studies that addressed gender issues proposed approaches for the software design phase of development. We only found one study, discussed in RQ3, where gender issues were approached prior to design, namely in requirements elicitation.

3.2 Threats to validity

In this study, we took actions throughout the planning stage of the systematic mapping study to mitigate these threats and consequently increase the correctness and completeness of the results. However, the research practices and methods used, like all studies, have limitations and implications that constitute threats to the validity of the resulting findings. For the identification of these threats, we followed the proposed guidelines in [189] for internal, external, construct, and conclusion validity.

3.2.1 Internal validity

Internal validity concerns the risk of not including all the information in this study that could have influenced the results. In systematic mapping studies there is always the risk of not including all relevant studies. When defining the search string used in the automatic search, we attempted to gather all the keywords' synonyms that represent the research questions, however, the results found can still be incomplete or inaccurate since there could be relevant studies that do not use the searched keywords. To minimize this threat, we tested several search strings based on the number of relevant results they retrieved, and chose the most optimal. There is also the possibility of excluding relevant studies. Since the study selection and data collection were performed by only one person, and the number of studies was high, a data extraction form and a list of inclusion and exclusion criteria were defined to avoid, at least to some extent, introducing selection bias in the process. However, since the number of studies was high, we could have missed some relevant work due to tiredness.

3.2.2 External validity

External validity concerns the extent to which the results can be generalized. To ensure we did not miss conferences and journals that published relevant research work regarding our purpose, we opted for a digital library, Google Scholar, that provides a very large set of publications in the Software Engineering field. We wanted assure that the automatic search, performed during the search strategy stage, returned the maximum results possible. Additionally, we performed a manual search on distinct digital libraries, to ensure that we found the greatest number of studies regarding our research purpose. Nevertheless, we did not consider Non-English or unavailable studies.

3.2.3 Construct validity

Construct validity refers to the degree to which the study measures what it claims to be measuring. The completeness and correctness of the mapping study depends on how well the it was conducted. One threat concerns the search string not including all the relevant keywords to capture the all the applicable studies to our research questions. This was mitigated by validating the search string within the authors, supervisors and an external reviewer. Additionally, to avoid narrowing the subject of the mapping study and consider all factors that are relevant for analyzing it, the research objective was divided in four sub-research questions and one main research question. Thus, we ensure the topic is well covered and the research questions are answered adequately.

3.2.4 Conclusion validity

Conclusion validity is concerned with the similarity of results if the review is performed by different researchers. We do not know how the search engines of the used digital libraries work. To ensure the results of our automatic search did not differ each time we ran the search string on the same or different digital libraries, we saved the retrieved articles in the bibliography management tool of Google Scholar, *My Library*. Therefore, the studies were organized for data collection and extraction. We also created a data extraction form for each selected study to structure the data necessary to answer each research question.

3.3 Synthesis of the results

Software technologies have unarguably become a significant part of society and people's everyday life, exerting a growing influence on our personal and social experiences, opportunities, and behaviors. The increasing society-wide use of software in several areas and for multiple purposes confers it a social character where its evolution is not only motivated by technical improvement but also according to society's changes in needs and expectations. Indeed, the software development process, especially the requirements engineering phase, is determined by a range of social factors and contextual influences, such as the needs and values of development teams, stakeholders, and organizations [32]. These values shape the decisions made during development regarding the intended software system. Therefore, the final software product or service embodies and reflects them, influencing, either positively or negatively, the community of individuals who use it.

However, the preferred needs and values of the individuals involved in software development might not reflect those of the software users because these individuals represent a very narrow subset of the population. While the user population can be highly diverse, the development teams, stakeholders, and managers responsible for the development and maintenance of open-source and closed-source software are overwhelmingly male and with high levels of education and income [151]. Since the software is the product of the values and interests of the individuals and organizations directly involved in their creation, the exclusion of a vast majority of potential users from participation in the decision-making process suggested the software may not match the needs, preferences and, values of its users, specifically female users. The under-representation of women and gender imbalance in positions of power in software development suggested a strong alignment between masculine values and the software's values.

Thus, our goal was to study the concrete influence of gender in the context of software development and identify what specific gender issues currently exist and where and why these issues occur. The first step towards fulfilling this goal was to conduct a systematic mapping study on gender issues in software engineering, which provided us with an understanding of how gender is conceptualized in the field of software engineering, what gender issues have been identified, how they affect the users, and what approaches have been proposed to address them.

The results of the study confirmed the hypothesis that gender plays a significant role in the attitudes toward software and that software systems are favoring the characteristics statistically associated with the male gender. A total 98 out of 729 studies were selected and the information presented in them was extracted and aggregated. These studies showed that, despite being a recent research topic, various domains have detected gender bias in algorithms, such as discrimination based on race and gender in automated facial analysis [21] or natural language models that reproduce gender stereotypes learned from biased datasets [92], and identified gender differences in attitudes towards software that are not accounted for when systems are developed, such as problem-solving software [26]. Gender differences in experiences, opportunities, roles, responsibilities, and levels of access and decision-making imply distinct needs and expectations from technology [90, 143, 75].

However, the assumption that software development is gender-neutral leads to these differences in needs and expectations being disregarded and software systems are favoring the needs and skills of male individuals, placing female individuals at a disadvantage when using technological products [107, 10, 23, 68, 92, 185]. For instance, employing the design technique I-methodology [18] that models users' behaviors based on those of development teams, which are usually predominantly men, without considering gender, results in software systems designed to expect users to learn and engage with new features through tinkering. Yet, female users are statistically less likely to do so compared to male users [26].

Moreover, the results of the systematic mapping study showed that software systems often reproduce societal gender stereotypes if gender is not addressed thoughtfully or in any way. When gender is explicitly addressed, confusion and misconceptions about gender and assumptions about women and girls as an homogeneous group can inadvertently integrate harmful stereotypes into the software. In particular, the 'shrink it and pink it' practice [138] that creates "soft pink", simplified technologies, such as fashion and wedding video games [77], that impose and reinforce traditional notions of femininity [18]. Additionally, [145, 132, 18, 5, 27, 82, 92] discussed how the lack of diversity within technology companies, organizations, and communities restricts the discussion of different viewpoints and constraints the inclusion of a diverse range of perspectives, which may integrate, whether conscious or unconsciously, untoward assumptions and stereotypes into the software. The studies propose the adoption of feminist and gender perspectives into software design to support the development of inclusive software.

Bellow we present a summary of the three main gender issues we found from the results of the systematic mapping study.

- **Software is gender-neutral** Predominantly male technology development teams assume software technologies as gender-neutral. Mostly, they perceive their own needs, preferences, and values as universal and applicable for everyone and anywhere. Additionally, there is a lack of understanding from those involved in software development about the concept of gender and its impact, predominating the idea that these two fields are unrelated, or that gender does not impact software.

- **Binary conceptualization of gender** If the concept of gender is addressed, it is typically simplified to a statistical binary category, sometimes associated with biological sex. This approach highlights the existence of gender differences. However, it does not fit them in a social context to understand why they exist, and therefore, does not provide any information about how to solve them. In fact, a strict female/male binary approach can embed gender stereotypes in the software.

- **Lack of approaches in RE** When awareness is raised regarding gender issues in the software development process, there are phew concrete approaches to identify and address them in the early phases, specifically in the requirements phase. In fact, we did not find any requirements engineering methods or tools to tackle gender issues from a non-binary perspective nor common and practical definitions for gender requirements. In the absence of a concrete method to address gender in requirements, software systems are failing to include diverse perspectives and are statistically privileging male users, disadvantaging and discriminating against users who identify with other characteristics.

Very few of the 98 studies proposed approaches for addressing gender issues during software development (see Section 3.1.3), which poses a gap in this research area. Mostly, the studies presented proposals for the design phase of the software. For instance, the IT&Me [103] and the SATIN 2 [127] projects both presented a case study in which a gender perspective is integrated in the design of a specific software application. Although they present a process that can be applied in other software projects, both depend on the developers' knowledge of gender issues and the constitution of multidisciplinary teams. Nonetheless, we found approaches that overcome this, such as the GenderMag method [26] and the RULES [173] attributes, that propose the use of gender differences in end-users to design inclusive features for problem-solving software. Similarly, GIF [77] presents a set of components to design gender inclusiveness in digital games. However, gender issues arise early in the software development process [185], and thus, they should be addressed in the early stages to avoid propagating them to following ones. Furthermore, we only found one paper that proposed a method for requirements, the MCM [183], that addressed gender differences in discourse to reduce misunderstandings during the requirements elicitation process. However, this method focuses on statistical differences in how requirements are expressed in natural language for the same requirements, and not on the elicitation of diverse requirements. Finally, a paper [94] proposed a conceptual model for gender differences in mobility patterns and technology acceptance criteria to be taken into consideration in the formulation of requirements for smart mobility technologies. This paper proposes the use of these differences, supported by empirical evidence, to ensure that the technology considers all types of users. Although it integrates a gender perspective in the development process, it does not conceptualize gender. It represents its impact on the smart mobility technology but presupposes knowledge of gender and related concepts by the development team. To build systems that achieve a quality such as gender-inclusion, it is necessary to understand the concept of gender as whole. That is, what is the meaning of gender in the context of software development, its impact on the software system and its influence on the social and individual attitudes of users. Therefore, a taxonomy for gender in the context of development is necessary to elicit, implement, and validate requirements that will lead to gender-inclusive software. A taxonomy can provide software development teams with the knowledge to make better informed decision regarding gender inclusion and avoid confusion and conflicts in the elicitation. Hence, a practical, evidence-based definition of gender is first necessary for software and requirements engineers to adequately address this concept and elicit gender-inclusive requirements. In this dissertation, our aim is to address this research gap by developing a gender-inclusive requirements engineering approach based on the results from the mapping study. Thus, we analyzed several studies from the existing body of knowledge that, although not proposed in the context of requirements engineering, are valuable for our goal and form the groundwork of our proposed approach.

3.4 Related work

The systematic mapping study we conducted aimed to provide a comprehensive overview of the existing approaches for addressing gender issues in software. We started our search by understanding how gender is conceptualized in the field, what gender differences exist and in which domains gender issues have been addressed, and focused on gathering the existing approaches for software development, with emphasis on requirements engineering. We also verified if the approaches had been validated. Since our focus was on software itself, we did not include works related to software teams or environments in the research questions driving our mapping study. Nevertheless, these works provide relevant insights regarding gender issues in participation, expertise and organizational structures in information technologies that complete our review of the state of the art in the topics this dissertation is based on. Moreover, this related work section presents further insights into how software teams and environments are addressing the gender gap in technology towards diversity of ideas and inclusion of female perspectives.

Gender issues arise early in the software development process. From the engineering team to senior managers and stakeholders, the individuals involved in the development of software products are predominantly male. Therefore, the ideas, perspectives, and experiences that shape the evolution of technology are limited. Software development is assumed to be a technical process with objective practices. However, it is driven mainly by the needs of males, which contributes, even if unintentionally, to gender bias in software products. Hence, gender awareness must be present from the beginning of the development process, namely in requirements elicitation, to reduce the chances of gathering a biased set of requirements that do not accommodate diverse perspectives. Although we did not find any paper that addressed gender issues during requirements elicitation nor analysis, we found studies with relevant findings that can serve as groundwork for this dissertation. One of these studies presented a participatory design process focused on gender, that was based on diverse female perspectives, to reduce the gender bias in research tools and designed artifacts [20]. The study discovered relevant gender differences in the use and inclusion of technology in daily life: women were more concerned about interpersonal communication and care for others, while men used technologies for entertainment; for women, being available for others was critical, while men reported not having issues with being unavailable; and finally, women had more challenges in managing their own and other people's needs and were interested in a more diverse set of topics than men. Still in technology design, two design projects that aimed to develop a virtual city for everybody were analyzed and compared in [125], where the female perspective was not included since it was assumed that the design technique employed, namely the I-methodology, was neutral. Consequently, the design of the projects matched the preferences, technical capabilities, and learning style of men. Both projects failed to attract the intended audience for neglecting gender as characteristic of the user. The study concludes that involving more women in the design process, as designers and as potential users, will lead to more inclusive technology. Additionally, it highlights the importance of designing technology adapted to everyone, specially to women, instead of women adapting to technology. The study proposes to not configure gender-neutral users since bias from the design team can subvert the intent. Another study showed that gender stereotypes of the individuals involved in development and the use of the I-methodology, which assumes technology as neutral, lead to gender issues in technology [7]. It suggests using methodologies that acknowledge diverse users, such as user-centered design [2], design for experience [147] and reflective design [148], to support gender-awareness and avoid perpetuating stereotypes that exclude of women from technology.

Regarding software development teams, we found evidence that gender diversity influences positively their communication [28], is a crucial element for performance [63], [62] and it is related to increased productivity [178]. These studies also evaluate subjective attributes (e.g., personality) and their relation to gender in team roles, for example, women with thinking personality and men with feeling personality were more suitable for team leader role. These findings can help forming a diverse development team, with appropriate roles for each individual, where gender perspectives can be openly shared.

Finally, we found a paper that discusses how agile software development environments support gender diversity [72] by comparing the methods used in the agile approach to women's style of management: both are focused on communication and collaboration during software development. The agile approach values individuals and interactions over processes and tools, and therefore can foster gender diversity in software development.

3.5 Conclusion

In this chapter we conducted a rigorous and auditable elicitation of the current state of the art about gender issues in software development: a systematic mapping study on gender issues in software engineering. By analyzing the selected studies and extracting the relevant information to answer our research questions, we gathered and classified a comprehensive set of existent gender issues in software systems, gender differences in attitudes, preferences and adoption of software systems and how these issues are being addressed during the development process. We also reviewed the existent approaches for addressing gender issues that served as groundwork for our solution. Specifically, we investigated how gender is understood in the software engineering field, what current issues exist in the software itself and what approaches have been proposed to solve them. We concluded that there are two main issues related to how gender is perceived during software development: software is assumed to be exclusively technical and neutral, while its development ignores the complex socio-cultural implications of gender, and when gender is incorporated in the development process, it is simplified and constructed as static and binary. Furthermore, we found a third issue concerning the limited existence of approaches for integrating a gender perspective during software development which constructs gender as a complex facet of the identity of users.

Additionally, we searched for studies that did not answer the main research question defined in the systematic mapping study, but were related to themes of this dissertation. This related work presented further insights into how software teams and environments are addressing the gender gap in technology towards diversity of ideas and inclusion of female perspectives. The knowledge regarding both individual and organizational gender issues served as basis and support for the development of our gender-inclusive requirements engineering approach.

Gender is a significant facet of an individual's identity and plays a fundamental role in their attitudes and behavior toward software technologies, as highlighted by the extensive literature gathered from the systematic mapping study and thus, is a crucial attribute to be considered for the development of inclusive software. With the gender issues in software engineering identified, we are now ready to begin tackling the concept of gender in the context of software development and develop a gender-inclusive requirements engineering approach.

4

Conceptual Modeling of Gender-Inclusive Requirements

Gender is a central component of an individual's identity and social life. As software technologies increasingly disseminate across private and public spheres of human life, it is fundamental to explore how gender intersects with such technologies. The importance of this research topic was confirmed by the systematic mapping study on gender issues in software engineering, discussed in Chapter 3, whose results demonstrated the existence of several gender issues in various software domains and thus the need for developing gender-inclusive software. In this chapter, we propose a conceptual model of gender-inclusive requirements. We resorted to the findings of the mapping study to identify and define the concepts and relationships that compose the model. The aim of this work was to tackle the three main gender issues we identified by developing both a definition of a common vocabulary and a conceptual model that standardizes these concepts to assist in the integration of a gender perspective in requirements elicitation.

4.1 Building the conceptual model

For the development of the conceptual model we followed an incremental three-step process based on the methodologies proposed in [45, 120]: specification, knowledge acquisition, and conceptualization. The process and the artifacts that resulted from each phase are depicted in Fig.4.1.

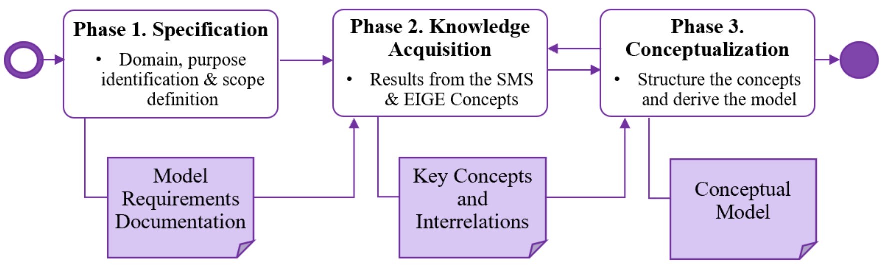


Figure 4.1: The process for developing the conceptual model

First, in the specification phase, we defined the domain and purpose of the conceptual model and present its scope; second, in the knowledge acquisition phase, we gathered the required knowledge for building the model; and finally, in the conceptualization phase, we classified and structured the previously collected domain knowledge to create the conceptual model. The last two steps were carried out simultaneously, alternating between the two according to the development progress. Moreover, we followed a middle-out approach when defining the concepts and their relations, as recommended in [176].

4.1.1 Specification

The specification phase involves two steps: the initial step concerns the identification of the domain and purpose of the conceptual model, and the second determines its scope. In the first step, we describe why the conceptual model is needed and what it will be used for, and by whom. This includes a description of the motivation for the development of the model, scenarios of use where it applies, and its range of intended users, written in natural language. In the second step, we identified the set of concepts and terms that satisfy the requirements of the conceptual model. In the following subsections, we present the results of both steps in detail.

Domain & purpose

The domain of the conceptual model concerns the intersection between gender and software technologies, exploring the mutually influencing relationship between gender and the software development and the software itself. It represents gender as a social construct and its relationship with software development and use within socio-cultural contexts. Its purpose is to represent the main gender issues we found from the mapping study and simultaneously mitigate them by supporting the development of gender-inclusive software for every user, regardless of their differences, through the elicitation of gender-inclusive requirements.

The model provides a common structured vocabulary for describing gender-inclusive concepts and the relations among them in software development. It is meant to be used as a tool for requirements engineers to elicit gender-inclusive requirements for software applications, services, and systems and ensure gender-inclusiveness is integrated from the beginning and throughout the software development life cycle. The model aims to assess the potential impact on users who are underrepresented in the process and ensure their individual and social characteristics (e.g., preferences, needs, perspectives) are consulted and not assumed. A summary of the results from this first step is presented in Table 4.1.

Scope

After determining the domain and purpose of the model, we defined its scope. The scope determines the range of knowledge that should be included in the conceptual model. In

Domain	Intersection between gender and the software development process
Purpose	Assist in the **elicitation of gender-inclusive requirements**
Types of Questions	Regarding gender issues in software development and interaction and how to develop gender-inclusive software through RE
Users	Requirements engineers, software engineers, stakeholders
Sources	Knowledge from the selected works of the systematic mapping study

Table 4.1: Summary of the first step of the specification phase

the following subsections we present the concepts that were represented in our model and those that were not included.

Gender conceptualization

Gender can be understood through a multitude of perspectives. To define the conceptualization of gender in our model, we resorted to the classification criteria from the systematic mapping study (Chapter 3). The conceptualization we intend to represent corresponds to the categories of *Social Construct* and partially to *Intersectional*. By partially *Intersectional* we mean gender is represented as one component of the many that build people's identities, but we do not include other social categories nor represent their intersection with gender.

The model represents gender differences in personal characteristics that influence interactions with software technologies, while biological sex differences that are not accounted for in software development remain outside the domain of the model. The gender differences in behavior extracted from the studies of the systematic mapping study only focus on individuals who self-identified as either female or male.

Moreover, the majority of the studies from where these differences were collected were previously classified into either the *Binary* or *Binary, Social Construct* categories of gender conceptualizations. However, because we intend to conceptualize gender as a *Social Construct*, the model includes the concepts of gender identity and gender expression as a spectrum to describe how gender shapes diverse behaviors and attitudes towards software technologies and go beyond the gender binary.

Furthermore, we represent gender as one dimension of an individual's identity, according to the *Intersectional* approach. However, in the context of this thesis, we do not include other possible dimensions of identity that are addressed by the studies that were classified into this category.

Finally, gender also represents the gendered structures and exclusionary contextual settings where software technologies are being developed. The pervasiveness of gender on organizations, communities, and stakeholders, and the impact it has on the requirements for a system and consequently, on the system itself, is represented.

Software application domains

As there are many different types of application software, each with distinct characteristics and purposes, we outlined the types of software for which our conceptual model applies by specifying the set of studies from where the knowledge was collected. We used the domain classification criteria from the mapping study to define the types of software our conceptual model can be used for.

In this dissertation, we focus on software applications, services, and systems, and therefore, we included the software types from the studies categorized as *Open Source Software*, *Social Networking Services*, *Game Development*, *Smart Cities*, *Mobility & IoT*, *Virtual Reality* and *End-User*, *Web* and *Mobile Applications* in the software domains scope of our approach. Both the *Human-Computer Interaction* and *Requirements Engineering* domains refer to research fields whose studies investigated gender issues in different software types or software in general, and thus, they were also included.

The second most predominant domain from the eleven categories was *Machine Learning*. In this dissertation, however, we did not address Artificial Intelligence, Machine Learning, and Chatbot systems due to their unique nature of development, application, and effects. We considered them to be a particular subgroup of technologies that would require a more focused and appropriate investigation to create the methods and tools to address gender issues during their development.

Moreover, during the planning phase of the systematic mapping study (see 3.1.1), we narrowed our search by excluding studies that addressed gender issues in educational software (e.g. e-learning platforms). Accordingly, we did not include concepts related to this category in the scope of our model. The domains are further presented for each corresponding gender differences in Section 4.2.

Human actors

Gender-inclusive software is any software system that was developed considering a gender perspective throughout development to produce software that supports the needs of diverse user populations, regardless of their differences. Hence, the model includes the concept of 'Human Actor' that represents the people who are related to the software system. Given the scope of software domains of our research, the human actors of the model are stakeholders and include users, end-users, end-user developers, software engineers, requirements engineers, and costumers. We focus on addressing the currently known underrepresented and under-served human actors in software development, intending to guarantee an inclusive software system for every user.

4.1.2 Knowledge acquisition

The objective of the knowledge acquisition phase is to identify, collect, and define the knowledge required for building the model according to its purpose. Our model intended to describe the existent gender issues in software development and provide a solution for them through the integration of a gender perspective. Thus, we gathered unstructured, semi-structured, and structured knowledge regarding gender in software development. We resorted to the results and findings of the systematic mapping study (Chapter 3) and to the European Institute for Gender Equality reports to build the glossary of terms of our model. For the gender and related concepts, we adopted definitions of the EIGE Glossary & Thesaurus[1] to ensure consistency with already well-defined terms [176].

We used the qualitative data from the data forms (Appendix A) and applied the text analysis technique [45] to the selected studies and formal documents. Furthermore, we adopted the 'middle-out approach' [176] to start with the most general and descriptive gender-related concepts, and then proceed to generalize or specify them when required. This approach allowed us to continuously verify the concision (i.e., all terms are relevant and there are no duplicate terms [45]) and completeness of the glossary throughout the process.

Finally, as a result of the previous comprehensive analysis of the collected information, the concepts and relationships were arranged in a semi-formal taxonomy form. The resulting taxonomy served as the foundation for structuring the domain knowledge into the conceptual model and it consists of four main concepts: *Gender*, *Sociocultural Context*, *Human Actor*, and *Software System*. These are the fundamental concepts for the definition of the *Gender-Inclusive Requirements*.

4.1.3 Conceptualization

In the conceptualization phase, the conceptual model is developed based on the constructed taxonomy. This phase is intended to structure the key concepts from the acquired knowledge and their interrelations into the conceptual model [45]. Based on the collected data, a first draft of the model was constructed. This process was subject to continuous evolution by alternating with the previous phase in order to review and refine the definitions of the concepts and their relationships.

One of the main goals of this phase was to reach a balanced common ground where the definitions of gender-related concepts were clear and objective enough to be understood and interpreted by the individuals involved in the software development process, while not being simplified in such a way that would neglect the complexity inherent to the concepts necessary for their understanding. In this sense, we chose UML for the construction of the model as it is a well-known modelling language in software engineering that allows a visual description of concepts in a specific domain of knowledge.

[1] https://eige.europa.eu/thesaurus/overview

4.2 A conceptual model for gender-inclusive requirements

The conceptual model of gender-inclusive requirements was constructed as a UML class diagram, shown in Figure 4.2. For clarity, we omitted the characteristics where gender differences where identified and their interrelations (see Table 4.5). It consists of 52 concepts.

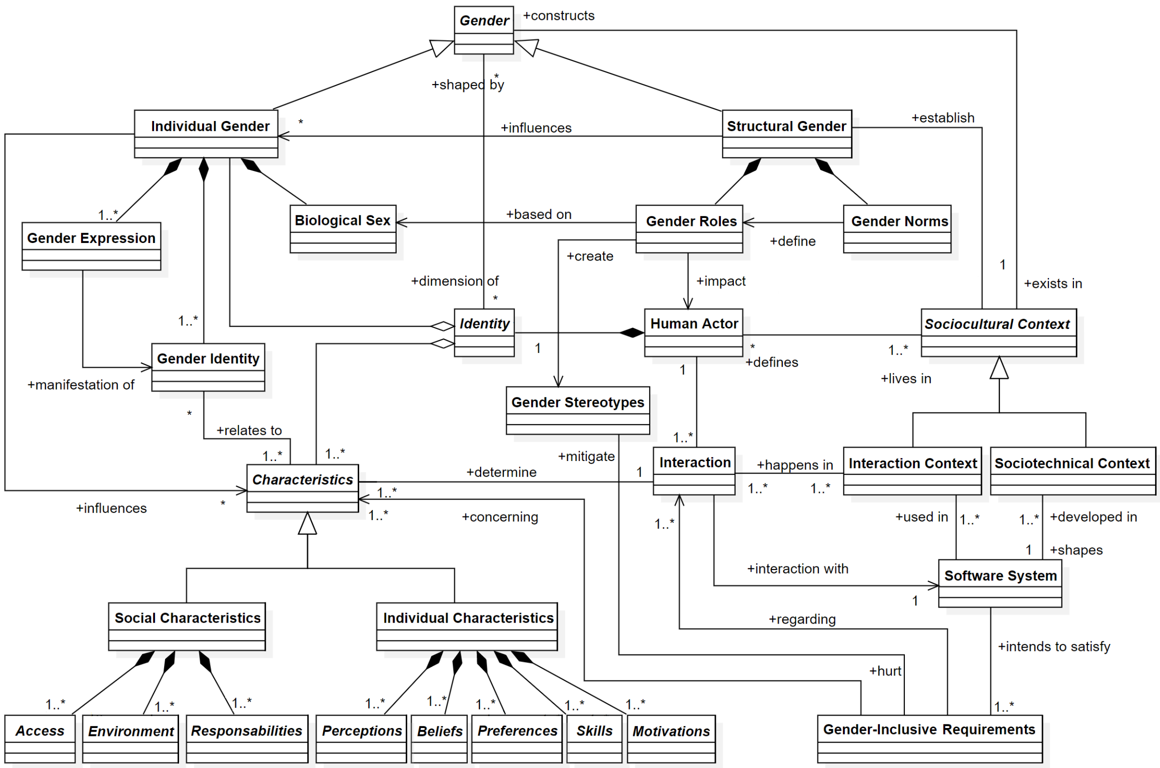


Figure 4.2: Conceptual model of gender-inclusive requirements

The model is composed of the following four components:

- **Gender.** Covers the concept of gender through two different but complementary perspectives: as a subjectively held self-identity and self-presentation to others, captured through the concept *Individual Gender*, and as an imposed set of norms and roles imposed by society which are socially constructed and maintained through performative acts and social conformity, represented by the concept *Structural Gender*.

- **Sociocultural Context.** Covers the sociocultural context where *Gender* is constructed and where *Human Actors* live in and are defined by. The sociocultural context of software development and the impact of the software itself as inherently social and the result of decisions that incorporate specific human values and objectives is represented by the *Sociotechnical Context*. Both *Individual Gender* and *Structural Gender* have an influence on the software development process and on the elicited requirements of system to-be developed, represented by the relationships between these two concepts and the concept of *Sociotechnical Context*. Simultaneously, the model includes the representation of the impact Gender has on users interacting with the developed system, the *Interaction Context*.

- **Human Actor.** Expresses the concept of Human Actors, who are subject to the gendered systems of the *Sociocultural Context* they live in and participate in social constructions of *Gender* by embodying and perceiving others based on the same gender-based norms. Therefore, they have differences in experiences, opportunities, roles, responsibilities, and levels of access and decision-making that will imply distinct needs and expectations from software technologies. Thus, human actors expect the *Software System* to be gender-inclusive and support their characteristics.

- **Software System.** Captures a software system as the product of human-based decisions in a *Sociocultural Context* influenced by *Gender*. Typically, it is developed in male-dominated environments where acknowledging different contexts is overlooked under the assumption of technical neutrality, and thus, it reflects the masculine culture and identity of its creators, at the expense of underrepresented perspectives. Therefore, the system intends to satisfy the *Gender-Inclusive Requirements* and support all its users, regardless of their *Characteristics*.

In the following subsections, we present each component of the model and detail their concepts as well as their relations.

4.2.1 Gender, individual gender & structural gender

In this section, we describe the gender component of the conceptual model.

Gender refers to the "social attributes and opportunities associated with being female and male and to the relationships between women and men and girls and boys, as well as to the relations between women and those between men. These attributes, opportunities and relationships are socially constructed and are learned through socialisation processes. They are context- and time-specific, and changeable" [52].

These social and culturally constructed norms, opportunities, and expectations create gender roles that are imposed on an individual based on their perceived *Biological Sex* [52], which refers to the biological and physiological characteristics that determine whether an individual is female, male, or intersex [149]. Although biological sex and gender are interrelated, they are distinct and independent of one another [122].

A person's individual experience of gender defines their *Gender Identity*, which may or may not align with their biological sex [55]. The demonstration of their gender identity refers to their *Gender Expression*.

Gender Expression is an individual choice about how a person wants to communicate their gender, and it can vary freely at any time. It is independent of biological sex and includes body appearance, dress, speech, behavior, personal traits, and actions [54].

In the conceptual model, individuals are represented by the *Human Actor*. The *Human Actor*'s experience of gender, their *Gender Identity*, which is independent of their *Biological Sex*, and their *Gender Expression*, the manifestation of their *Gender Identity* constitute the *Individual Gender*.

The *Individual Gender* represents the *Human Actor*'s self-identified gender [137] and is one part of their *Identity* [145]. Both *Gender Identity* and *Gender Expression* are conceptualized as a continuum where the *Human Actors* can align themselves to describe how gender shapes diverse behaviors and attitudes towards software technologies and go beyond the gender binary [163]. Furthermore, the *Human Actors* can vary their *Gender Identity* according to the context of the *Interaction*, which allows the construction of fluid and complex identities [138].

The person's gender expression is also perceived and interpreted by others based on *Gender Norms*, that is, "standards and expectations to which women and men generally conform, within a range that defines a particular society, culture and community at that point in time" [57]. Women and men internalize and embody social expectations for gender norms and thus behave accordingly [57].

The concept of gender is constructed within a social context through social expectations and interactions and perpetuated by *Gender Norms*. These are defined and reinforced in a particular *Socio-cultural Context* through social expectations and interactions, creating gendered power asymmetries in society and constituting gendered systems that perpetuate inequality.

Gender systems, which involve economic, social, cultural and political structures [61], institutionalize gender inequality by establishing and perpetuating distinctive *Gender Roles* for individuals who learn them through socialization processes [162].

In this context, *Gender Stereotypes* emerge and are internalized by *Human Actors* that perpetuate them by conforming to the expected behavior and reinforcing it to others [19, 84]. A summary of the concepts presented in this subsection is shown in Table 4.2.

Because we conceptualize gender as a social construct, there is no inherent truth to the concept, and it only exists in context. Therefore, in the model, *Gender* is an abstract class that is defined in the *Individual Gender* and *Structural Gender* classes. In this sense, integrating a gender perspective should not be seen as a single process with a binary judgment of inclusive/non-inclusive requirements, but rather a much more nuanced discussion of what constitutes gender-inclusive software through at least two dimensions: *Individual Gender* and *Structural Gender*. The first is composed of the *Gender Identity*, *Gender Expression*, which can both be one or more accordingly to their definitions, and *Biological Sex*, which is only one. Provided that gender expression is a person's choice for expressing their gender identity, we modeled the relation between the two concepts as the ordered association *manifestation of*. The second, *Structural Gender*, is composed of *Gender Norms* and *Gender Roles*, which are associated through *define*, only exist as long as structural gender exists. This concept *influences* the many existing individual genders of human actors. The gender roles imposed on the human actor are *based on* their biological

Gender Structural feature of society and a component of a Human Actor's identity	**Individual Gender** A person's self-identified gender, which is one part of their identity. It is a continuum where Human Actors can freely align themselves and construct their fluid and complex identities	**Gender Identity** Human Actor's individual experience of gender, independent of sex
		Gender Expression Individual choice about how a Human Actor wants to communicate their gender identity, and it can vary freely at any time.
		Biological Sex Biological characteristics that determine whether an individual is female, male, or intersex
	Structural Gender Social and culturally constructed norms, opportunities, and expectations imposed on Human Actors based on their perceived sex according to binary essentialism	**Gender stereotypes** Determine what is expected, allowed, and valued in a woman/man and girl/boy in these specific contexts
		Gender Norms Standards and expectations to which women and men conform, within a range that defines a particular society, culture, and community at that point in time
		Gender Roles Internalized and embodied social expectations learned through socialization processes in which women and men behave accordingly

Table 4.2: Gender concepts and respective definitions

sex. In the following subsection, we further discuss gender inequality in the software development field and the consequences it has on *Software System* itself.

4.2.2 Sociocultural context, sociotechnical context & interaction context

This section presents the sociocultural context component of the conceptual model.

Technology has unarguably become a significant part of society and people's everyday life, exerting a growing influence on our personal and social experiences, opportunities, and behaviors. The increasing society-wide use of technology in several areas and for multiple purposes reveals how technology is agreeing to a variety of social interests and values. The use and application of technology within *Sociocultural contexts* confers it a social character where its evolution is not only motivated by technical improvement but also according to society's changes in needs and expectations [42, 186, 191]. These social and cultural factors and events of particular time periods are represented in the model as the *Sociocultural context*. A *Software System* is one such technology as it is created to serve a given purpose and acquires meaning depending on the *Sociocultural context* of its use [180], the *Interaction Context*.

The software development process, especially the requirements engineering phase, is determined by a range of social factors and contextual influences [32], such as the needs and values, represented in the model as *Characteristics*, of development teams, stakeholders, organizations, and communities, which are represented by the *Sociotechnical Context*. These values shape the decisions made during development regarding the intended *Software System*. Consequently, the final *Software System* embodies and reflects them, influencing, either positively or negatively, the community of individuals, *Human Actors*, who use it [81, 15].

Both software development and the impact of the final product are inherently social and the result of decisions and interactions that incorporate specific social values and objectives. However, the preferred needs and values of the *Human Actors* involved in software development might not reflect those of the software users because these individuals represent a very narrow subset of the population. While the user population, the *Human Actors* who interact with the software in an *Interaction Context*, can be highly diverse, the development teams, stakeholders, and managers responsible for the development and maintenance of open-source and closed-source software, the *Human Actors* who are represented by the *Sociotechnical context*, are overwhelmingly male and with high levels of education and income [185, 151]. Since the *Software System* is the product of the values and interests of the individuals and organizations directly involved in their creation, excluding a vast majority of potential users from participation in the decision-making process suggests the software may not match their needs, preferences, and values. The lack of diversity in software development across multiple dimensions raises concerns about whose values software systems embody.

The under-representation of women in software development points to a divide between the *Sociocultural contexts* related to the gender diversity dimension of software development, *Sociotechnical Context*, and software use, *Interaction Context*. *Software systems* are developed in male-dominated environments where acknowledging different contexts is overlooked under the assumption of technical neutrality. Thus, embody and reflect the male culture and identity of its creators [132], at the expense of underrepresented perspectives who have been historically excluded [145]. Specifically, it indicates a strong alignment between masculine ideals and values and the software's values [125, 138, 19, 5, 18, 140].

The historical patterns of gender discrimination, the consequent lack of gender diversity, and limited decision-making power from underrepresented *Human Actors* (see Chapter 2) restrict the discussion of different viewpoints. Likewise, it constrains the inclusion of a diverse range of perspectives, which may introduce, conscious or unconsciously, harmful assumptions and *Gender Stereotypes* about diverse perspectives into the *Software System* [101, 145].

Accordingly, software development is not impartial, and software is not neutral. The impact of these values needs to be understood and treated as properties of the *Software System* to avoid unintended consequences [119]. The diversity of social and individual contexts must be acknowledged and addressed early during the development process to ensure the integration of conscious and diverse social values into the software to make it inclusive for every user, regardless of their differences.

Therefore, we emphasize the distinction between the two contexts in the conceptual model to provide a clear understanding of how ignoring the *Sociotechnical context* of software during development may unintentionally lead to systems that replicate the existing structures of gender inequality in society. Table 4.3 summarizes the concepts presented in this subsection.

<table>
<tr><td rowspan="3">Socio-cultural context
The social and cultural factors and events of a particular time period that influence Human Actors</td><td rowspan="2">Socio-technical context
Range of socio-cultural factors and contextual influences, that impact the system</td><td>Stakeholders
Typically a very narrow subset of the population: mainly men with high levels of education and income</td></tr>
<tr><td>Organizations and communities
The sociocultural values of organizations and communities where the system is developed</td></tr>
<tr><td>Interaction context
Range of socio-cultural factors and contextual settings where users interact with the software</td><td>End users
People who use the software, can be diverse in behaviors, preferences, and needs</td></tr>
</table>

Table 4.3: Sociocultural context concepts and respective definitions

The information that will lead to the formulation of *Gender-Inclusive Requirements* can only be elicited through the understanding and specification of the context of technology use, the *Interaction Context*, which is itself deeply interwoven with understanding people's identities and everyday practices. These are represented in the model through the concepts of *Human actor*, *Identity*, and *Characteristics*, which we discuss in the following subsection.

4.2.3 Human actor, identity & characteristics

The concept of *Human Actor* represents the people who are involved with the Software System. The *Human Actor* lives in a particular *Socio-cultural context* and has a unique *Identity* that is characterized by their *Individual Gender* and *Characteristics*. The *Human Actors* are subject to the gendered systems of the socio-cultural context they live in and participate in social constructions of gender by embodying the *Gender Roles* attributed to their *Biological Sex* and by perceiving others based on the same social norms. Therefore, their *Identity* is shaped by *Gender*, and consequently, their *Characteristics* as well. The *Characteristics*, which are influenced by the *Individual Gender* of the *Human Actor*, represent personal attributes that determine their *Interaction* with the *Software System*.

The concept of *Characteristics* represents the individual and social traits of a *Human Actor* and entails the users' attitudes towards software technologies that were empirically found to be influenced by the user's gender. However, the intent of addressing them in our model is not to attribute users a female/male label and develop software according to categorical groups. Instead, it aims to understand potential attitudes of users and explore beyond the 'ideal' or 'universal' user, which currently implies a male user.

Therefore, we conceptualize *Individual Gender* as a spectrum where a *Human Actor* is not expected to be consistent with one gender identity for all the *Characteristics* but rather vary because these are the result of complex and contextual individual experiences. For instance, a *Human Actor* can have an information processing style statistically more prevalent among female users while showing attitudes towards risk more prevalent in male users, regardless of their gender identity. Hence, using personal attributes that are part of the individuals' identities preserves their individuality and complexity. Simultaneously it allows the adoption of a gender perspective that addresses gender issues without resorting to binary essentialist notions about women and men.

Accordingly, we can consider diverse groups of *Human Actors*, account for behavioral diversity among them, challenge binary assumptions and gender stereotypes, and encourage the representation of diverse perspectives so that all users benefit equally from the system when interacting with it. Table 4.4 presents these concepts and their respective summarized descriptions.

<table>
<tr><td rowspan="2">Human Actor
Individuals with a unique identity that are involved with the software system in a sociocultural context</td><td rowspan="2">Identity
The distinguishing and unique character and personality of a human actor</td><td rowspan="2">Characteristics
Part of the identity of a human actor and determine the interaction with the software system</td><td>Social Characteristics
Attributes of a human actor in relation to other human actors</td></tr>
<tr><td>Individual Characteristics
Personal attributes of a human actor</td></tr>
</table>

Table 4.4: The concepts of Human Actor, Identity, Characteristics, and Interaction

This conceptualization of the *Human Actor* follows the 'quality of pluralism' [5] that seeks to recognize the complex and unique identities of users across socio-cultural contexts to foster engagement with diversity and challenge the homogeneous points of view that underpin the assumptions made about users in the early stages of development. Pluralism encourages a human-centered approach that embraces diversity rather than universal solutions based on inconsiderate simplifications [5].

The quality of pluralism is the fundamental conceptualization of the *Human Actor* for understanding the concept of *Gender-Inclusive Requirements*. All *Human Actors* have distinct and unique identities that lead to different needs and goals regarding the system to-be. However, as discussed in subsection 4.2.2, the human decisions and values that constrain the technical considerations of the software systems are defined by an asymmetry between the *Socio-technical Context* and the *Interaction Context*.

The characteristics that are addressed, prioritized, and supported by the software system are those of the identified stakeholders for the system's requirements elicitation, the first stage of the requirements engineering process [32]. Men are overwhelmingly represented in this process and their *Characteristics* are, inadvertently or not, best supported by the *Software System* at the expense of overlooked ones [26]. Women are almost entirely underrepresented, and software systems do not support the characteristics statistically more prevalent in female users.

Human Actors who are underrepresented or fall outside of the assumed or envisioned user are disadvantaged and may have to adapt to the system to use it or not use it at all. Female users are the most harmed from the gender issues that result from this lack of support of the underrepresented human actors' characteristics. Nonetheless, these gender issues in software affect all users of any gender identity because no human actor is a 'typical' female or male user [25]. Human actors that interact with the software system have pluralist, nuanced identities, and thus all benefit from gender-inclusive software [179].

Awareness of the gender segregation of the stakeholders who participate in the requirements engineering process is crucial for eliciting gender-inclusive requirements. Therefore, we distinguish between the *Human Actors* who are stakeholders of the system, in the traditional definition of requirements engineering [154] and those who use it and are affected by it, but are underrepresented in the development process. The first are the stakeholders of the *Software System* and include, but are not restricted to, users, customers, regulators, requirements engineers, and software engineers. They are the individuals who compose the organizations and communities of the socio-technical context. In contrast, underrepresented human actors are either end-users who solely use the system and are not involved in its development process or stakeholders of the system but with limited decision-making power, lower status, or part of a minority in the organization or team. From a conceptual perspective, there is a fundamental reason why this distinction is necessary for defining gender-inclusive requirements.

Underrepresented stakeholders typically have lower levels of seniority or lower status and roles within the team or the organization. Thus, they can feel intimidated to express their opinions and perspectives as they may not feel safe to voice their ideas or that they are not heard. Specifically if it refers to a gender perspective, stakeholders may avoid sharing it as to not be categorized, victimized (i.e., unfair and negative treatment of someone for calling out a discrimination) or stereotyped [132]. In addition, if the environment and social context is male dominated, individuals may adopt the mainstream culture for adaptation and integration, and thus forget or avoid introducing gender perspectives [185]. That is, underrepresented stakeholders may find it easier to accommodate to the perspectives and views of the environment than to resist.

Therefore, the *Human Actors* who are stakeholders of the software system, in the sense that they are represented during the development process and have decision-making power over the elicitation, analysis, and prioritization of requirements, have the responsibility to work towards a more inclusive and open environment where all stakeholders can express their opinions and perspectives freely and without judgment.

Furthermore, it allows the discussion of the impact of the system they are developing for different users, leading to the creation of gender-inclusive requirements. To achieve this, *Human Actors* would use and integrate the conceptual model in their requirements engineering techniques to elicit information regarding the *Characteristics* of users who are underrepresented.

We organized the characteristics as either *Individual*, representing the personal attributes of a *Human Actor*, or *Social*, representing the *Characteristics* that are established in relation to others. These two concepts were introduced in the model after the collection of the gender differences in attitudes towards software as a way to classify and organize them in the model. This classification facilitates the selection of the most suitable *Characteristics* for the type of *Software System*. *Individual Characteristics* include *Perceptions* (Perceived Risk, Perceived Financial Cost, Perceived Security, Perceived Ease of Use, and Perceived Usefulness), *Beliefs* (Credibility, Trust, Cost-benefit, Self-efficacy, and Sense of belonging), *Preferences* (Linguistic and Communication Styles and Visual Design), *Skills* (Cue Detection, Information Processing Style, and Awareness), and *Motivations* (Willingness to learn, Motivation, and Tinkering). *Social Characteristics* include *Responsibilities* (Time commitment and Routine integration), *Access* (Access to technological resources), and *Social Environment* (Social Interaction and Community). The set of characteristics that influence the users' attitudes towards software are shown in Table 4.5.

In the following two subsections, we detail the empirical evidence on gender differences found for each characteristic that impact users' interaction with software systems. The data we present consists of statistical differences between female and male users (see Section 4.1). However, we conceptualized this information in the model as two ends of a continuous sequence, a spectrum, for each characteristic. One end of the spectrum denotes the instantiation of the characteristics as observed statistically in female users and the other end denotes those of male users. In between, there is a range of variability and diversity of instantiations for that characteristic. For instance, men are statistically more likely to have higher self-efficacy, that is, to believe they have the ability to complete tasks in software systems in various situations. On the other hand, women have less self-efficacy, believing they might not be able to complete a task in the software system effectively, when in fact they can. Thus, the concept *Self-efficacy* is plotted with *Lower Self-efficacy* on one end and *Higher Self-efficacy* on the other, with varying degrees of self-efficacy in between, as illustrated in Fig.4.3.

The wide range of instantiations is where most human actors align themselves. Thus, software that is intended to be gender-inclusive must support everyone across each characteristic's spectrum. Resorting to the above example, human actors do not have exclusively "lower self-efficacy" or "higher self-efficacy". Mostly, they have varying degrees of self-efficacy. By addressing the two ends, it is intended to support every potential user across

Characteristic	Description	Concept
Risk	Perception of possible outcomes when using a software [26]	Perceptions
Financial Cost	Perception regarding the financial cost of a software [99]	Perceptions
Ease of Use	Perception that using the technology will be effort-free [83]	Perceptions
Usefulness	Perception that using the technology will provide utility [83]	Perceptions
Credibility	Level of credibility attributed to a software [126]	Beliefs
Trust	Belief in the reliability and trustworthiness of a software [99]	Beliefs
Privacy	Concerns and behavior for privacy when using a software [143]	Beliefs
Cost-benefit	Earned benefit compared to the cost of trying a software [173]	Beliefs
Self-efficacy	Belief in the ability to use software in varied situations [26]	Beliefs
Sense of Belonging	Feeling of fitting in with an online culture or community [112]	Beliefs
Linguistic & Communic. Style	Linguistic and communication styles in an online community / website interface / software [112]	Preferences
Visual Design	Aesthetics of the software interface, including imagery, colorfulness, complexity, and fonts [126]	Preferences
Cue Detection	Cue detection in interface design, language, and community norms of a particular software [126]	Skills
Information Processing Style	Strategies for processing new information and solving problems in a software task [26]	Skills
Awareness	Previous experiences and knowledge about a software [47]	Skills
Willingness to Learn	Desire to acquire knowledge about a software [10]	Motivations
Motivation	Reasons behind one's behaviors towards software [26]	Motivations
Tinkering	Exploratory behavior when using a software [26]	Motivations
Time Commitment	Time one has available for using a software [48]	Responsibilities
Routine Integration	Software compatibility with one's habits, behavior, patterns, and environments [94]	Responsibilities
Access	Access to technological resources [94]	Access
Social Interaction	Type and quantity of social interaction provided by an online community / software [99]	Environment
Environment	Conditions for interacting with the software [8]	Environment
Community	Size, culture, and environment of an online community [47]	Environment

Table 4.5: Characteristics that influence attitudes towards software

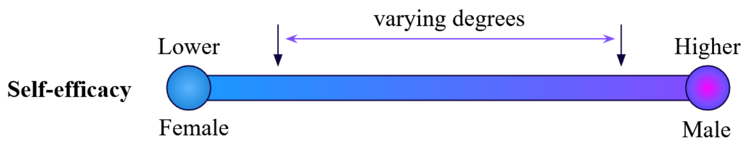


Figure 4.3: Conceptualization of the self-efficacy characteristic

the range.

At present, the *Socio-technical Context* is addressing and prioritizing the *Characteristics* of just one end of the spectrum, which is currently those that are statistically more prevalent in male users. As a result, gender issues arise in the created software systems. We further discuss the specific gender issues associated to each characteristic, their interrelations, and the effect they have on the interaction and on the user.

4.2.4 Individual characteristics

The *Individual Characteristics* encompass the *Human Actor*'s personal attributes, and include *Perceptions*, *Beliefs*, *Preferences*, *Skills*, and *Motivations*. Human actors have complex identities that result from their experiences in life, which also include the influence of their individual experience of gender. This experience, highly influenced by the *Sociocultural Context* they live in and the respective *Structural Gender* established, is deeply felt by the human actor and expressed through unique, context and time specific attitudes. These are manifested in their attitudes towards *Software Systems* when *interacting with* it.

Perceptions

The *Perceptions* of a *Human Actor* consist of the following five concepts: *Perceived Risk*, *Perceived Financial Cost*, *Perceived Security*, *Perceived Ease of Use*, and *Perceived Usefulness*. The first refers to the perception of the possible outcomes when using a particular software technology, that is, the combination of uncertainty plus seriousness of outcome [173]. A user's perceived risk plays an important role in their decision about whether to use a software system or particular features of a system. Risk perception was found to influence attitudes towards problem-solving software [22, 11], e-commerce applications [49, 76], and social networking services [97]. Gender issues arise because software usually requires a certain level of risk from its users to fully be used, harming those who are more risk-averse, who statistically tend to be female [26]. However, risk-tolerant users, typically male users, are not aware of the risk perceived by risk-averse users. By providing more information and safe options for risk-averse users, the software can benefit them without impacting risk-tolerant users, and thus, benefiting all its users.

The second, financial cost, is defined as the perceived monetary cost of acquiring and adopting a new software system by a human actor, which can be a critical factor during decision-making [99]. This concept was found to influence attitudes towards mobile systems, where female users have more concern and are more sensitive of purchasing cost, while males demonstrated less financial stress [99]. The third concept represent the differences in perceived security of a system and what behaviours result from these perceptions. In [109], gender differences were not found in security behaviours where less technical skills were required, such as enabling automatic updating of software, securing devices with passwords, and using pre-installed security software. However, when more technical knowledge was required, such as installing security software, enabling firewalls and keeping regular backups, gender differences were significant [109]. Although female users showed higher levels of information privacy concerns and believed that the consequences of a security threat would be worse for them, their overall levels of security assurance behaviors and threat prevention were lower than those of males [109]. That is, lower levels of information technology knowledge determined lower level of security behaviours. Thus, this concept is related to the concept of *Awareness* in the model.

The perceived ease of use is defined as the the degree to which a person believes that using the system will be free from effort, and it influences the intention to use and the attitudes towards the system [83]. When interacting with a new technology, women are statistically more influenced by their perception of how easy it is to use it [8]. For women, ease of use is a major contributor to use intention and it is viewed as the primary factor when deciding to use a system or not. Moreover, perceived ease of use is related to perceived usefulness [95]. For men, perceived ease of use is less important when since they tend to be more familiar with software technologies, features, and innovations [8]. Finally, the perceived usefulness is defined as the extent to which a person believes that using a particular software system will enhance their performance [83]. It also influences the decision whether to use or not the software system.

Beliefs

The *Beliefs* also consist of five concepts: *Credibility*, *Trust*, *Cost-benefit*, *Self-efficacy*, and *Sense of Belonging*. The first, *Credibility*, conceptualizes the level of credibility attributed to a particular software technology [126] by a *Human Actor*. Statistically, female users were more critical of and cared more about visual design when assessing the credibility of a system, attributing it more credibility when it made use of professional images, fonts and a multi-color themes [126]. Inadequate palette of colors in the user interface may cause doubts in the website's credibility [126]. In this sense, credibility is closely related to the concept of visual design. Male users were less critical and cared more about usability features rather than the visual aspects [126]. Additionally, the concept of credibility is related to the concept of trust in a software system. Trust can be determinant to a female user's acceptance and adoption of a system as it is considered an optimal concern [99]. Statistically, they showed less trust in the software system and referred to the lack of trust in a community as a reason to restrict their contributions online as well [47]. Male users showed less concern for the need to trust the system to use it and regarded software systems more trustworthy than the female users [99, 94]. Nevertheless, increasing female users' awareness plays an important role in increasing their trust in the system [99].

The concept of cost-benefit in the context of software interaction refers to one's personal beliefs on the benefits that using a software system will be worth the cost of trying, that is, the comparison between cost of trying new feature/software/product and the possible benefit of doing so [173]. Statistically, men have higher levels of self-efficacy and awareness, and thus, the perception of the cost of learning a new system or feature is lower [173]. For women, whose self-efficacy and awareness are statistically lower, the perception of the cost of learning a new system or feature may be higher [8]. Cost-benefit is also directly related to the concept of risk perception: a users' cost-benefit evaluation and subsequent behavior is strongly influenced by their perceived risk of trying the new system or feature [173]. If the cost evaluation based on the perceived risks are too high in relation to the benefits the software may provide, female users may choose not to follow through with the action of trying an unfamiliar system or feature [8].

Self-efficacy refers to the beliefs in one's capabilities to mobilize the motivation, cognitive resources and courses of action needed to meet a given situation's demands. In the context of software use, self-efficacy translates to the person's judgment of their capabilities to use computers in a variety of situations [26]. Self-efficacy affects performance outcomes, influencing the success of an individual when performing a task. For instance, if a user believes they have a lower level of skills than they actually have, they may conclude they cannot accomplish or be successful at a task, and thus, they may not proceed to try [71]. Statistically, female users have lower computer self-efficacy than males, which can affect their behavior with technology, causing females to be less confident in their ability to complete tasks [11], to have lower feature usage [10], and blame themselves if there is a problem [26]. Males users have higher self-efficacy and attribute failures to the difficulty of the task. Furthermore, technically skilled female users also demonstrate lower self-efficacy: they mention lacking adequate qualifications, lacking comfort with expertise and not being confident in their abilities to interact in the community as the main barriers for participating in online programming communities [47]. The concepts of self-efficacy and cost-benefit are directly related: due to female users' low self-efficacy, they may believe that the cost of learning a new feature is higher than it would actually be [173]. In addition, self-efficacy may have direct effect on the ease of use and interfere with the user's willingness to explore (*Tinkering*).

Finally, sense of belonging or ambient belonging refers to the feeling of fitting in or not with a culture or community that is passively elicited by one's surrounding environment. In the context of software interaction, one's sense of belonging refers to the feeling of fitting in with an online culture, service, or community that is passively elicited from social environment in which one is engaging with, for example, through cue detection [112]. The users' sense of belonging is also one of the most decisive concepts for the formulation of gender-inclusive requirements as the gender bias that result from it could unconsciously discourage women from "taking STEM courses, applying for a technical job, or voicing their opinion online" [112]. Sense of belonging is related to the concepts of visual design, linguistic and communication styles, cue detection, and environment. Stereotypical masculinity in interfaces and online environments and communities negatively impacts female users' sense of belonging: they anticipate lesser success, lower levels of confidence in their technical abilities, and less interest [47, 112]. Additionally, because they are typically a minority in these environments, female users may feel anxious about the way their gender would be perceived. For instance, female users may be reluctant to speak up to avoid being categorized, victimized (i.e., unfair and negative treatment of someone for calling out a discrimination) or stereotyped [132]. One of the most common and persistent demonstration of stereotypical masculine culture in the software development field is related to the 'geek culture', characterized by the idolization to the computer geek, antisocial and nerdy male [104]. For women who are software developers, 'geek culture' discourages them more than men and promotes expectations of male success and continual questioning of their abilities [47]. In this environment, female software developers may not feel welcomed and therefore may not want to engage and contribute to open source software. The gender issues that result from the concept of environment are further discussed in its subsection.

Preferences

The concept of *Preferences* includes the *Linguistic and Communication Styles* and *Visual Design* preferences of users. The first refers to the preferred and expected linguistic and communication styles in an online community, website interface, or in a software [47, 112, 79]. The gender issues that arise from this concept can have particularly harmful consequences if not addressed, as they can effectively discourage participation and action-taking through software technologies. One potential gender issue in language and communication style stems from the use of binary language, such as using "both genders" or describing users as opposites to each other, and gender-exclusive language, that is, using 'he/him' pronoun as default or 'guys' and similar phrases when referring to a group [16, 19]. Gender-exclusive linguistic cues can indicate a negative, segregated, and exclusionary environment which decreases individuals' perceived sense of belonging and motivation to engage with the system [112]. Addressing this gender issue can also help reflect whether male users are being prioritized by positioning them as the default or as the 'normal, average' user. Gender-inclusive language should be adopted as well as sensitive wording, inclusive terminology, and the users' preferred pronouns. Additionally, the use of gender inclusive language encourages a comfortable environment in online communities [47, 16]. Ignoring this concept results in gender issues related to masculine gender-exclusive and sexist language. The concept of linguistic and communication styles underlies many other concepts, particularly three concepts included in the social characteristics, online environment, interaction and community, which we discuss in their respective paragraphs.

The second concept, *Visual Design*, represents the aesthetics of the interface, including imagery, colorfulness, complexity, and fonts [126]. Visual design can unconsciously impact users' attitudes and intentions towards the system [112]. There are two gender issues that result from this concept: one is related to gender differences in preferences where those statistically more prevalent in female users are not taken into consideration, and the other is related to the integration of visual gender stereotypes in the system, which negatively impacts all users, especially female users. The former is a consequence of ignoring gender during interface design under the justification that the design is 'neutral'. However, this decision usually results in masculine preferences being adopted. Statistically, male users care more about usability features than the visual design of a system's UI [126]. Still, they showed a preference for symmetrical website layouts [112]. Although female users are not affected by symmetry [112], the visual design of an interface is of considerable importance for them. Female users are more visually discerning and prefer interfaces with higher visual complexity and colorful websites [112]. Additionally, they are more sensitive to and critical of the use of color in general, which makes them more critical and sensitive to UI changes and more responsive or sensitive to UI upgrades [126].

The aim of taking a gender perspective regarding this concept is to create a visual design that is appealing to all users by considering how complex and nuanced users can be, and thus ensure the system is not lowering any group of users' participation or motivation to engage because of aesthetic and design choices. Furthermore, the visual design is susceptible to gender stereotypes when integrating a gender perspective is misunderstood or is not at all. Essentialist perspectives about gender lead to assumptions about women and girls as an homogeneous group, specially if women are not present or are a minority in the development process. As a result, harmful stereotypes can be integrated into the software, for instance, the 'shrink it and pink it' practice [138] creates pink, simplified technologies (e.g., fashion and wedding video games [77]), that impose and reinforce traditional notions of femininity on women [18]. Inscribing stereotypical feminine visual design in the interface of the software perpetuates gender roles and marginalizes female users. Moreover, the predominant masculine culture in software development may result in stereotypical masculine visual elements being adopted into the interface, which can make potential users who do not identify with such design feel like the software is not meant for them and decrease their motivation for interacting with it. This design negatively impacts the majority of potential users, including men, who are also more positively affected by the gender-inclusive design [112]. Therefore, normative assumptions based on the gender binary, such as 'men like blue, women like pink' should be challenged. Inclusive and diverse user interfaces are appealing to and benefit all users, contributing for a more positive interaction.

Skills

The *Skills* of a *Human Actor* include three concepts: *Information Processing Style*, *Cue Detection*, and *Awareness*. In the context of software use, information processing style is defined as individual differences in how people problem solve and use software features to accomplish a specific task, that is, how they process new information to solve problems in a software system [26]. Statistically, female users process information in a comprehensive manner, that is, they gather information comprehensively to try to form a complete understanding of the problem before trying to solve it [26]. Additionally, female users make elaborate inferences in order to make a decision whether the problem is simple or complex [179]. On the other hand, male users avoid comprehensive information processing and use it only if a complex task requires it. Instead, they process information in a selective and heuristic manner [26] (e.g., follow the first promising information, and if required then backtrack [179]). Each process has its advantages for solving a specific task but only if the software environment where it is being solved supports it [22]. Otherwise, it can significantly reduce the effectiveness of the approach. Indeed, software features are designed around the way male users tend to use software systems and they are more supportive of the information processing style statistically more prevalent in male users [26]. This gender bias in the software interfaces and workflows is reducing female users' effectiveness and undermining their problem-solving abilities [22], creating a gender gap in the users' use of the systems. However, gender differences in information processing styles are individual differences in cognitive styles that cluster by gender. Therefore, by accounting for all information processing styles and supporting all users regardless of which process they choose to follow, we are closing the gender gap and creating gender-inclusive software that is better for everyone.

Cue detection refers to the ability for detecting and assimilating social cues from community norms or visual and linguistic cues in the interface design and language of a particular software technology [126]. Thus, cue detection underlies the concepts of visual design, linguistic and communication style, social environment, interaction and community. A user's ability to detect and respond to cues also influences other characteristics, such as their sense of belonging, willingness, and information processing style. Because men are statistically more likely to process information heuristically than women, they beware specific cues that are highly available and highlighted in the main visual context, missing more "subtle" cues [68, 126]. On the other hand, women are statistically more likely to process information comprehensively, and thus they are more accurate in decoding cues and assimilate all cues available [159]. They consider the system's attributes objectively and subjectively by processing their associated psychological complex cues [126]. For instance, a stereotypical masculine visual design in a web interface can prompt social identity threat in female users and decrease their sense of belonging and willingness to engage with a community [112]. Similarly, they can feel excluded or marginalized in online environments if these communicate subtle cues of sexism, hostility, and negative criticism [112]. Identifying subtle gender biased verbal and nonverbal cues in the system's interface design, language, or community norms is fundamental to ensure the system is gender-inclusive. Although apparently harmless, they can have powerful conscious and unconscious impacts on users who have high levels of cue detection ability, which will influence their decision to interact or not with the software.

The last characteristic of the skills, *Awareness*, refers to the user's initial knowledge, experience levels, and awareness of potential usage features of one or more software systems, and it contributes to the user's interpretation of the interaction experience [47]. The dissemination of technical knowledge has been strongly affected by gendered social relations, traditional gender roles and gender stereotypes which have been creating significant barriers for women to acquire it since the mid-20th century [41]. Although technology has been incorporated in our daily lives, there is still a gender divide in digital skills that prevents women from participating fully in the digital society [101]. The concept of awareness is related to the concept of access: without access to digital resources, one cannot increase technical knowledge and higher experience levels. Women have less access to technology from a young age. For instance it was found that parents were more than twice as likely to have given a computer to their son than to their daughter [101]. Further in life, the development of digital skills follows the same gendered patterns, resulting in gender differences in technical knowledge and experience. Lack of awareness of features was a common issue for female users in tools with many features, which affected their interaction with the system [47, 107]. However, by observing others using these unfamiliar or new features raised their awareness and knowledge [47]. Finally, awareness is directly related to self-efficacy: higher levels of awareness contribute to higher levels of self-efficacy, and lower levels of awareness contribute to lower self-efficacy [127]. Thus, by raising female users' awareness, their self-efficacy will also be higher, which will eventually lead to more confidence in experimenting and usage of increasingly more technical and complex systems, advancing gradually their digital skills. Software systems should provide mechanisms that allow users with less digital skills to use it properly and fully to give them the opportunity to improve.

Motivations

Willingness to learn, *Motivation*, and *Tinkering* are the three concepts that comprise the concept of *Motivations*. The first characteristic is related to a user's willingness to try new and different features on an unknown or familiar software technology. It reveals the users' motivation levels and amount of effort they will make to use the software system. Thus, it affects their overall performance and perspectives of future performance enhancement [26]. Female users may be less likely to explore unfamiliar features, less willing to adopt new features and have a significantly lower acceptance of new features [173]. They may prefer to engage with familiar features in repeated usage to avoid wasting time learning new ones [25]. Male users are more likely to explore and try new features, while showing less anxiety when interacting with the system [25]. Willingness to learn is related to concept of motivation for using a software system.

A person's motivation for using a software system is based on the perceived benefits of using such system, that is, what the person values in the system, and can affect which software features users choose to use [26]. To develop gender-inclusive software, it is necessary to understand the users' motivations for interacting with it. If there are users who are not motivated to use it, the reasoning behind the lack of motivation should be analyzed. Statistically, female users tend to see technology for what they can accomplish with it, meaning that they were motivated to use it to accomplish tasks, whether personal or professional [11]. Social motivations were also important for female users, such as being available for close friends and family [116]. In the case of male users, their motivations are more likely to be related with personal interest and enjoyment of the system itself, that is, they are motivated by technology as a benefit per se, which corresponds to the software systems expectation from users [22]. Thus, to maximize the number of users, the system should provide them equal benefits whether they use it with a concrete purpose, as a source of fun, or both with varying degrees.

Finally, *Tinkering* can be described as exploratory behavior when using a software system that improves both motivation to learn and task performance [26]. Statistically, female users are more likely to prefer learning software features using process-oriented learning styles [26]. Although they are less likely to playfully experiment, when they do, they tend to it positively and reflect during the process, which predicts effectiveness in the task [14]. On the other hand, male users have a greater propensity to tinker. However, they tend to tinker without reflecting, and thus, tinkering negatively predicted effectiveness in the task [25]. Tinkering is directly related to the concepts of self-efficacy and risk perception: the higher the user's self-efficacy and lower risk perception, the more a user is willing to explore and to adopt unfamiliar features or systems [10]. If a user has lower self-efficacy and perceives higher risks towards a new system or features, they are less likely to be open to learn through exploration. The main gender issues that arise from not accounting for gender differences in tinkering is that software systems are being designed with the expectation that the user will engage in exploratory behavior to discover the features required to succeed at their tasks, which disproportionately disadvantages female users. Addressing this concept is fundamental as introducing a clear path for process oriented users does not stop the users who learn from tinkering from doing it with the system. Thus, users who do not use tinkering as their strategy will not be penalized, and the software will benefit everyone and their types of learning choices regardless of the strategy they choose to employ.

4.2.5 Social characteristics

When gender is not addressed, the 'default' users equals male users, and thus ignoring gender-based differences in concerns and experiences and the effects of gendered social roles. Gender differences in the division of labor, mobility patterns, access to resources of time, money, skills and technologies, decision-making opportunities, and relationships between them imply differences in their daily routines that should be accounted for to provide an equally beneficial system. In the model, the concept of *Social Characteristics* describes these differences in the context of the interaction with the software system. It represents the *Human Actor*'s social behavior and it is composed of *Responsibilities*, the life and work responsibilities, *Social Environment*, the preferred online social environments, and finally *Digital Access*, the level of access to technological resources.

Responsibilities

The concept of *Responsibilities* represents the human actor's life and work responsibilities. These are influenced by gender-specific roles that determine what is considered to be socially appropriate for individuals of a specific sex and to which individuals mostly conform [162]. These socially established norms create a gender division of labor that deeply affects the types of roles women and men are involved with [53]. Differences in assumed responsibilities contribute significantly to differences in concerns and necessities from a software system, represented by the concepts of *Time Commitment* and *Routine Integration*. The former is intended to represent the time a user has available for using a particular software technology, that is, how much time a user can spend on learning to use or enhancing their interaction with the technology [48, 187]. The latter represents the degree to which a new software technology fits with a potential user's existing values, experiences, and daily life needs and wants [94].

Within the division of labor, women tend to assume the 'triple role', namely the productive, reproductive, and community management (i.e., voluntary unpaid work that is an extension of the reproductive work, which involves the management of collective consumption resources such as health care and education) roles, culminating in more working hours which are usually fragmented and unpaid, and much less free time [53]. Women manage professional, family, and housekeeping responsibilities as well as leisure time activities [20], and consequently have much less time for experimenting, understanding or learning new features of a software system, discouraging its use and acceptance. In the programming field, the lack of time is also a major obstacle for female software developers to engage in online communities and to contribute to voluntary programming outside of working hours [47]. Furthermore, the concept of time commitment is linked to cost-benefit [25, 187]. When deciding whether or not to use the system, the perceived required amount of time in relation to the time the user has available outweighs the benefits of its use. Men, on the other hand, tend to only assume the productive or community politics role (i.e., officially-recognized leadership paid roles usually related to national politics) in society, both directly paid and with increased power and status as a result [53]. Thus, they have more free, leisure, and personal time to experiment, learn and interact with technologies [187, 20].

Gender differences in the assumed societal roles extends to the concept of routine integration, which is closely related to time commitment. Women have more challenges in managing their own and other people's needs [20], and therefore the integration of the system into their daily activity patterns, habits, and environments, as well as for supporting and caring for others (e.g., close relatives, children, friends), is crucial [38, 94]. Also, women were statistically more likely to see new technological options and services as tools to accomplish specific tasks, and thus their motivation for integrating them in their professional and personal routines is influenced by the degree to which they meet the needs of these tasks [26, 118] For men, the integration of the software system in their routines is less important. They have more free and manageable time and care less for emotional connection and caring for friends and family, and thus use the system as a tool to support them in their professional lives and as entertainment to amuse themselves [20].

Digital access

The concept of *Digital Access* has one characteristic, *Access to digital resources*, that represents the user's access to a particular software technology and to the technologies required for using it effectively [94]. The differences in access to technological resources is also conditioned by the socially-established gender roles discussed for the concept of responsibilities. Due to the uneven distribution of financial resources that result from the assumed roles, women have less access to technology than men [151]. Men download more video games and purchase more software, electronic devices, and computer equipment, which in turn increases their self-efficacy and awareness [151]. The gender divide in access and use of digital technologies constitutes a barrier for women's use, participation, and adoption of the system. Although the divide varies depending on the socio-cultural context, it persists globally [151]. Access to technology is essential for an active participation in present and future society.

Hence, addressing the gender issues that arise from this concept is one of the most fundamental goals for gender-inclusive requirements. The system's requirements should be sensitized to users who have low access to digital resources to ensure the developed system is accessible to all. By being equally accessible, users can use the system to develop digital skills, make connections, seek information, and be active members of the digital society. This constitutes an opportunity for the system to be an important tool in women's digital empowerment.

Social environment

The concept of *Social Environment* gathers the concepts of *Social Interaction*, *Environment* and *Community*. Although these three concepts are clearly distinct, they are closely related and mutually impact each other. The first, *Social Interaction*, refers to the type of social interaction the technology provides to the user, which affects user adoption and use of the system [99]. Female users were more socially oriented, and thus, preferred technology that enabled them to connect to others, specially family and friends [99]. As discussed, women tend to assume the 'triple role', giving them a feeling of overall responsibility and concern about interpersonal communication and care for others [20]. Therefore, their focus regarding social interactions is on maintaining already established relationships and being available for others [169]. For male users, social interactions are more related to social entertainment [169]. They showed less care for emotional connection to family and friends and for others, as well as less concern with being unavailable [20]. In addition, they favored status in social interactions [99]. In open source communities, the concept of social interaction was also found to influence the attitudes of female users: social barriers such as technical documentation issues, cultural differences, stranger discomfort, and lack of personal connections can discourage females from engaging in the community [107, 47]. The type of social interaction provided is influenced by the environment where it occurs. The *Environment* is defined as the characteristics and conditions for participating in a community provided by a particular software technology [113]. Female users prefer an environment of collaboration rather than competition [77], and gamification had no effect on their motivation or performance [107]. In open source communities, women feel discouraged to participate due to the competition and negativity of the environment, such as negative feedback, lack of possibility to ask for help without judgment, and unwanted criticism that attempts to demean and discourage their contributions [47]. Online environments should be equally supportive of all types of users regardless of their characteristics, that is, it should be designed in such a way that promotes openness to diversity and prevents abusive behaviours.

Finally, the environment is determined by the community. The concept of *Community* represents the size, culture, and norms of an online community [47]. This concept is of particular importance because if it is not addressed, it threatens women's willingness, contribution and participation in online communities. In open source communities, culture is unfriendly to women due to a 'geeky-macho mentality' [104]. Consequently, women who intend to contribute but do not relate to this stereotype may fear the size and negativity of the community and fear they lack the qualifications to participate [107]. Therefore, their perceived sense of belonging and motivation to engage decreases [19]. In addition, the unwelcoming atmosphere and the lack of women and familiar people can also be a reason to feel uncomfortable, specially if they are not aware of unspoken social expectations and programming community norms [47]. Furthermore, this type of online community is also characterized by masculine gender-exclusive language, normalized sexism and gender bias [19]. For instance, if the users' gender is not visible, the acceptance rate of pull requests from women is higher than those from men. However, if gender is visible, the acceptance rate of pull requests from women is lower than those from men [79]. Furthermore, in FLOSS communities, men position their notions of openness and code development as the authority, even as these notions contribute to women's exclusion, and simultaneously discredit the necessary mechanisms for women's inclusion [117]. These gender bias have several direct and indirect effects on women's participation in these communities which are currently not being addressed. Thus, women feel more comfortable in small size communities or sub-communities and concentrate their work across fewer and more familiar projects and organizations [47, 107]. Additionally, they are more likely to engage if they see other women doing it [114]. If open source development is meant to be truly an equal, democratic model of creation of software, it must provide an inclusive community with an environment that is safe and motivating for all users to increase their participation, performance, and specially, contribution.

4.2.6 Software system

This concept represents the *Software System* the *Human Actor* interacts with in a *Interaction Context*. The *Software System* intends to satisfy the *Gender-Inclusive Requirements* regarding the *Interaction*. The *Interaction* can happen in various *Interaction Contexts*, which can be distinct from the *Socio-technical Context* where the *Software System* was developed and where the *Human Actor* may or may not have participated. All human actors are part of the *Interaction Context*. Nevertheless, the *Human Actor* expects the *Software System* to be inclusive and support their *Individual Characteristics* and *Social Characteristics* for a successful *Interaction*. The concept of *Interaction* represents a point in time at which the *Human Actor* formulates the intention to experiment a particular software system or features, followed by its continued use and experience rationalization, and finally the decision-making whether to adopt or integrate them in their daily personal or professional lives. This concept includes the *Human Actor*'s interpretation of the experience and how successful they were at achieving their initial goals for interacting with the system. Hence, the information for the formulation of the *Gender-Inclusive Requirements* must concern not only the user and their personal attributes, but also the contextualization of users as individuals in a social environment. The *Software System* that results from the software development process holds a high degree of responsibility for its social impact, and therefore, requires thoughtful reflection to ensure the software is equally available, usable, and useful for different *Human Actors*.

4.2.7 Gender-inclusive requirements

The lack of representation of women in technology fields and the differences between women and men in terms of their relative position of power and knowledge in the software industry and communities result in biased *Software Systems* that disproportionately affect women. Gender issues arise early in the software development process but are continuously overlooked and poorly understood. The conceptual model provides a common language for understanding how these issues arise and how they can be mitigated by emphasizing requirements elicitation as a crucial phase of the software development process for setting the goals and direction towards the creation of gender-inclusive *Software Systems*.

The elicitation is a fundamentally human-based process [32] and thus, the *Socio-technical context* where it will be performed is decisive for both the elicited information for the definition of the *Gender-Inclusive Requirements*, and also for the *Software System* that is developed based on those requirements. That is, the information required for the formulation of the software's *Gender-Inclusive Requirements* can only be elicited in development environments where gender inclusion is, at least, intended. The individuals who participate in the requirements elicitation must support and commit to gender inclusion and diversity in their own organizations and communities for an effective elicitation. Therefore, gender-inclusive requirements also refer to requirements on the process of development and can constrain the decisions of the software development methods and techniques.

Regarding the requirements on the product, the relevant information for *Gender-Inclusive Requirements* concerns the multiple and diverse *Characteristics* of *Human Actors*, according to the *Software System*'s application domain. *Gender-Inclusive Requirements* can vary, depending on the *Interaction Context* where the *Software System* is used, and are not intended to be a set of fixed and separate requirements, but rather capture the needs and perspectives of diverse users and integrate them into the requirements of the software. This results in functional and non-functional software requirements. Thus, an analysis of whose *Characteristics* are being prioritized and which ones are being neglected is required to assess whether the decisions being made discriminate against users that are underrepresented during the software development process.

4.3 Summary

In this chapter, we proposed a conceptual model for gender-inclusive requirements based on the key gender-related concepts gathered from the EIGE glossary and the systematic mapping study. The model was developed with the goal of addressing the three main gender issues found in this study. It describes the concepts and relationships that expose how human-based decisions leverage the development process. Therefore, the software system is not a technically neutral product. It is subject to the concept of gender from both the individual and structural perspectives that are established by and through human actors in sociocultural settings. From this representation, gender is not a binary categorical variable but a complex social construct, which addresses the second main gender issue. Finally, the main goal of the model was to address the third issue that referred to the lack of approaches to address gender issues in the development process, namely, a lack of common and practical definitions of gender and a lack of tools and guidance for integrating gender in software. The conceptual model tackles these issues by providing a representation of gender domain knowledge that can be used as a resource for the elicitation of gender-inclusive requirements.

4.4 Conclusion

The proposed model is a first step towards identifying gender issues and eliciting information for the formulation of gender-inclusive requirements. In this sense, it provides a basis for analyzing the specific gender domain concepts and determining what knowledge must be integrated into the information of the system's requirements to achieve gender-inclusive goals. Furthermore, a process to guide or assist requirements engineers in introducing the conceptual model into their preferred techniques is necessary. Therefore, in next chapter, we propose a framework that serves as a guide for requirements engineers and stakeholders interested in gender-inclusion on how the conceptual model can be incorporated in the development of their software products.

5

A Framework for Eliciting Gender-Inclusive Requirements

In this chapter, we propose a framework as the first step towards the elicitation of gender-inclusive requirements, based on the conceptual model. The objective of the framework is to integrate the model in the established practices of requirements engineers. It includes a set of general guidelines and question-based checklists to elicit information for the formulation of gender-inclusive requirements for a software system. The application of this framework is illustrated using a simple case study in which we developed a prototype tool to support the process of using the framework.

5.1 GIRE framework foundations

The conceptual model of gender-inclusive requirements (see Chapter 4) provides a representation of potential gender issues in software development and software itself, their origin, and consequences. It captures the concepts and relations to consider in requirements elicitation for improving gender inclusiveness. In summary, it represents a first step into the integration of a gender perspective in the context of software development. As described in the model, this integration should be in the requirements' activities of the software development cycle, specifically in the elicitation. Therefore, we developed the GIRE (Gender Inclusive Requirements Elicitation) framework. The main goal of the framework is to assist requirements engineers in the integration of the model into their established practices.

The conceptual model is the backbone of the framework we propose. Its intent is to map each concept of the model to a set of requirements for the system. The framework supports the mapping of the concepts of the model by providing the following:

- The conceptual model and a glossary of its concepts.
- A document for each concept of the model that includes a set of general guidelines and question-based checklists that support the three-step process for eliciting information for the formulation of gender-inclusive requirements for a software system.

The output of the process proposed in the framework is one or more documents that include the set of selected concepts, the defined goals, and the questions to support the elicitation of this information in semi-structured interviews, questionnaires, or creativity techniques.

The framework includes the following key practices:

- Analysis and study of the conceptual model and taxonomy of gender-inclusive software development.
- Definition of gender-inclusive requirements through the four criteria that underlie gender-inclusiveness in the context of software development.
- Recommendations for supporting gender-inclusive goals before the elicitation process.
- Plan the elicitation of the requirements that, if respected, will lead to the gender-inclusiveness goals being satisfied, and consequently, make the software inclusive to the people who use it.
- Elicit gender-inclusive requirements that are adequate and relevant to the software application domain and are aligned with previously established goals.
- Produce clear and accessible documentation for the addressed concepts.

These should not be seen as an exhaustive and closed process that can be done independently of other activities in the software development cycle. The framework is meant to assist in the use of the model through the already established practices of the organization or community. It is a starting point for requirements engineers and stakeholders to discuss potential gender issues in software system's functionalities and work toward satisfying gender-inclusive requirements, which will make the software inclusive to the people who use it.

5.1.1 Applicability

Because the development of the framework was grounded in the conceptual model, the application scope of the model defines the framework's applicability limits. Thus, the applicability of the framework concerns the same software application domains as the conceptual model. Similarly to the model, the framework can be applied for software systems where gender is either directly or indirectly conceptualized. By software systems that directly conceptualize gender, we mean software that explicitly uses a gender conceptualization in its functionalities and features, such as an application that requires the user's gender for its use. On the contrary, indirect conceptualization denotes software systems that do not include or require any gender concept, such as an excel spreadsheet, but where gender was found to influence the interaction with its users.

The framework assumes the commitment and support for gender-inclusion and diversity by the organization, community, or group that will develop the software system. The use and application of the framework requires an intention to understand how gender inequality in the software development field creates biased systems, and therefore, expects the aim of creating software whose values are aligned with society's. Hence, it does not include specific motivations or recommendations for individuals or organizations who believe gender is not relevant for requirements engineering, who feel uncomfortable engaging with gender, or are opposed to work towards gender-inclusion and diversity. From this standpoint, the target groups of the framework are those involved in the various steps of the requirements elicitation process who support gender-inclusion and constitute, but not limited to the following: requirements engineers, software engineers, analysts, domain experts, and customers.

5.1.2 Framework criteria

The gender perspective provided by the model should be instantiated according to the application domain of the system and according to the organization and system's goals regarding gender-inclusion and diversity. These goals will guide the elicitation of the system's gender-inclusive requirements. These are only effectively elicited when the model's four dimensions and their relations are addressed. From these, we defined four general criteria that compose the foundations for the proposed framework.

These four criteria define the goals, the relations between them, and the issues that are intended to be addressed, while simultaneously ensuring they are clear enough to all concerned in the process. The artifacts provided by the framework are intended to assist with the elicitation of gender-inclusive requirements that satisfy these criteria.

Criterion One

Address historical patterns of gender discrimination to overcome them

This first criterion was defined from the relations between the *Gender* and *Sociocultural Context* components of the model. The goal of this criterion is to communicate objectively how historical patterns of gender discrimination in the software field created gendered structures and exclusionary contextual settings that still exist and need to be explicitly addressed to avoid perpetuating them to the software system.

Gender Issues: *Historical patterns of gender discrimination in the field.* Stakeholders may be unaware of the historical patterns of gender discrimination that prevented and disadvantaged women from keeping their participation in the software engineering field and how and why they still exist today. Gendered systems, institutions, and structures prevail in the software engineering field, reinforcing, and perpetuating gender-based discrimination and inequality. If historical patterns of gender discrimination are not recognized and addressed, they are reproduced through structural gender norms, such as institutionalized practices and socialization processes.

Recommendation: Analysing the mutual relationship between gender inequality in society and inequality in software is fundamental to comprehend how overlooking structural gender issues can unintentionally lead to a biased and a discriminating system that replicates these same structures of gender inequality in society. Stakeholders with more responsibility and authority should reflect on how they can use their power and resources towards the creation of inclusive spaces and systems. For example, how can they create and facilitate supportive and safe environments where others involved in the process can equally collaborate and share different points of view?

The satisfaction of this criterion can involve not only the requirements of the system but also the requirements of the process.

Criterion Two

Identify and remove gender stereotypes and gendered assumptions

The second criterion represents the relation between the *Sociocultural Context* and *Software System* components of the conceptual model. This criterion intends to prevent the elicitation of information that embodies unintended gender bias that can perpetuate gender inequality, both in the sociotechnical and interaction context. Ensure the elicited information for the system does not further real-world imposed restrictions, and if possible, prioritize the goals that can empower users, especially women, trans and non-binary individuals.

Gender Issue *Reproducing the patterns of gender discrimination:* Given the historical sociocultural context of the software development field and the consequences it still has today, such as the current lack of diversity and inequality in positions of power, gender stereotypes may arise, consciously or unconsciously. If gender is not addressed explicitly, these gender-based stereotypes and assumptions can be made and left unquestioned. Indeed, removing gender from the discussion inadvertently results in masculine perspectives, preferences, and experiences being embed into the system, disadvantaging users who do not possess the same characteristics.

Recommendation. Furthermore, this criterion can result in both requirements for the development process and for the software system itself.

Criterion Three

Incorporate multiple perspectives, preferences, and needs, and supporting everyone regardless of gender differences

The third criterion was reasoned from the relation between the *Software System* and *Human Actor*. The goal of this criteria is to create awareness of the differences in attitudes towards software systems that arise from experiencing gender in different socio-cultural contexts to accommodate them in this system without reproducing gender stereotypes.

Gender Issues *Neutrality as universal:* Stakeholders may perceive their own needs, preferences, and values are "neutral", and therefore universal and applicable for everyone and anywhere. Additionally, they may suggest an "average" user with no mention to gender. However, if the stakeholders do not demographically represent the users in terms of gender and/or the socio-technical context is male dominated, the information gathered for the creation of requirements will be focused on men's perspectives, needs, and preferences. Furthermore, there is the possibility of men's perspectives being perceived as neutral while those of people of different genders' (for example, women, trans or non-binary individuals) perspectives as gender specific. Stakeholders may position their experiences at the centre of the discussion without considering the influence of gender on them.

Recommendation. Propose to the stakeholders to engage in reflexivity: reflect on their own position in society (gender identity, education, socioeconomic status), the social context where the software system is being created, their own influence on this social context and the decisions that are made, and particularly, how they are situated in relation to users. Challenge stakeholders to reflect on how their characteristics influence their understandings and intentions, and how they can contribute to the process of creating a more inclusive software. The information gathered should be object of reflection by the stakeholders to discover explicit and implicit needs of users who were previously unthought of.

Criterion Four

Account for behavioural diversity within groups of different genders to recognize the complex and unique identities of potential users

Finally, the fourth criterion represents the relation between the components *Human Actor* and *Gender*.

Gender Issue. *Categorical essentialism* Some stakeholders may confound integrating a gender perspective with analysing differences between women and men in the interaction with software systems. This framing of gender as an inherent, static, and binary categorical variable aligned with sex limits the understanding of gender as a complex component of the users' identities and can reinforce gender stereotypes and traditional expressions of gender on users.

Recommendation. To challenge this perspective, encourage the stakeholders to embrace a more sensible, rich, human-centred approach and adopt the quality of pluralism to reflect on the diversity of potential users and diversity within users, as well as the broad range of life, cultural, social, individual experiences that influence their attitudes towards the system. Introduce the concepts of gender as a social construct in an accessible and clear way, and demonstrate the benefits of rejecting a single, universal point of view that categorizes users into focus groups. The elicited requirements should account not only for the different genders, but also for the differences within genders. Recommend stakeholders to think of gender as a spectrum where users align themselves willingly.

5.2 A process to use the GIRE framework

To satisfy the criteria described above, we propose a three-step process that should be integrated into the general requirements elicitation process. Table 5.1 summarizes the phases required for the elicitation of gender-inclusive information.

Phase	**Goal**	**Techniques**	**Key Roles**
Pre-elicitation	Create a strategy for gender-inclusion before the elicitation	Work sessions	Requirements engineers, software engineers, stakeholders, analysts
Plan the elicitation	Select the knowledge of the model, the techniques and stakeholders	Work sessions	Requirements engineers, analysts
Elicitation process	Elicit information for the formulation of gender-inclusive requirements	Interviews, Workshops, Creativity techniques	Stakeholders facilitated by requirements engineers

Table 5.1: Phases of the proposed framework

5.2.1 Pre-elicitation

The conceptual model emphasizes the system's requirements elicitation as the fundamental process for ensuring the software will be gender-inclusive from the beginning of the development process. However, a successful elicitation of gender-inclusive requirements requires that the four components of the model and their relations are addressed, that is, that the four defined criteria of this framework are satisfied. Therefore, before planning the elicitation, the framework includes a pre-elicitation phase that discusses recommendations for creating diversity and inclusion strategies that will support the process.

This phase of the framework intends to address the first two criteria: the relations between gender, sociocultural context, and the software system. Its goal is to discuss what software engineering methods and processes will be most appropriate to support a gender-inclusive elicitation effectively in the context of the software's application domain.

The integration of a gender perspective in the context of requirements elicitation, requires a deliberate and rigorous prior preparation by the responsible requirements engineers. The effectiveness of the framework activities will depend on the requirements engineers' preparation and grasp of the domain knowledge of the conceptual model. The training sessions should provide requirements engineers with the knowledge required to proceed for the planning of the gender-inclusive requirements.

Moreover, all participants in the framework process should be provided with training on the conceptual model and its glossary as well as to the four criteria in this framework. This means creating awareness on existent gender issues in software development and providing training on gender inclusion to mitigate, challenge and potentially overcome them. Ensuring that all who are involved are adequately aware about the gender-inclusion goals and underlying objectives of using this framework prepares the participants to actively and deliberately collaborate with requirements engineers.

Furthermore, training on gender issues should promote organizational support for gender inclusion and diversity. Perceived organizational support concerns the degree to which its members perceive that the organization regards them and values their work and well being [4]. In the context of agile software development teams, providing equal career opportunities and equal respect through organizational support and policies for women's advancement improves team work quality and performance [4]. Finally, training on gender inclusion can help ensure the satisfaction of the first two criteria as it provides team members the necessary knowledge of the domain to discuss a gender perspective appropriately. The outcome of this phase can be one or more work sessions with the responsible managers, requirements engineers, and software engineers to define the development tasks and techniques that are aligned with gender-inclusive goals.

5.2.2 Plan the requirements elicitation

In the second phase, requirements engineers plan the elicitation process. We propose a document with goals, potential gender issues, a number of practical recommendations and question-based checklists for each concept of the model to assist requirements engineers in integrating a gender perspective when planning the elicitation. The documents provide stakeholders with key concepts relating to the development of gender-inclusive requirements. In addition, we created a semi-formal but practical glossary that includes the concepts and relations of the model. These are intended to facilitate communication and improve decision-making in the elicitation process.

Requirements engineers should explore and develop an understanding of the conceptual model. Following this analysis, requirements engineers should agree on a gender perspective that should be specified according to the required system domain. This involves the selection of the concepts that will be addressed in the elicitation through their documentation, which we described next.

Concept selection

The first step in the planning phase is the selection of concepts that will be approached in the third and final phase. The concepts of the model are organized in a top-down structure, from which we created a checklist where the requirements engineers can simply look at the more general concepts, and then specify, if intended, according to the system domain. The more general concepts refer to the four main components of the model, the concepts of *Gender*, *Sociocultural Context*, *Human Actor*, and *Software System*, which are addressed through the four criteria. Then, the concept selection should be made based on the domain of application of the system to be developed. For instance, if a software system requires a direct conceptualization of gender, then the concept of *Individual Gender* should be further addressed.

Concept documentation

The documentation developed for requirements engineers consists of a document sheet with descriptions, goals, recommendations and a set of questions in plain text and written in English. The documents were structured to assist in semi-structured interviews, questionnaires, brainstorming or creativity techniques.

Each concept of the model has an associated document with following structure:

- **Description.** After the name of the concept, the document includes its definition as in the glossary of concepts of the model. This helps the identification and memorization of the concept quickly without resorting to additional documentation.

- **Goal.** A clear description of the purpose of the concept and what it intends to add to the elicitation process. These help requirements engineers and stakeholders understand the meaning of the concept. For the former, it represents a clarification of the purpose of the concept, which provides them with a more informed and comprehensive choice about its integration (or not) in the elicitation process for a given software system. For the latter, it helps them discern the intent of the concept to understand the type of information to share and describe in the elicitation. Nevertheless, the goals are general enough to support exploration beyond what is stated.

- **Gender Issues.** A brief and general description of potential gender issues that can occur if this concept is not addressed, including the consequences for the system and for its users. These help requirements engineers identifying possible gender issues in the information provided by stakeholders. In addition, they can help both requirements engineers and stakeholders to assess their assumptions and thoughts.

- **Recommendations.** A brief and simple description of the corresponding recommendations to consider for mitigating or overcoming the gender issues mentioned before. These recommendations provide a more thorough understanding of the questions and their importance from the stakeholders or anyone involved in the process. These are also intended to help requirements engineers avoid gendered assumptions from the information provided by stakeholders, such as introducing gender stereotypes in the requirements or reinforcing gender roles to the future users of the system.

- **Questions.** The customizable question-based checklist for assisting in the implementation of the recommendations. It consists of questions that address the potential gender issues and the recommendations for overcoming them. The questions are primarily intended to encourage an engagement with diversity, as well as drawing attention to the value and necessity of thinking of users as people with a wide variety of identities and experiences in specific contexts.

Alongside the questions and recommendations, the document has checkboxes and space for requirements engineers to write down the answers, reminders, and ideas.

Concept questions

The questions are intended to support the elicitation of gender-inclusive requirements in semi-structured interviews, questionnaires, brainstorming or creativity techniques. They were constructed in natural language, in English, and based on the descriptions of the empirical gender differences in attitudes towards software (see Chapter 4).

In the documents, the questions focus on the software system and what it can provide or mitigate to meet the goals of the respective concept. However, they can be adapted to reflect the objectives of the elicitation technique in which they will be used. For instance, the following question: *"Does the system improve all users' motivation to learn new features equally?"* can be restructured to *"How would you learn a new feature in this system?"* or *"Do you feel more motivated to learn a new feature for what it can accomplish for you or more for enjoyment?"* so it can be integrated in a script of questions for a semi-structured interview.

Nevertheless, not all questions need to be addressed when a concept is chosen. Requirements engineers should select the questions best suited to their goals and system domain. These are not exhaustive but rather the starting point for a discussion about the potential attitudes of users and what the system should include to support them without privileging or neglecting any user.

5.2.3 Eliciting gender-inclusive requirements

Once the elicitation has been planed, requirements engineers can start eliciting information regarding the selected concepts. The documents from the previous phase must be used by requirements engineers during the communication with stakeholders to elicit the necessary information for the subsequent formulation of requirements. The questions should be used as a resource for collecting information directly from the stakeholders or creating further awareness and encouraging exploration of the concept.

Moreover, requirements engineers should act as facilitators to ensure transparency about how gender is discussed, and thus, identify potential gender stereotypes while collecting gender-inclusive information. The documents include the list of recommendations to help requirements engineers uncover both types of information.

5.3 A supporting tool for the GIRE framework

To illustrate the use of our framework we chose as case study the development of a prototype of web-based tool that supports the elicitation planning phase of the framework. The tool implements a clear workflow path, with three intuitive steps, for creating the gender-inclusive requirements document. This is the output of the process.

Concept selection

The purpose of this prototype tool is to support the framework's elicitation planning phase. We followed the process for using the GIRE framework to create a gender-inclusive prototype tool. Given the application domain of the tool, we selected the characteristics related to end-user development, in particular to problem-solving software: **Motivation, Self-efficacy, Information Processing Style, Tinkering (learning style)**, and **Risk Perception**. Gender is conceptualized indirectly (see Section 5.1). Thus, it was not necessary to address the concept of Individual Gender.

Question selection

After selecting the concepts, we selected the set of appropriate questions regarding the context and the goals of our tool.

Motivation We selected the following three questions regarding the concept of Motivation to further explore in the elicitation phase: i) Does the system offer step-by-step instructions? ii) Does the system present the information organized? and iii) Does the system provide clear workflow paths for task-oriented users?

For instance, given that the tool's objective is to support a process, the concept of motivation, in this case, allows us to understand how to make the process in the tool satisfactory for users who are task-motivated and also to explore how to make it more motivating for users who are more motivated by the technology itself than by the process or goal of the tool. We applied this reasoning to the remaining concepts.

Self-efficacy Regarding this concept we selected: i) Does the system provide feedback to the user about its state? ii) Does the system present clear error messages and instructions to possible solutions if a problem arises? iii) Does the system include tutorial materials always available for consultation? iv) Can the system increase the user's confidence about computing tasks (unfamiliar or not)?

Information Processing Style i) Does the system provide enough information for users to feel confident about their decision? ii) Does the system present all and only the information the user needs to complete their task? Is there information that could be perceived as ambiguous? iii) Does the system provide enough information for users to feel confident about their decision? iv) Does the system provide more than one path to achieve the same result? v) Does the system provide that information so that users who intend to form a complete understanding of the problem can do so straightforwardly?

Tinkering i) Does the system offer clear paths/steps for process oriented learning users? ii) Does the system provide a step-by-step process? iii) Does the system provide all the information necessary to complete a step in a task? iv) Does the system provide information about its features in the interface? Can the system make a user feel anxious from the lack of obvious information? v) Can the system cause frustration and disengagement in a user who has less willingness to tinker?

Risk Perception i) Does the system provide support for risk-averse users? Does the system require risk taking for users to successfully find/use/choose certain functionalities of the system? ii) Does the system provide options to continue even if a mistake was made? iii) Does the system provide options to return to the previous state without affecting the progress if a mistake was made? iv) Does the system correspond to the user expectations when they risk using it?

5.3.1 Elicitation

Following the framework's indications, we chose to adopt more creative elicitation techniques, which allow us to focus on users and, through them, understand what requirements the tool must satisfy. Thus, through the selected questions, we built a mind map on the possible attitudes of users towards the tool. In the center, we placed the users. Thus, through the selected questions, we built a mind map on the possible attitudes of users towards the tool. In the center, we placed the users. From this, we created five branches, one for each selected concept, where we placed descriptions referring to each one of them. Fig.5.1 shows a snippet of the mindmap for the Motivation concept.

Figure 5.1: A snippet of the mindmap created for the GIRE supporting tool

5.3.2 Prototype development

We implemented the prototype in React, a javascript library for building user interfaces. The landing page for the GIRE prototype tool is shown in Fig.5.2. It includes the possibility to read the instructions before using the tool for users with lower self-efficacy. A user with high self-efficacy can go directly to the workspace and start the process.

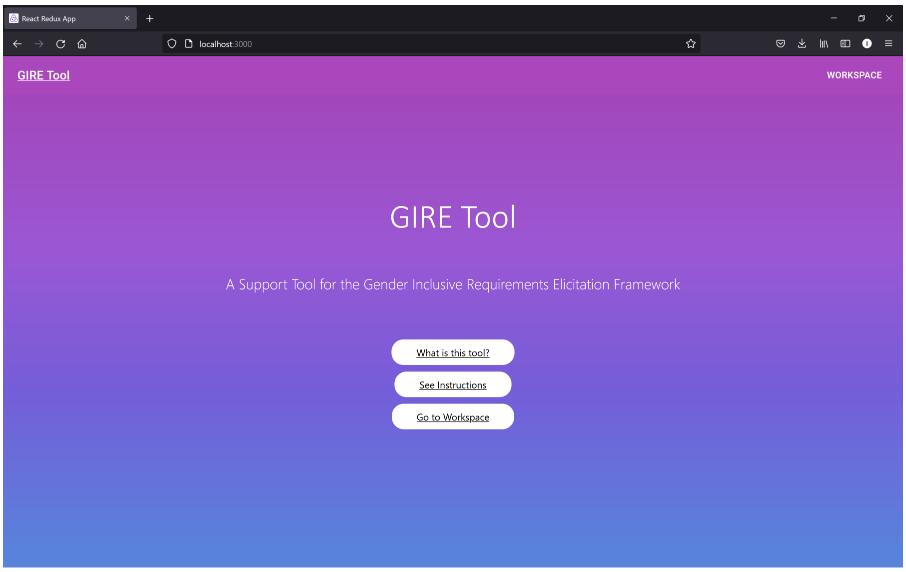


Figure 5.2: Landing page of the GIRE prototype tool

The Fig.5.3 and Fig.5.4 show the first step in the process. This first page presents the complete model so process-oriented users can see the full conceptual model and understand what concepts they want to select. In addition, for tinkering users, each concept is clickable so they can interactively explore the conceptual model and feel curious about its details.

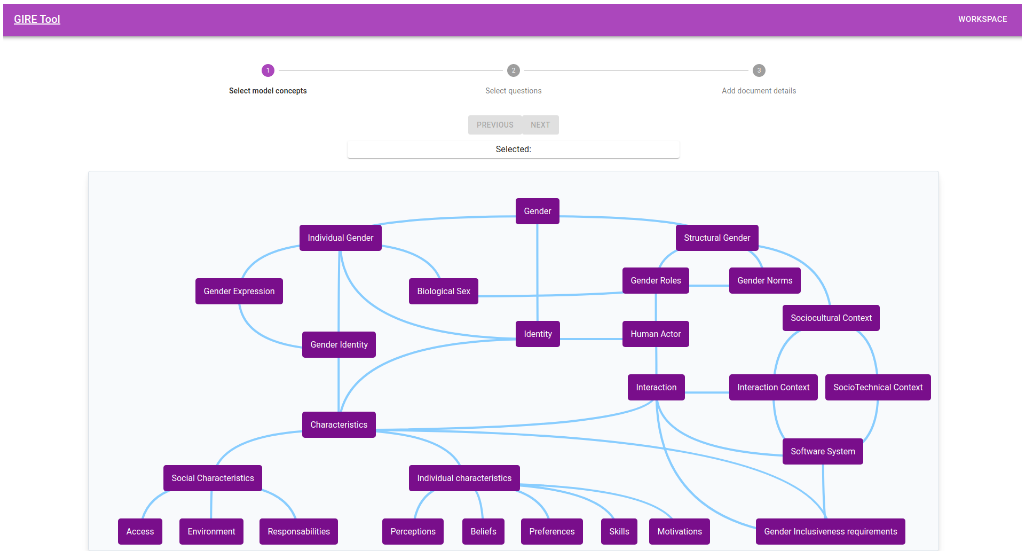


Figure 5.3: Selection of concepts from the conceptual model

The full details of the characteristics are only shown if the user selects that option. This allows the support for both lower and high self-efficacy users as for the first, it provides them with the full details they require to complete the process. For the second, they can skip the details for beginners, as they are optional, and complete the task effectively.

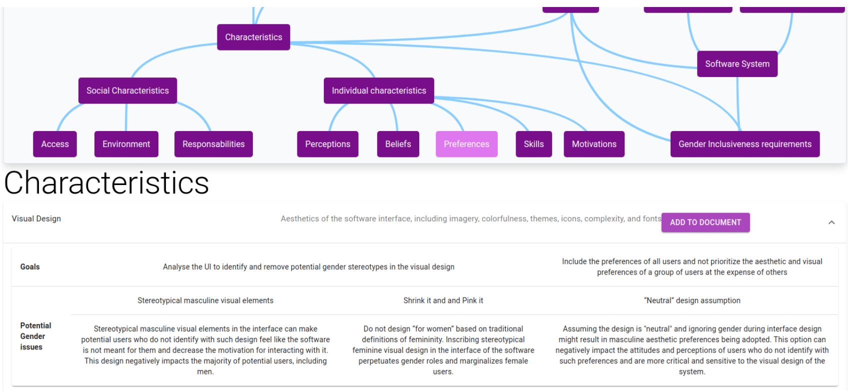


Figure 5.4: Full details of the selected concept

In Fig.5.5 is shown the process of selecting the questions. The top bar with the process indicates the step in which the user is at the moment. This can make users with lower self-efficacy feel confident about the step they are taking as they can see in what step of the tool process they are currently on, what they have done, and what steps follow. The "selection of questions" step only has the questions and selection options so task-oriented users can focus on that task only and feel motivated to finish this step.

In Fig.5.6, we show the page for entering the details of the requirements document. This is the last step of the process.

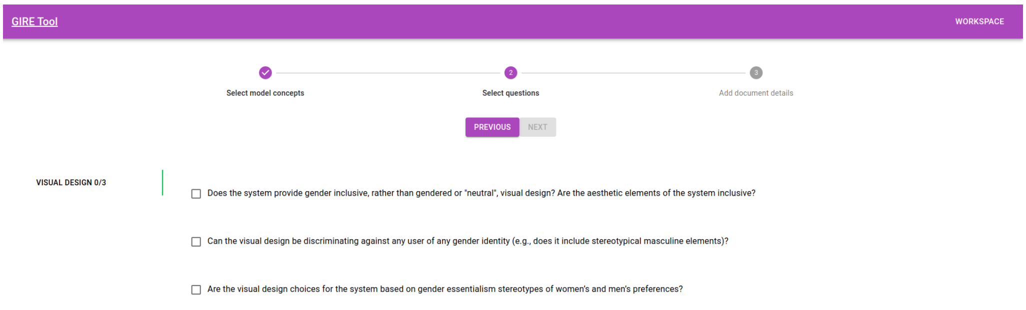


Figure 5.5: Questions' selection from the previously selected concepts

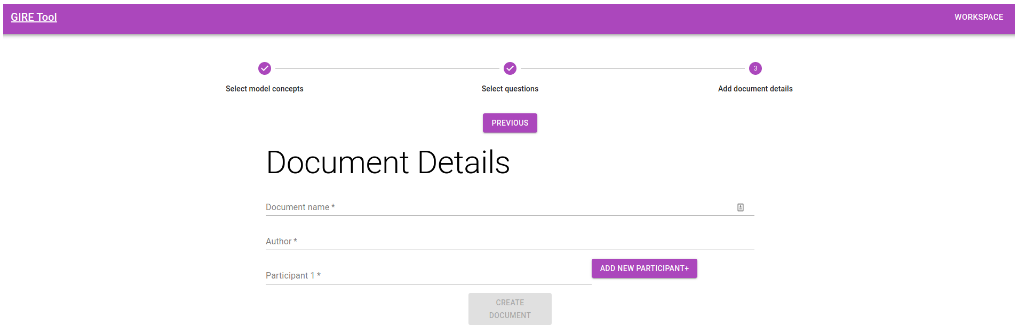


Figure 5.6: The form for document details

5.4 Conclusions

In this chapter, we presented the GIRE framework, a framework to integrate the conceptual model into requirements elicitation techniques. We first present the foundations for the construction of the framework, followed by a description of the process of using the framework. Finally, we present the development of a gender-inclusive prototype tool that supports this process. The prototype was developed through the application of the framework to be inclusive for all users, regardless of their characteristics.

6

Evaluation

This chapter presents the evaluation of the conceptual model for gender-inclusive requirements. The evaluation consisted of a static validation [189], that is, the proposed model was presented through a presentation guide to the participants who analyzed it and provided their opinions by filling in a previously prepared questionnaire. We start this chapter detailing the creation of the presentation guide and the questionnaire form, and describing our participants selection process. Then, we detail the results of the evaluation and provide a critical discussion of these. Finally, we discuss threats to validity of the performed evaluation and present the conclusions for this chapter.

6.1 Evaluation planning

The purpose of this evaluation was to collect viewpoints, from novices and experts, about the conceptual model for gender-inclusive requirements. The evaluation consisted of a static validation [189], which involves presenting and analysing a proposed solution without requiring it to be used [189].

To achieve this goal, we created a presentation guide that introduced the conceptual model to the participants and a questionnaire that collected both quantitative and qualitative data on the participants' viewpoints. The guide must be read before answering the questionnaire. The questionnaire collected the participants' opinions on the completeness of each of the four components of the conceptual model, the usefulness and difficulty of understanding of the model, its overall positive and negative points, and finally, if they would use and/or recommend it. Although gender inclusion has been receiving increasing attention in software engineering, the field is still introducing the topic. Therefore, we also collected the participants' perspectives on gender inclusion in software engineering in the questionnaire. In this way, we gathered a more comprehensive understanding of the opinions on the conceptual model and the reasoning behind them.

The questionnaire was created using *Google Forms*, a free web application that allows the creation and management of online surveys. In addition, the information collected through the survey is automatically entered into a spreadsheet. The guide was created using a *Microsoft PowerPoint* presentation. It was placed in a *Google Docs* folder as a PDF file and was shared in the form introduction through a link to the folder. In the following subsections, we detail the created presentation guide and questionnaire.

6.1.1 Guide

To introduce the conceptual model for gender-inclusive requirements to the participants, we created a presentation guide [1]. Reading the guide was required before completing the questionnaire. It contains a total of 20 pages with explanatory text in English and images of the conceptual model and its components. To provide participants with the context and relevance of the work, the guide includes the motivation for the development of the model, the main gender issues that we intend to tackle with it, and the goals of this work. Then, we provide an explanation for each component as well as the definitions for all the concepts of that component. Finally, we present an explanation for the overall model. The guide is shown in Appendix C.

6.1.2 Questionnaire

The purpose of the questionnaire was to evaluate the conceptual model's usefulness and difficulty of understanding, its overall positive and negative points as well as assessing if the participants would use or recommend it. In addition, we intended to evaluate the completeness of each of its four components. The questionnaire [2] included quantitative, multiple choice, and qualitative, open field, questions. It was grouped in nine sections: the introduction, a pre-survey to collect information on the participants' profiles and opinions on gender inclusion in software development, an overview of the conceptual model, four sections for each component of the model (Gender, Sociocultural Context, Human Actor, Software System), the overall opinion of the conceptual model, and finally, the guide evaluation. The information and questions of the questionnaire were written in English, however the participants were free to answer in English or Portuguese. A PDF version of the questionnaire is shown in Appendix B.

Next, we detail each section of the questionnaire.

Introduction

This section introduces the context and the objectives of the evaluation questionnaire. It provides the link to the *Google Docs* folder where the presentation guide is available and informs of its reading necessity to answer the questions that follow. It also informs the participant that the answers are anonymous. Finally, it presents the maximum time it will take to complete the questionnaire.

[1] The presentation guide is available in the following link: https://drive.google.com/file/d/1_GOcBM-ronXlTxduFLaa8FTIaSvwLE2P/view

[2] The questionnaire form is available in the following link: https://forms.gle/FZjWVuwttJmrmzUp9

Pre-survey

This section collected personal information of the participant regarding their age, gender, level of academic training, scientific domain of degree, current occupation, and experience with conceptual modeling. Additionally, it included the following questions to assess the participants' opinions regarding gender inclusion in software development:

- A question about the importance of gender inclusion in software development, with 5 options ranging from 1 to 5, where 1 is not important and 5 is very important;
- A multiple choice question that asks which best describes their perceptions of gender, where the first option reads "Persistent structural inequalities with statistical evidence.", the second "Inequalities are mainly due to culture and education.", the third "There are no barriers because there is female presence and inequalities are circumstantial.", and finally "I don't have an opinion.";
- A validation boxes question that asks in which situations is gender inclusion relevant in their professional activities; This included six options and one open field option for participants to add situations more specific to their activities that were not mentioned by the given options.
- An open field question that asks what approaches for addressing gender issues in software development do they know.

These four questions were formulated with the aim of collecting the participants views, concerns, and knowledge on gender inclusion in software development. The first question helps us understand the relation between the participants' opinion on how important gender inclusion in software development is with their opinion about the importance and utility of the model. This enables us to gather a more complete set of information and make a richer analysis of the participants' opinions. The second question collects the participants' views on the concept of gender and the third collects information of how they introduce gender in their daily professional activities. Finally, the last question aims to understand the participants' current knowledge on the existent approaches for addressing gender issues that they could possibly be using as reference for comparison.

Overview of Model

This section presents an image of the conceptual model for gender-inclusive requirements with its four components highlighted and a brief description of its main objective. Then, it informs the participant that the following sections present questions for each component individually.

Components

In this evaluation, we aimed to collect the participants opinions on the completeness of each component of the conceptual model. Thus, we created four different but similar sections for each one. Each section has one multiple choice question of "Yes" or "No" that asks if the component expresses what it intends to express. Bellow we present the question for each component.

- The first section presented the Gender Component of the model. It asks if the component *captures the concept of gender as perceived by individuals (self-identity) and society*.
- The second presents the Sociocultural Context component and asks if the concepts are *enough for capturing the relation of gender to the sociocultural contexts of software use and development*.
- The third, if the concepts of the Human Actor component are *enough for capturing a person with a unique identity of individual and social characteristics influenced by their gender*.
- Finally, the Software System component section asks if this component *is enough for capturing a software system as the product of human-based decisions in a sociocultural context influenced by gender*.

In addition, we included one open field question that requests the participant to further explain their choice if they selected "No" in the previous question. This allowed us to collect more valuable perspectives regarding the completeness of each component.

Overall opinion about the model

After the evaluation of each component's completeness, we aimed to assess the usefulness and difficulty of understanding the complete conceptual model. Furthermore, we aimed to collect the participants' opinions on the positive and negative aspects of the model and if they would use and/or recommend it.

Therefore, we created a section that presents the complete conceptual model to the participants and asks their overall opinion about it. It has two questions, one about the usefulness of the model and the other about the difficulty to understand it. Both provide a scale of options, from one to five, where one is 'Not useful' and 'Not difficult' and five is 'Very useful' and 'Very difficult' respectively. Then, it asks participants if they would use the model and if they would recommend it. Both are two multiple choice questions with three options: 'Yes', 'No', and 'Perhaps'. In addition, it asks to whom could the model be useful and provides a multiple boxes question with the possibility of open field. Finally, it asks two open field questions for participants to share their opinions on the conceptual model. The first requests the strongest points of the model and the second requests the weakest.

Guide evaluation

This final section asks participants two multiple choice questions, with a 'Yes' or 'No' options for each. The first asks if the provided guide was useful. The second, if it was easy to understand. This section of the questionnaire aimed to assess the quality of the guide. Also, if the participant understood the model, thus providing answers to the questionnaire that reflected such understanding.

6.2 Participants selection

To evaluate the conceptual model, we considered two groups of participants: experts and academics or industry practitioners. The first group included bibliography authors that have expertise in diverse areas of computer science and have addressed gender in their work and experts in conceptual modeling that could have or not addressed gender. We considered the list of authors from the main studies of the systematic mapping study (see Chapter 3). These participants can provide us an experienced, well-informed evaluation of the model in terms of its adequacy and utility. On the other hand, the second group included academic staff and students as well as industry practitioners from diverse areas. We included this group of participants in our evaluation to understand how the conceptual model was perceived by individuals of different backgrounds, such as academic training and degrees. Although gender issues are well documented across various social and individual dimensions of society, these topics are still typically considered sensitive to address in practice, particular in the computer science field. Thus, it was important to assess both academic and industry perceptions on the introduction of a gender perspective in software engineering through conceptual modelling.

We sent an email to each person with an introductory formal text, a brief justification for why the person was receiving this email and the invitation to participate in the evaluation followed by the link to the form. Furthermore, this link was not available to participants who were not invited to avoid unintended responses. The responses to the form were collected within a period of fifteen days. In a total of 64 people who were contacted through the invitation email, 31 submitted their responses.

6.3 Results and discussion

The results from the questionnaire form were extracted from the *Google Forms* application to a *Microsoft Excel* spreadsheet. Then, we performed a data cleansing to ensure the information was consistent and homogeneous, particularly in the open field data. This process guaranteed the information was correct and ready to be treated. In the spreadsheet, each column mapped to a question and the rows to the corresponding answer of each participant.

To carry out a more complete analysis of the data, we created a Python file and used *Plotly*[3], an open source Python graphing library for creating interactive, high-quality graphs. As *Plotly* does not come built in with Python, we installed the necessary packages to use its three main modules that allow the creation of specific graphs from imported datasets. We imported the spreadsheet and implemented bar charts for the Likert scale questions and pie charts for the multiple choice questions.

The discussion of the results follows a top-down approach. First, we present and discuss the general results for each question. Then, we detail the discussion by including qualitative responses from the participants that complement and augment the quantitative analysis. For each question regarding the components of the model, we discuss the qualitative responses according to years of experience. Therefore, in the following subsections, we present and discuss both forms of data jointly as to construct a more substantial foundation for outlining conclusions.

6.3.1 Participants demographic

These questions provided information about the characteristics of our respondents. As we will discuss in this subsection, our aim of demographic diversity was achieved. Particularly in the participants' gender, current occupation, and academic training.

In terms of gender, the 31 participants were diverse, with fifteen women (48.4%), fifteen men (48.4%), and one non-binary (3.23%), as shown in Fig.6.1

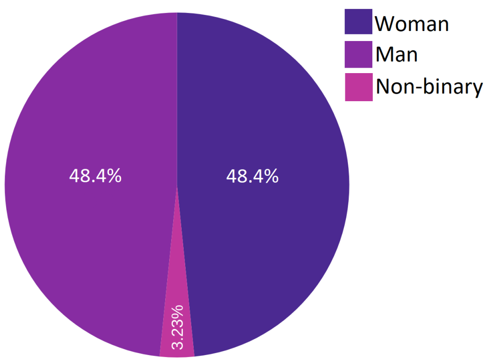


Figure 6.1: Percentage of participants by gender

Regarding the scientific domain of the participants' degree, 22 participants (71%) answered Computer Science, as shown in Fig.6.2. The remaining scientific domains of the participants' degrees (29%) were very diverse, and included both technical and non-technical domains.

[3] https://plotly.com/python/

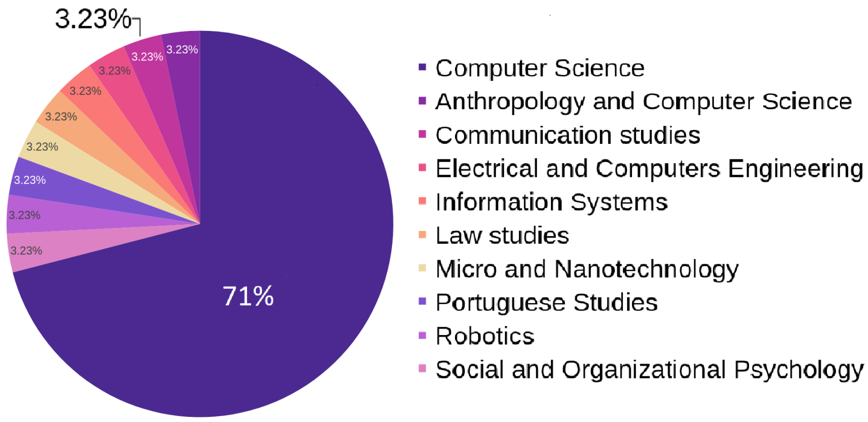
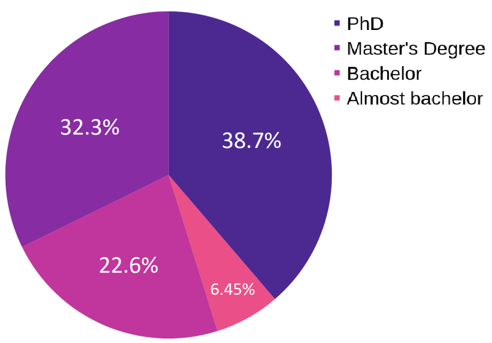


Figure 6.2: Percentage by degree

Thus, we divided our analysis into three groups, a group of participants whose degree is Computer Science that represents 71%, a group whose degree is not Computer Science but it is related to science or engineering, representing 12.9% of the participants (Electrical and Computers Engineering, Information Systems, Micro and Nanotechnology, and Robotics), and a group that encompasses the degrees related to Languages and Humanities, representing 16.1% (Anthropology, Communication Studies, Law Studies, Portuguese Studies, and Social and Organizational Psychology). We use these three groups for the remaining of the discussion.

Academic training was also very diverse amongst our participants, with 6.45% Bachelor students, 22.6% had completed Bachelor, 32.3% a Masters Degree, and 38.7% PhD. The results are shown in Fig.6.3.

Similarly, the participants' current occupations were also diverse and quite balanced, as shown in Fig.6.4. From a total of 31 participants, 35.5% are students, 22.6% academics, 22.6% practitioners, 16.1% researchers, and 3.23% researcher and practitioner.



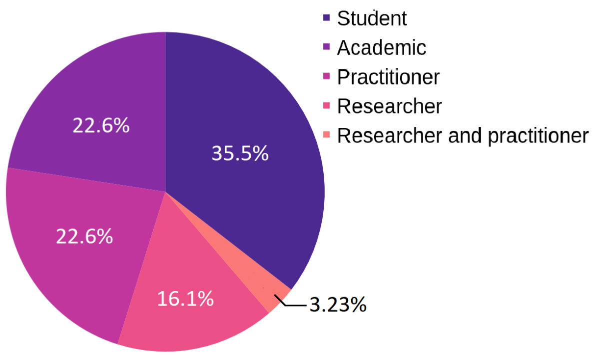


Figure 6.4: Percentage by current occupation

Regarding years of experience in conceptual modeling, we asked participants to select the range suitable for their level. We divided the years of experience into four groups where 0-2 years represent participants with little to no experience, 3-5 years we consider the participant has significant experience, 6-10 years considerable experience, and from 10+ years we consider the participants experts. The results are shown in Fig.6.5.

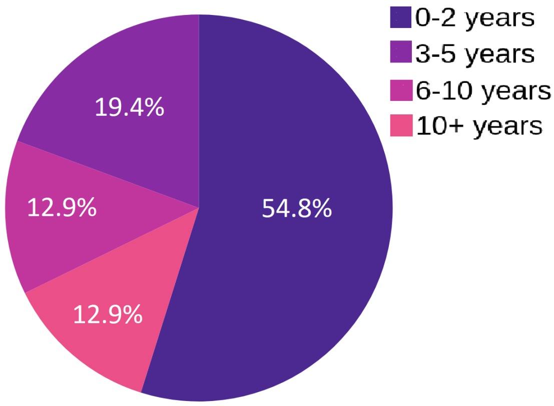


Figure 6.5: Percentage by years of experience in conceptual modeling

A total of 54.8% participants had less than two years of experience in conceptual modeling. However, from the 9 participants who do not have a Computer Science degree, 7 (77.8%) selected this option. Thus, they represent 22.6% of the 54.8%. Furthermore, 19.4% of the participants had three to five years of experience and 12.9% had six to ten years. Finally, we had a total of 12.9% of participants with more than ten years of experience.

Perceptions of gender issues

The aim of this analysis was to understand the participants' opinions of the importance of gender inclusion in software development and their perceptions on gender issues in software development. Gathering the participants' perspectives on gender inclusion is relatively significant to understand the extent to which the participant cares about this topic, but does not find the model valuable, or does not consider it a relevant matter and thus neither the model.

Therefore, we asked participants to rate their opinion on the importance of gender inclusion in software development in a scale where "1" represents "Not Important" and "5" represents "Very Important". As shown Fig.6.6, the results indicate that eighteen (58%) of our participants consider this topic "Very Important", followed by nine (12.9%) participants who consider it important. Two (6.45%) participants consider gender inclusion moderately important and other two (6.45%) do not consider it important. Finally, no participant chose option "2", slightly important.

Nevertheless, the results demonstrate that the majority of participants consider gender inclusion in software development very important. Thus, it suggests the topic the model addresses is timely and relevant.

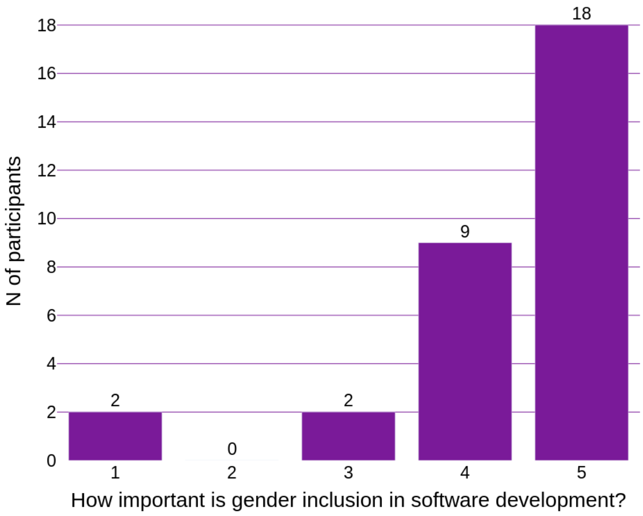


Figure 6.6: Results for the participants' opinion of the importance of gender inclusion in software development

Similar to the previous question, we sought to understand the extent to which the participant is unaware of existing gender issues in software development, and therefore, does not consider that solutions or approaches, such as the model, are needed. However, only one (3.23%) participant responded that inequalities are circumstantial as there is already female presence in the field. In addition, one participant (3.23%) did not have an opinion on the topic. Indeed, the results indicate that the majority of participants are aware of current gender issues in software development, as shown in Fig.6.7.

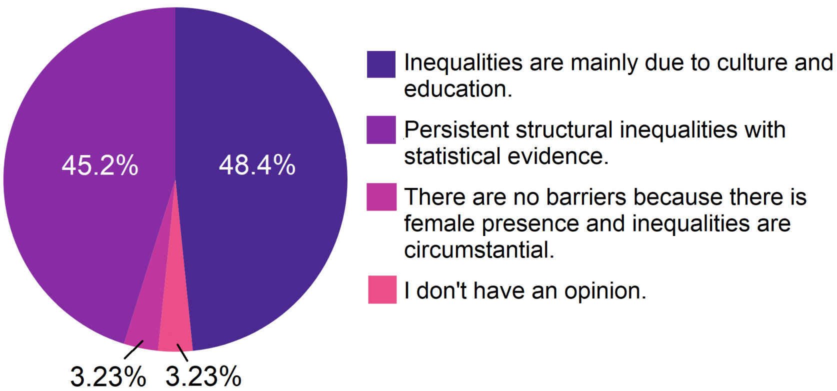


Figure 6.7: Results for the participants' perceptions of gender issues in software development

Fifteen (48.4%) participants consider that gender issues are mainly due to culture and education while fourteen (45.2%) are aware of the statistical evidence that demonstrates persistent structural inequalities. Furthermore, we included these two distinct perspectives because the conceptual foundation for the model involves evidence of structural inequalities possibly unfamiliar to participants who relate inequality to culture and education. Thus, we sought to understand if these participants would still consider the model comprehensible.

Knowledge of other approaches

When asked about their knowledge of approaches for addressing gender issues in software development, the majority of the participants left the open field empty or specifically wrote that they do not know any approach. Within the group of participants who responded, the GenderMag method was mentioned by six (19.35%) participants. Furthermore, some participants mentioned that they did not know any formalized approaches but were aware of recent research on the topic of gender bias and gender inclusion:

- "I cannot point to anything specific, I am more aware of those investigating it and reporting on it (e.g. Joy Buolamwini) than practical tools for implementation".

In addition, some participants commented that, although they did not know any specific approaches to gender inclusion in software development, they had knowledge of gender inclusion practices in general:

- "Formally none. But informally, group activities, equal payment and equality in general" and "We have been developing a project which we considered diversity and gender as relevant criteria during the selection of the members."

In general, participants were interested in and familiar with the topic of gender issues in software development but were not aware of formal approaches or tools for addressing them, besides the GenderMag method.

6.3.2 The components

This subsection presents the results for the components' evaluation, where each of them was dedicated to collect feedback concerning the concepts of the respective component. The overall results tend to demonstrate that the concepts and relationships in the components are enough to represent what they intend.

Gender component

The results for the gender component were rather positive, as shown in Fig. 6.8. Irrespective of academic level or years of experience in conceptual modeling, 93.5% participants answered that the concepts were enough to capture gender both as a self-held identity and as a social construct. Only two participants, with less than two years of experience in conceptual modeling, answered "No".

However, the two participants (6.45%) were bibliographical authors with backgrounds in Anthropology and Computer Science and Human-Computer Interaction. Thus, they have significant knowledge and experience regarding the concepts of this component and provided valuable feedback on how to improve it:

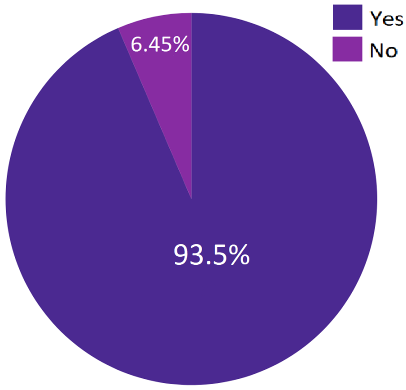


Figure 6.8: Results for the gender component evaluation

- "There may be differences in perceived gender, gender expression, and gender roles. Biological sex is more complicated than male, female, or intersex (drawing on the Guide). Gender roles are not necessarily based on Biological Sex (unless I'm reading the diagram wrong). "Men" and "women" may not be the only categories of these roles if we look beyond a Western context."

These suggestions open the way for further refinement of the concepts and the inclusion of additional ones related to a potential dimension of culture in the model.

Sociocultural context component

The results for the sociocultural context component were also quite positive with 87.1% participants considering that the component was enough to represent the relation of gender to the sociocultural contexts of software use and development. These are shown in Fig.6.9.

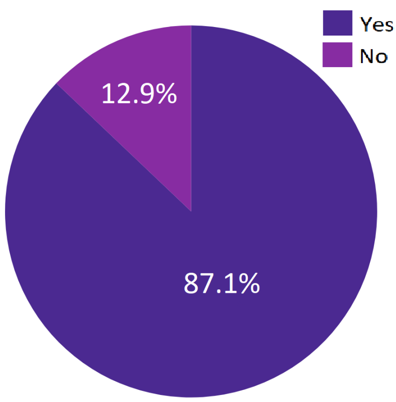


Figure 6.9: Results for the sociocultural context component evaluation

The four (12.9%) participants who answered negatively provided similar motivations for their responses. In general, they raised questions regarding the oversimplification of this component's concepts that, from their point of view, was not adequate given their relevance in the model:

- "I think this consideration is good. The thing to consider here is how turning "Sociocultural Context" in this model form reduces the extensive complexity and nuance of context of development and use."

Three (9.7%) of the four participants had less than 2 years of experience in conceptual modelling and one had more than ten years of experience. The latter provided the following suggestion:

- "I don't really think it's binary; I even think that the "Sociocultural Context" class shouldn't be abstract. Have you ever thought about the possibility of having a third class being a specialization of the two (Interaction Context and SocioTechnical Context) with multiple inheritance. In other words, I think it's not that simple."

The provided feedback raised important issues and questions that should be further explored for improving this component. During conceptualization, we aimed for a purposeful simplification of this component in the context of the model domain, while ensuring it did not disregard the complexity of the concepts. The objective was to provide model users with the fundamental concepts that would sufficiently represent the impact of gender in the sociocultural contexts of development and use. However, the concept of sociocultural context is indeed complex and entails many aspects that could be investigated in the future to fully represent what it aims.

Human Actor component

The results for the human actor component are presented in Fig.6.10. A total of six (19.4%) participants answered that the concepts and relations of this component were not enough to capture what it intended to. Four (13%) of these participants had less than two years of experience in conceptual modeling, one (6.4%) had six to ten years, and one (6.4%) had more than ten years of experience.

Although the human actor component had an overall positive evaluation, it was the component that had more negative responses among the four components. The participants who answered "No" provided different reasons for not agreeing that the human actor component did not sufficiently captured people's unique identities influenced by their gender. Mostly, participants mentioned concepts they would add to those the model already contemplates. The following two quotes are from the two participants who had more experience in conceptual modeling.

- "I was thinking about culture and background, as I'm not sure if those concepts are really included in the perceptions and beliefs. Typically, perceptions and beliefs are also shaped by culture (in terms of society) and by education (not formal education, but the education we receive at home, from parents or guardians)"

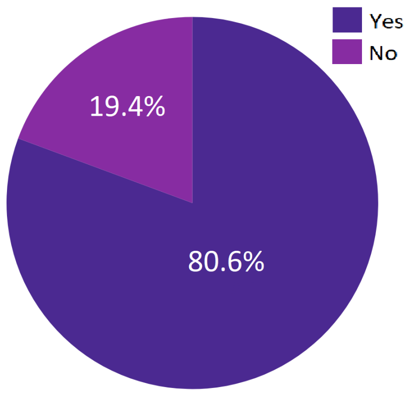


Figure 6.10: Results for the human actor component evaluation

- "I think the environment should be specialized in Family Environment, Workplace, and Social environment."

The information regarding the mentioned concepts was not found in the selected studies of the systematic mapping study, and thus, we did not have evidence that supported their inclusion in the model. However, this feedback constitutes initial evidence that these concepts should be integrated into the conceptual model, if confirmed and validated by more experts in future evaluations.

Software System component

Figure 6.11 shows the results for the software system component evaluation. The evaluation was quite positive, with 93.5% of the participants considering that component was enough to capture a software system as the product of human-based decisions in a sociocultural context influenced by gender. From the two (6.45%) participants that gave a negative response, one (3.225%) had more than ten years of experience in conceptual modeling and one (3.225%) had less than two years.

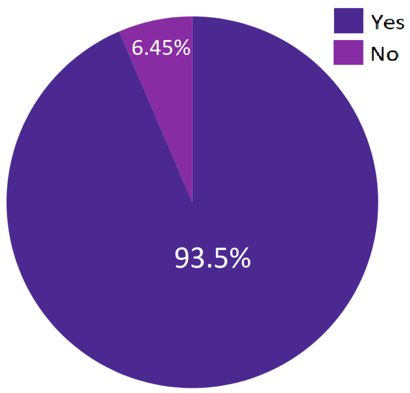


Figure 6.11: Results for the software system component evaluation

The qualitative data provided for this component was especially useful to understand the participants' negative responses. They considered that, due to the simplification of complex concepts, the purpose of the model should be framed differently. One of the participants provided the following feedback:

- "It's just really a fundamental issue with me in how the question is asked and what the model should achieve. Maybe it makes more sense to frame it as an invitation to look at different aspects for reflection instead of describing individual experiences."

We agree that one of goals of the model is to encourage reflection in requirements elicitation. However, as a conceptual model, it also aims to describe and represent the concepts we identified in the systematic mapping study. Thus, we constructed the question in such a way that would allow us to assess the adequacy of the concepts and relationships of each component.

6.3.3 The conceptual model

After evaluating the model's components, we asked participants about their overall opinion about the conceptual model. First, we asked participants to rate the usefulness of the model from 1 to 5 where one is "Not useful" and five is "Very useful". Second, we asked participants how difficult it is to understand the model with the same format as the previous question, but 1 represents "Not Difficult" and 5 is "Very Difficult".

The questions assess two essential aspects of the model. As a conceptual model, it must be sufficiently clear and precise that its users are comfortable in applying it as a means for discussing gender issues and inclusion in requirements elicitation. At the same time, it must be comprehensive enough to cover all the relevant concepts to be reliable and useful in the task it aims to achieve. The results were positive for both usefulness and difficulty in understanding from participants with fewer years of experience to experts.

Then, we asked participants whether they would use the model and whether they would recommend it. The results from this analysis were also positive, particularly among more experienced participants.

Finally, the quantitative results are explained in more detail through the qualitative data that was collected through two open-ended questions. These invited participants to provide their opinions on the strongest and weakest points of the model. Most feedback was positive and provided valuable outlooks for further improvement of the model.

Usefulness

In Fig.6.12, we can see that the responses clearly concentrate on the "very useful" and "useful" options, thus suggesting the model conveys the usefulness it intended to, both for participants with more and less experience in conceptual modeling.

Indeed, fourteen (45.2%) participants responded that the model was "very useful" and twelve (38.7%) responded with "useful", that is, a total of 83.9% positive answers.

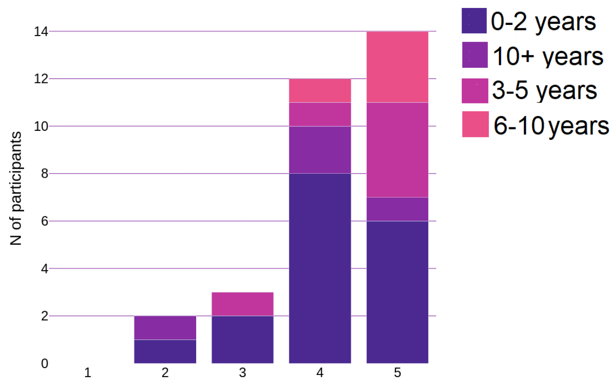


Figure 6.12: Results for the usefulness of the model by years of experience in conceptual modeling

Three (9.7%) participants considered it "Moderately useful" and two (6.4%) "Slightly useful". Yet, as shown in Fig.6.12, four out of the five participants that gave a 2 or 3 rating evaluation have less experience in conceptual modeling and possibly, were unsure or did not come to a conclusion about the usefulness of the model.

Furthermore, four of the more experienced participants (three with 6-10 years and one with 10+ years of experience) considered the model very useful, while three (one with 6-10 years and two with 10+ years of experience) considered it useful. With the exception of one, all participants with higher levels of experience in conceptual modeling (22.6% in 25.8%) considered it to be useful or very useful. Moreover, no participant considered the model "Not useful". We perceive very positive results from the analysis of the usefulness of the model. The feedback suggests that both less experienced and expert participants found the model mostly useful, thus demonstrating the utility of this work for different groups of participants.

In addition to the years of experience, we sought to understand if participants who did not consider gender inclusion an important topic did not recognize the model's usefulness. Therefore, we crossed the data on perceived importance of gender inclusion in software development with the opinion on the usefulness of the model. The results are shown in Fig.6.13, where 1 represents the answer "Not important", 3 "Moderately important", 4 "Important", and 5 "Very important" No participant selected option 2 "Slightly important" and therefore it is not represented in the figure (see Subsection 6.3.1).

This analysis yielded interesting results. One of the participants who considered that gender inclusion is "not important" (option 1) likewise did not perceive utility in the model. The other evaluated gender inclusion as a "very important" (option 5) topic but did not consider the model useful. Hence, it indeed suggests the importance of understanding the participants' viewpoints on gender inclusion when assessing the usefulness of the model to separate those that do not consider the topic relevant in any circumstance or are critical of the model itself.

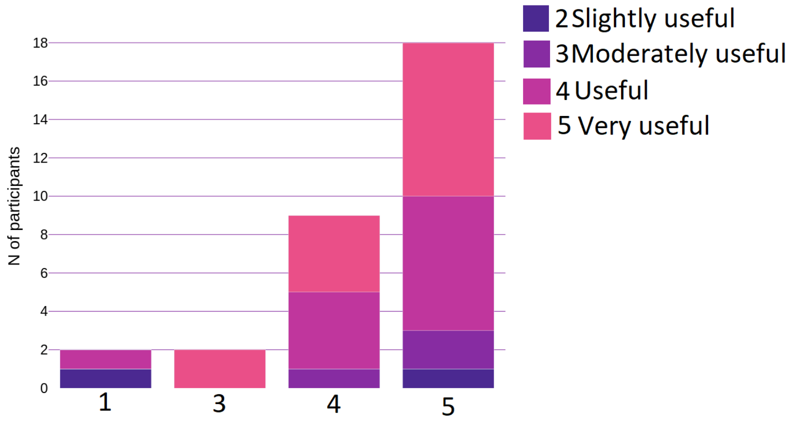


Figure 6.13: Results for the usefulness of the model by perceived importance of gender inclusion

Furthermore, the participants who considered gender inclusion in software development a "moderately important" topic all evaluated the model as very useful. This may indicate that these participants do not consider it a priority or urgent topic, but still recognize its importance, and therefore, were able to contemplate the model's utility. In addition, we had one participant that viewed the model as "useful", despite selecting option 1 for the importance of gender inclusion. Possibly, this participant might not be interested in the subject but acknowledged the potential utility of the model.

Finally, among the participants who considered the topic to be "important" or "very important," there is a visible tendency of positive perspectives on the usefulness of the model. We consider this observation valuable as it also suggests that the participants who regard this topic as important held positive viewpoints of the model's usefulness.

Following this question, we asked participants how difficult it is to understand the model.

Difficulty of understanding

The results for the difficulty of understanding indicate that participants considered it slightly to moderately difficult to understand, as shown in Fig.6.14.

From the participants who did not considered the model difficult to understand, two have more than ten years of experience in conceptual modeling and one has six to ten years of experience. These results are consistent with these participants levels of experience. Furthermore, two participants with less than two years of experience also answered "not difficult" to understand. Eight (25.8%) participants considered "slightly difficult" to understand the model and ten (32.26%) participants considered "moderately difficult", a total of 18 participants. Thus, the majority of our participants considered it slightly to moderately difficult to understand. This seems plausible given the participants varied levels of experience and academic degrees.

Moreover, seven participants considered the model "difficult" to understand. From the

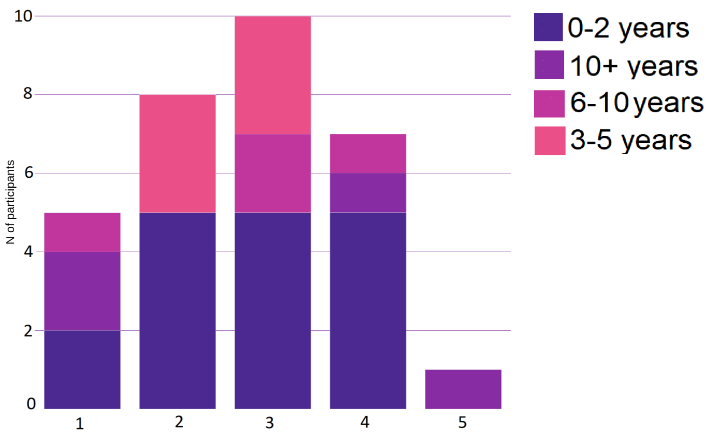


Figure 6.14: Results for the difficulty of understanding the model by years of experience in conceptual modeling

seven, one had more than tens years of experience in conceptual modeling and the other six to ten years. The remaining five participants had less than two years of experience. Indeed, participants with less experience in conceptual modeling had a range of perceived difficulty, from option 1 to 4. This evidence can be due to the fact that a participant can have little experience in conceptual modeling but have significant experience with gender-related concepts, methods, or studies, and therefore can understand the model with less difficulty.

Hence, to complement this discussion, we analyzed whether participants who did not have the knowledge on gender issues that constitutes the conceptual foundation of the model could still understand it. The results are shown in Fig.6.15, where 1 represents the answer "I don't have an opinion", 2 "Inequalities are mainly due to culture and education", 3 "Persistent structural inequalities with statistical evidence", and 4 "There are no barriers because there is already female presence and inequalities are circumstantial" (see Subsection 6.3.1). This analysis produced rather interesting and insightful results.

First, the participant that mentioned not having an opinion on gender issues considered that the model was not difficult to understand, which may suggest this participant did not take enough time to fully comprehend it. Furthermore, the option that corresponds to the model's conceptual foundation is "Persistent structural inequalities with statistical evidence" (option 3). Indeed, all participants who answered "not difficult", with the exception of the one mentioned above, chose this option. This analysis indicates that participants with a more thorough knowledge of gender issues in software development understood the model with less difficulty. This group's perceptions of difficulty ranged from "not difficult" to "difficult".

Participants who chose the second option demonstrated knowledge on gender issues but not on all the concepts the model encompasses. This is visible in Fig. 6.15 as this group of participants' perception of difficulty ranged from "slightly difficult" to "difficult", with the majority considering it "moderately difficult".

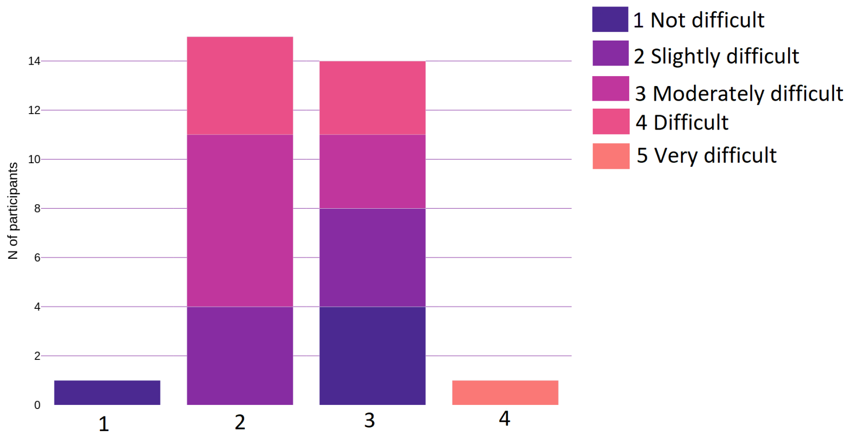


Figure 6.15: Results for the difficulty of understanding the model by gender issues perceptions

Finally, the participant who answered option 4, which corresponds to the perception that there are no gender issues and that these have been overcome, considered the model very difficult to understand. This is reasonable as the domain of the model covers specific knowledge related to gender issues and gender inclusion the participant may not be aware of, thus making it more difficult to grasp at first.

In conclusion, we perceive that our aim of balancing the model's level of precision and specification to accommodate the complexity of reality that is both useful and understandable was achieved.

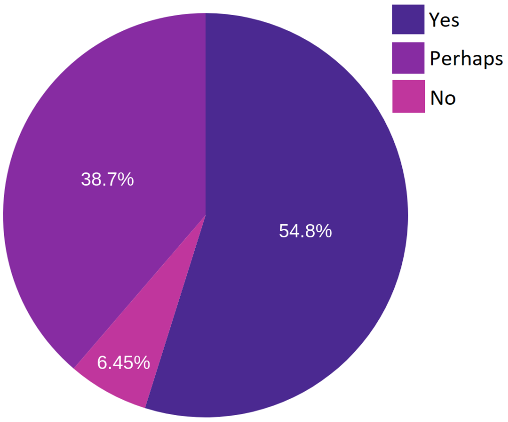
In the following subsections, we present and discuss the responses of about whether the participant would use and recommend the model.

Use and recommendation

The results for whether the participant would use the model were positive, with 54.8% participants answering "Yes" and 38.7% answering "Perhaps", as shown in Fig.6.16. Moreover, the question of whether the participant would recommend the model yielded more positive results, with 80.6% participants answering "Yes" and 16.1% answering "Perhaps", shown in Fig.6.17.

Considering the participants' demographics, we perceive 54.8% "Yes" responses to whether they would use the model as a positive result. Indeed, when asked whether they would recommend the model, 80.6% participants answered "Yes". Moreover, 38.7% participants answered "Perhaps". This suggests that, although they perceived the usefulness of the model, they may have had difficulty understanding how to integrate it into practice or how it would compare to established approaches.

To settle this analysis, we filtered the results to these questions to only include the responses of the most experienced participants (more than six years of experience in conceptual modeling). The results are shown in Fig.6.18



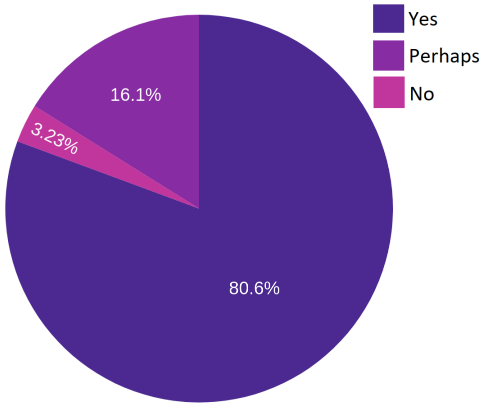


Figure 6.17: Percentage of participants who would recommend the model

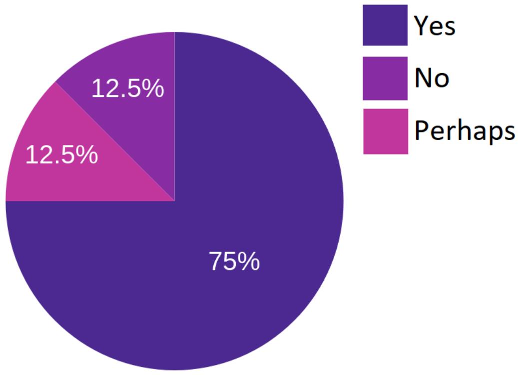


Figure 6.18: Percentage of expert participants who would use the model

A total of 75% of the participants with higher levels of experience in conceptual modeling responded "Yes" to whether they would use the model. In line with the previous observation, the percentage of "Yes" responses increased. These participants have a comprehensive and extensive knowledge of conceptual modeling and therefore can provide a more grounded perspective on the use and application of the model.

In summary, we consider the qualitative results positive regarding the evaluations of both participants with less experience and experts.

Finally, the form had two open field questions where participants could optionally provide their opinions on the strongest and weakest points of the model. The questions were introduced to further understand the individual opinions of the participants. In addition, they were intended to complement the quantitative data presented by understanding their overall reasoning and motivations for their close-ended responses.

Strongest points

Nearly all participants provided their perspectives when asked about the strongest points of the model. Thus, suggesting a likely interest and further analysis of the model by the participants. The feedback was positive and quite comprehensive as it provided important and diverse viewpoints on the value of the model. For instance, participants mentioned why and in what contexts the model could be valuable for them. Moreover, they complimented the organization of the model and the adequacy and completeness of the concepts.

- "I could see the use of this model as a translation tool to highlight the role of gender in software development and various aspects of gender to consider. I applaud the effort that clearly went into the model. I think the really key point is that context of development and context of use can be different."
- "The conceptual model is very detailed and accurately reflects the different elements related to gender inclusion."
- "It provides a seemingly up to date summarized deconstruction of the concept of gender and its impact on software development, presented in a technical language easily understandable by the target audience."
- "I think this model conceptualizes the key aspects of gender inclusiveness for software development very well. The compiled list of the Human Actor characteristics was a great resource to understand many different aspects to consider when striving to develop inclusive software."

Furthermore, participants with less experience in conceptual modeling (0-2 years) provided positive feedback regarding the ease of understanding and accessible visual illustration of the model.

- "The conceptual model is very concise, and shows a high level of completion and detail regarding gender inclusion. It is also accessible and easy to understand."

This qualitative feedback complements and further explains the quantitative results that showed less experienced participants considered the model slightly to moderately difficult to understand in average.

Overall, this feedback is very encouraging and provides valuable information of how diverse participants perceived the model.

Weakest points

Regarding the model's weakest points, the answers were varied. However, they could be grouped into two types: The first group focused mostly on the need to apply the model to a practical example as a proof of concept or to integrate it into established practices of software development so it would be adopted by software practitioners.

- "I would like to see a practical application of the model in an example case. It's hard for me to understand the impact of using the model without it. Although, as it stands, it's already an interesting point of discussion and a way to raise awareness for related issues."
- "It would be great to see a practical example of how the model can be used to formulate gender inclusive requirements. As a future step, a defined method (like GenderMag) for gender requirements formulation could be developed to ease adoption among software development teams across the world."
- "Some more framing of the model could be useful - how should users of the model approach it. As a heuristic for unpacking aspects of gender to consider - rather than capturing the whole truth/reality of gender - seems like it would be a valuable way to introduce the tool and how it could be of use."

The comments point to an overall recognition of the conceptual model's potential for the development of inclusive software if it is further incorporated into current development practices, and thus, suggesting the value of the framework proposed in this dissertation (see Chapter 5).

The second group mentioned that the model could be too complex to be fully understood. These concerns were mentioned primarily by the participants who did not have Computer Science as their degree's scientific domain.

- "There is a lot of data to collect and model! I am not sure how easy it would be to solicit all of that information from users in a practical application. Similarly I am not sure how it would be applied in practice (but I am not a software developer)."
- "It could have an additional modeling for non technical professionals."

Indeed, the main goal of the model was to provide a representation of gender in the context of software development and provide requirements engineers the knowledge to take a gender perspective in requirements elicitation. Yet, since the elicitation process can involve stakeholders who do not have a technical background, it would be positive to have a more accessible conceptualization of the model so they could maintain a more informed and active collaboration with requirements engineers. Nevertheless, their feedback was overall positive and demonstrated their interest in this work.

6.3.4 Guide Evaluation

We included two questions to evaluate the guide provided for completing the form. The results for the usefulness of the guide are shown in Fig.6.19 and for the ease of understanding are shown in Fig.6.20.

The results were overall unanimously positive regarding both usefulness, with 100% answering yes, and ease of understanding of the guide, with 90.3% yes and 9.7% no, from which we can assume that the participants understood the model.

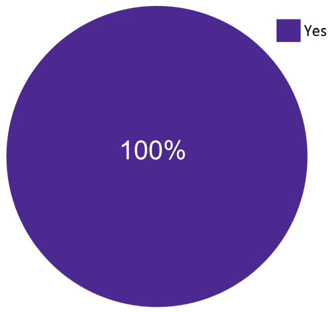
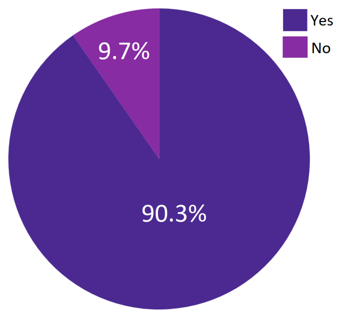


Figure 6.20: Results for the ease of use of the guide



6.4 Threats to validity

After presenting and discussing the evaluation of our proposed conceptual model, we discuss the threats to its validity. We followed the proposed guidelines in [189] to identify them. They are divided in internal, external, construct, and conclusion validity.

6.4.1 Internal validity

We tried to make our guide as succinct and accessible as possible. However, it still may feel overwhelming for a participant who does not have enough knowledge on gender concepts and conceptual modeling to comprehend. We mitigated this threat by asking participants a set of questions related to gender inclusion before answering the questions regarding the model to assess their knowledge on the topic. Although we did not asked participants directly about their experience with gender-based approaches for software development, we asked their experience on conceptual modeling. Nevertheless, we had no evidence of this effect on participants as they considered the guide useful and easy to read. Moreover, we did not inform participants in the illustrative guide that the conceptual model was based on a systematic mapping study. This would have been a valuable information to contextualize the reasoning for the concepts, and thus, provide participants the opportunity to give a more informed answer. Therefore, it constitutes a possible threat to the validity of the results. Furthermore, the questionnaire was divided in nine sections, which could have exhausted the participant and affected the quality of their responses. However, we tried to maintain a low number of questions in each section. The questions themselves were almost all multiple choice and open field questions were optional. However, the majority of participants provided answers in optional and open-ended questions, suggesting this threat may have been mitigated.

6.4.2 External validity

Although the number of participants who answered the questionnaire was considerable given the time provided to participate in the evaluation, it is still a risk to generalize the results from such sample size. Therefore, the number of participants can raise concerns about the validity of the evaluation results.

Despite the limited number of participants, our sample had a relatively high representativeness, as some participants were bibliographic authors with experience in gender concepts, some were experts in conceptual modeling, and some both. However, some participants may not have the knowledge to provide a comprehensive analysis of the model. For instance, a participant with knowledge on gender issues may not be able to understand the model as it was constructed in a technical language. Additionally, participants with more experience in conceptual modeling may not be aware of gender issues or gender-related concepts. Nonetheless, these participants can represent potential stakeholders in the requirements elicitation process, and therefore their perspectives are also important.

To mitigate this threat, we created a guide that introduced participants to the model and provided a description of each of its concepts. In addition, the guide provided context of the existent gender issues that served as motivation for this work. As the participants considered the guide both useful and easy to understand, we are confident that this threat has not compromised the results. Nevertheless, we asked participants about their perspective on gender issues and gender inclusion to assess whether the usefulness/difficulty in understanding the model response was because the participant's perspectives on gender concepts or because they found the model itself not useful/difficult to understand. Yet, we did not ask participants directly about their experience with gender-based approaches in software engineering, which poses a threat to the generalization of these results.

Finally, both the guide and the form were in English so we could gather the opinions of international bibliographic authors. However, some of the participants to which we send an invitation and answered the form did not have English as mother-tongue, which could have impacted their understanding of the gender concepts presented in the guide and the questions in the form. Nevertheless, the participants provided quite comprehensive English responses to the open-ended questions and considered the guide both useful and easy to understand. Thus, we found no evidence of negative impact of this decision.

6.4.3 Construct validity

The questions of the form pose a threat to the validity to this study due to their construction. Particularly, if the questions were constructed in such a way that they represented what they intended to ask and if they were the right questions to assess the goals of this study. Furthermore, we can not ensure the participants interpret the questions in the way we intended to. However, we constructed the questions in an iterative, careful process that attended to the wording and sentence simplicity and objectivity with the aim of removing ambiguity. Additionally, we constructed the form in distinct and structured sections to provide participants the context of which they were responding to.

6.4.4 Conclusion validity

In this evaluation, we concluded a relationship between perspectives on gender issues and difficulty of understanding the conceptual model, and importance of gender inclusion and usefulness of the model. To ensure the reliability of these conclusions, the questionnaire had a relatively high number of questions, with both quantitative and qualitative clear questions about the participants' profiles, opinions and viewpoints. Thus, we carefully collected enough and clear information from the participants to mitigate the possibility of wrongfully concluding that there is or not a relationship. However, the limited number of participants could pose a threat to the inferences made on the collected data and the conclusions of the evaluation.

6.5 Conclusions

This chapter presented the evaluation of the conceptual model of gender-inclusive requirements. First, we outlined the evaluation planning, including the goals of the study and the tools created to support them. We sought to understand the viewpoints of novices, practitioners, and experts on the conceptual model. As the inclusion of a gender perspective in the software engineering field is still a recent endeavour, we also collected their perspectives on this topic to convey a more detailed discussion of the results. To achieve this, we created a guide that presented the conceptual model with a brief context of its development and a questionnaire that collected both quantitative and qualitative data. Then, we described the participants' selection and invitation process. Finally, we presented and discussed the results of the evaluation. The results were very positive for the four components and for the complete model. Although we received some critiques and suggestions, these were constructive, and thus, open the way for prospective research directions. Overall, the feedback received from the participants was very encouraging and provided valuable insights for future improvement.

7

Conclusions and Future Work

In this closing chapter, we present the contributions of the work developed in this dissertation. We reflect on what we proposed ourselves to accomplish and highlight what we consider to be our main achievements. We finish with suggestions that can open the way for future research directions.

7.1 Contributions

Gender equality and inclusion are fundamental to a prosperous and sustainable society where people can live freely, without fear of oppression, discrimination, and violence. Today, more than ever, we have the knowledge and tools to achieve it and ensure a better future for everyone. However, inequality and exclusion persist in various sectors of society. One of them is the ICT field, which is still struggling to represent the diversity of those it serves. The lack of diversity and power imbalance in software development affects the created products and systems, which results in negative consequences for those who interact with them. Given the importance of software technologies in today's society, addressing this problem becomes urgent and integral to ensure that these systems are gender-inclusive and benefit everyone. In this dissertation, we studied gender issues in software engineering, from which resulted in the proposal of a Conceptual Model for Gender-Inclusive Requirements.

The conceptual model aims to support software and requirements engineers in eliciting gender-inclusive requirements by providing a common taxonomy of gender concepts that impact the software development and systems. Its main objective is to facilitate discussion and analysis of such concepts in the elicitation process to include them in the specification of requirements. The knowledge regarding the key gender-related concepts was collected from a systematic mapping on gender issues in software engineering. The results of the study confirmed three main gender issues: i) software is assumed to be exclusively technical and neutral; ii) software development ignores the complex sociocultural implications of gender and builds it as static and binary; iii) the limited existence of approaches for integrating gender as a complex facet of the identity of users during software development. Although we found approaches for addressing gender issues in software, they focused on the design phase of development. In addition, we also found that there is still a gap on how address gender and gender issues in Requirements Engineering. Despite the impact of gender in the software systems, there is a lack of structured knowledge and guidance for the integration of these concerns in requirements. This systematic mapping study contributed to the state of the art on this topic and served as the foundation for the construction of the conceptual model for gender-inclusive requirements.

Aiming to assist software and requirements engineers in the process of eliciting gender-inclusive requirements, we proposed a framework designed to integrate the conceptual model into their established practices, the GIRE (Gender-Inclusive Requirements Elicitation) framework. This framework represents a first step towards using the knowledge of the conceptual model in a useful, reusable, and practical way. As a case study of the framework application, we developed a gender-inclusive prototype tool to support the framework's elicitation planning phase.

Finally, we performed an empirical evaluation of the conceptual model regarding its usefulness and ease of understanding, whose encouraging results and feedback open the way for future research directions. This evaluation included the collection of participants' perspectives on the importance of gender inclusion in software engineering whose results demonstrated the relevance of this work.

We highlight the following achievements:

- A contribution to the state of the art on gender issues in software engineering (Chapter 3).
- A conceptual model for gender-inclusive requirements (Chapter 4). This conceptual model is the first of its kind.
- The GIRE framework, a first version of a framework to integrate the conceptual model into requirements elicitation techniques (Chapter 5).
- A gender-inclusive prototype tool to support the framework's elicitation planning phase (Chapter 5) as a case study for the viability of the framework.
- An empirical evaluation of the conceptual model with 31 novice and expert participants (Chapter 6).
- A paper titled "Conceptual Modeling of Gender-Inclusive Requirements" accepted for publication and presentation at a CORE A conference: the ER 2021 conference, 40th International Conference on Conceptual Modeling.

The intersection of gender with software technologies is well studied in Humanities disciplines, such as Philosophy, Psychology, and Social Sciences, and in the field of Human Computer Interaction. This interdisciplinary research laid the foundation to better understand the impact and importance of gender in the context of software development.

However, this knowledge had yet to integrated in Requirements Engineering. In this context, this work intends to be a contribution to the study of gender issues in the field of Requirements Engineering and, therefore, a step towards the development of gender-inclusive software. At the same time, we hope this work contributes to further analysis and advancement of this topic in the Humanities disciplines.

7.2 Future work

In this dissertation, we studied gender issues in software engineering and how to incorporate this knowledge into software development. This work resulted in the proposal of a Conceptual Model for Gender-Inclusive Requirements. This conceptual model represents an initial contribution for defining gender-inclusive requirements and formulating them in the elicitation process. However, there is still a long road to follow to successfully develop gender-inclusive systems that benefit everyone. For future work, we list the following possible approaches and research opportunities regarding the conceptual model:

- Despite the positive results in the evaluation of the conceptual model, its concepts and relations could be further refined. The model conceptualized gender as a social construct. Although we consider this conceptualization in line with the results of the systematic mapping study, another approach would be to conceptualize it as an intersectional concept. Because gender is never 'just gender', the conceptual model could include other dimensions of identity that intersect with gender itself and impact the characteristics presented in the model, such as race, culture, age, or class. The conceptual model was built with this perspective in mind, as the concept of gender is part of the concept of identity, from which other dimensions can be added to it.

- Furthermore, the sociocultural context component could be further detailed to include more concepts related to the abstract class and the two classes that implement it to allow a more solid representation of the context of development and use of software. This improvement was suggested by experts in conceptual modeling in the performed evaluation.

- In this evaluation, we contacted authors of the selected studies, which included both experts in gender issues and experts in gender issues in software. However, we focused mainly experts in conceptual modeling. Thus, in a future evaluation, it would be valuable to contact participants with experience in specific approaches for addressing gender (e.g., the GenderMag method) to provide their feedback on the conceptual model. These participants would have a basis for comparison between the methods they already know and the conceptual model, and could provide a more informed opinion about the strengths of the model as well as its limitations and opportunities for improvement from a practical and applicable point of view. This evaluation would provide important information as to how well-defined and complete are the concepts and their relations.

- Finally, further expert validation is required to expand the performed evaluation. These should assess the validity of the results obtained as well as extend the findings with additional analysis and viewpoints. These four main points would be the first steps to improve the proposed conceptual model.

- Future developments of the GIRE framework are needed to ensure its robustness and completeness. The proposed framework is a first step towards the integration of the conceptual model in the elicitation process. However, there remains much work to be done. First, we did not describe the documentation of all concepts, which should also be completed for the remaining ones and thus, finishing the first version of the framework. Second, the framework should provide a way to model and analyze gender-inclusive requirements. In addition, it could provide mechanisms for deriving the final gender-inclusive specifications.

- We developed a prototype tool that supported the process of the GIRE framework. This prototype followed the process of the framework itself to ensure it was gender-inclusive. This has the following two implications: first, although the development of the prototype tool served as a case study for the utility of framework, a more comprehensive evaluation would be needed. The framework should also be evaluated by participants with experience in methods and tools for addressing gender and gender issues in software. This would provide with insightful and valuable feedback for improvement. This evaluation should include both the evaluation of the framework process and its results. For instance, an evaluation could involve two different groups eliciting requirements for a case study system where one uses the framework and the other does not, and a final comparison is made. Another example would be to use the framework for eliciting requirements for a given system, and use the GenderMag method to validate the results, namely, if using the framework led to inclusive design decisions. Second, the prototype tool should be fully implemented and also subject to evaluation regarding its gender-inclusiveness, for example, with the GenderMag method.

- It would be important after completing this work to apply the GIRE framework to capture gender-inclusive requirements for real case studies. This evaluation could be performed either similarly to the case studies described above or by integrating the framework in the practices of the organization or community that is participating. This would allow us to validate not only the utility of the conceptual model and framework in the identification of gender-inclusive requirements, but also its adaptability and applicability for real software systems, and the impact for real users.

- Finally, after the work described above, we would like to create a catalogue of gender-inclusive requirements to support their reuse in the elicitation process. This would involve the identification of patterns in requirements based on the expertise collected in the evaluations and case studies of the conceptual model and the GIRE Framework. Then, the knowledge should be structured into a catalogue that provides a better understanding of the gender-inclusive requirements taxonomy and facilitate communication between those involved in development. In the future, the catalogue could be integrated in the process of the GIRE framework, which could itself enable and support the use of the catalogue's knowledge in the elicitation process as the basis for design decisions and implementation of solutions that result in better and inclusive software for everyone. Overall, the main goal of this work would be to lay the foundation for aligning Requirements Engineering with one of the most fundamental human rights: gender equality.

Bibliography

[1] *About Google Scholar*. visited in July 2020. URL: https://scholar.google.com/intl/en/scholar/about.html (cit. on p. 21).

[2] C. Abras, D. Maloney-Krichmar, J. Preece, et al. "User-centered design". In: *Bainbridge, W. Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications* 37.4 (2004), pp. 445–456 (cit. on p. 40).

[3] A. Acien et al. "Measuring the gender and ethnicity bias in deep models for face recognition". In: *Iberoamerican Congress on Pattern Recognition*. Springer. 2018, pp. 584–593 (cit. on pp. 26, 28).

[4] A. Y. Aksekili and C. J. Stettina. "Women in Agile: The Impact of Organizational Support for Women's Advancement on Teamwork Quality and Performance in Agile Software Development Teams". In: *International Conference on Lean and Agile Software Development*. Springer. 2021, pp. 3–23 (cit. on p. 76).

[5] S. Bardzell. "Feminist HCI: taking stock and outlining an agenda for design". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2010, pp. 1301–1310 (cit. on pp. 27, 28, 34, 37, 51, 53).

[6] S. Bardzell and J. Bardzell. "Towards a feminist HCI methodology: social science, feminism, and HCI". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2011, pp. 675–684 (cit. on pp. 27, 28, 34).

[7] C. Bath. "Searching for methodology: Feminist technology design in computer science". In: *Proceedings of GICT 2009* (2009) (cit. on pp. 34, 40).

[8] L. Beckwith and M. Burnett. "Gender: An important factor in end-user programming environments?" In: *2004 IEEE symposium on visual languages-human centric computing*. IEEE. 2004, pp. 107–114 (cit. on pp. 29, 33, 56–58).

[9] L. Beckwith, M. M. Burnett, and S. Wiedenbeck. "Gender HCI issues in end-user software engineering environments". In: *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2007 (cit. on p. 29).

[10] L. Beckwith et al. "Effectiveness of end-user debugging software features: Are there gender issues?" In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2005, pp. 869–878 (cit. on pp. 29, 37, 56, 59, 64).

[11] L. Beckwith et al. "Gender hci: Results to date regarding issues in problem-solving software". In: *AVI 2006 Gender and Interaction: Real and Virtual Women in a Male World Workshop paper*. 2006, pp. 1–4 (cit. on pp. 28, 57, 59, 63).

[12] L. Beckwith et al. "Gender HCI: What about the software?" In: *Computer* 39.11 (2006), pp. 97–101 (cit. on p. 28).

[13] L. Beckwith et al. "On to the real world: Gender and self-efficacy in Excel". In: *IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC 2007)*. IEEE. 2007, pp. 119–126 (cit. on p. 29).

[14] L. Beckwith et al. "Tinkering and gender in end-user programmers' debugging". In: *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 2006, pp. 231–240 (cit. on pp. 29, 33, 64).

[15] N. Bettenburg and A. E. Hassan. "Studying the impact of social structures on software quality". In: *2010 IEEE 18th International Conference on Program Comprehension*. IEEE. 2010, pp. 124–133 (cit. on p. 51).

[16] A. Bradley et al. "Gendered or neutral? Considering the language of HCI". In: *Proceedings of the 41st graphics interface conference*. 2015, pp. 163–170 (cit. on pp. 26, 28, 60).

[17] C. Breger, G. Elyasi, and I. Mariano. "Reducing Gender Bias in Natural Language Processing methods". In: *Meta Research* (2020), p. 17 (cit. on p. 26).

[18] S. Breslin and B. Wadhwa. "Exploring nuanced gender perspectives within the HCI community". In: *Proceedings of the india hci 2014 conference on human computer interaction*. 2014, pp. 45–54 (cit. on pp. 27, 28, 32, 34, 37, 51, 61).

[19] S. Brooke. ""Condescending, Rude, Assholes": Framing gender and hostility on Stack Overflow". In: Association for Computational Linguistics. 2019 (cit. on pp. 27, 29, 34, 49, 51, 60, 67).

[20] S. Buchmüller et al. "Bridging the gender and generation gap by ICT applying a participatory design process". In: *Personal and Ubiquitous Computing* 15.7 (2011), p. 743 (cit. on pp. 40, 65, 66).

[21] J. Buolamwini and T. Gebru. "Gender shades: Intersectional accuracy disparities in commercial gender classification". In: *Conference on fairness, accountability and transparency*. 2018, pp. 77–91 (cit. on pp. 27, 28, 32, 37).

[22] M. Burnett et al. "Finding gender-inclusiveness software issues with GenderMag: a field investigation". In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 2586–2598 (cit. on pp. 27, 28, 57, 61–63).

[23] M. Burnett et al. "Gender differences and programming environments: across programming populations". In: *Proceedings of the 2010 ACM-IEEE international symposium on empirical software engineering and measurement*. 2010, pp. 1–10 (cit. on pp. 29, 33, 37).

[24] M. Burnett et al. "Gender HCI and microsoft: Highlights from a longitudinal study". In: *2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE. 2017, pp. 139–143 (cit. on pp. 27, 28, 32).

[25] M. Burnett et al. "Gender in end-user software engineering". In: *Proceedings of the 4th international workshop on End-user software engineering*. 2008, pp. 21–24 (cit. on pp. 29, 54, 63–65).

[26] M. Burnett et al. "GenderMag: A method for evaluating software's gender inclusiveness". In: *Interacting with Computers* 28.6 (2016), pp. 760–787 (cit. on pp. 3, 27, 28, 30, 32, 33, 37, 38, 54, 56, 57, 59, 61, 63–65).

[27] J. Cassell et al. "Genderizing hci". In: *The Handbook of Human–Computer Interaction. Mahwah, NJ: Erlbaum* (2002), pp. 402–411 (cit. on pp. 28, 37).

[28] G. Catolino et al. "Gender diversity and women in software teams: How do they affect community smells?" In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Society (ICSE-SEIS)*. IEEE. 2019, pp. 11–20 (cit. on p. 40).

[29] L. Cen and D. Ruta. "A Map-Based Gender Prediction Model for Big E-Commerce Data". In: *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE. 2017, pp. 1025–1029 (cit. on p. 26).

[30] L. Chen et al. "Learning user embedding representation for gender prediction". In: *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE. 2016, pp. 263–269 (cit. on p. 26).

[31] T. Chintakovid and S. Wiedenbeck. "User perceptions and gender in end-user debugging: How do they affect outcomes?" In: *2009 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE. 2009, pp. 217–224 (cit. on p. 29).

[32] M. G. Christel and K. C. Kang. *Issues in requirements elicitation*. Tech. rep. Carnegie-Mellon Univ Pittsburgh Pa Software Engineering Inst, 1992 (cit. on pp. 36, 51, 54, 68).

[33] U. N. S. Conferences. *Beijing Declaration and Platform of Action, adopted at the Fourth World Conference on Women*. 1995 (cit. on p. 2).

[34] *CORE Rankings Portal*. visited in July 2020. URL: https://www.core.edu.au/conference-portal (cit. on p. 23).

[35] C. K. Coursaris, S. J. Swierenga, and E. Watrall. "An empirical investigation of color temperature and gender effects on web aesthetics". In: *Journal of usability studies* 3.3 (2008), pp. 103–117 (cit. on p. 29).

[36] J. Dick, E. Hull, and K. Jackson. *Requirements engineering*. Springer, 2017 (cit. on p. 11).

[37] E. Dinan et al. "Multi-Dimensional Gender Bias Classification". In: *arXiv preprint arXiv:2005.00614* (2020) (cit. on pp. 26, 28).

[38] C. Draude and S. Maaß. "Making IT work: integrating gender research in computing through a process model". In: *Proceedings of the 4th Conference on Gender & IT*. 2018, pp. 43–50 (cit. on p. 65).

[39] EIGE. *Effectiveness of Institutional Mechanisms for the Advancement of Gender Equality: Report*. 2014 (cit. on p. 1).

[40] I. El Asri and N. Kerzazi. "Where Are Females in OSS Projects? Socio Technical Interactions". In: *Working Conference on Virtual Enterprises*. Springer. 2019, pp. 308–319 (cit. on pp. 26, 29).

[41] N. Ensmenger. "Making programming masculine". In: *Gender codes: Why women are leaving computing* (2010), pp. 115–141 (cit. on p. 62).

[42] A. Feenberg. *Questioning technology*. Routledge, 2012 (cit. on p. 50).

[43] J. Feine et al. "Gender Bias in Chatbot Design". In: *International Workshop on Chatbot Research and Design*. Springer. 2019, pp. 79–93 (cit. on pp. 26, 28).

[44] *Femininities*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1129 (cit. on p. 17).

[45] M. Fernández-López, A. Gómez-Pérez, and N. Juristo. "Methontology: from ontological art towards ontological engineering". In: (1997) (cit. on pp. 7, 42, 46).

[46] J. E. Font and M. R. Costa-Jussa. "Equalizing gender biases in neural machine translation with word embeddings techniques". In: *arXiv preprint arXiv:1901.03116* (2019) (cit. on pp. 26, 28).

[47] D. Ford et al. "Paradise unplugged: Identifying barriers for female participation on stack overflow". In: *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering*. 2016, pp. 846–857 (cit. on pp. 26, 29, 33, 34, 56, 58–60, 62, 63, 65–67).

[48] G. Gao, A. Min, and P. C. Shih. "Gendered design bias: Gender differences of in-game character choice and playing style in league of legends". In: *Proceedings of the 29th Australian Conference on Computer-Human Interaction*. 2017, pp. 307–317 (cit. on pp. 26, 29, 34, 56, 65).

[49] E. Garbarino and M. Strahilevitz. "Gender differences in the perceived risk of buying online and the effects of receiving a site recommendation". In: *Journal of Business Research* 57.7 (2004), pp. 768–775 (cit. on pp. 29, 57).

[50] D. Gefen and C. M. Ridings. “If you spoke as she does, sir, instead of the way you do: a sociolinguistics perspective of gender differences in virtual communities”. In: *ACM SIGMIS Database: the DATABASE for Advances in Information Systems* 36.2 (2005), pp. 78–92 (cit. on p. 29).

[51] D. Gefen and D. W. Straub. “Gender differences in the perception and use of e-mail: An extension to the technology acceptance model”. In: *MIS quarterly* (1997), pp. 389–400 (cit. on pp. 26, 29).

[52] *Gender*. 2020. URL: https://eige.europa.eu/thesaurus/terms/1141 (cit. on pp. 16, 17, 48).

[53] *Gender (or sexual) division of labor*. 2021. URL: https://trainingcentre.unwomen.org/mod/glossary/view.php?id=36 (cit. on pp. 64, 65).

[54] *Gender Expression*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1177 (cit. on pp. 16, 48).

[55] *Gender Identity*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1179 (cit. on pp. 16, 48).

[56] *Gender Issue(s)*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1184 (cit. on p. 17).

[57] *Gender Norms*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1194 (cit. on pp. 16, 17, 49).

[58] *Gender Perspective*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1197 (cit. on p. 17).

[59] *Gender Socialisation*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1219 (cit. on p. 17).

[60] *Gender Stereotypes*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1222 (cit. on p. 17).

[61] *Gender System*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1225 (cit. on pp. 17, 49).

[62] A. R. Gila et al. “Impact of personality and gender diversity on software development teams’ performance”. In: *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*. IEEE. 2014, pp. 261–265 (cit. on p. 40).

[63] A. R. Gilal et al. “A rule-based model for software development team composition: Team leader role with personality types and gender classification”. In: *Information and Software Technology* 74 (2016), pp. 105–113 (cit. on p. 40).

[64] T.-T. Goh. “Exploring gender differences in SMS-based mobile library search system adoption.” In: *Educational Technology & Society* 14.4 (2011), pp. 192–206 (cit. on p. 29).

[65] C. S. González-González et al. "Gender Design Methods for Engineering, Responsible Innovation and Interaction". In: *Proceedings of the XIX International Conference on Human Computer Interaction*. 2018, pp. 1–2 (cit. on p. 28).

[66] C. Gralha, M. Goulão, and J. Araújo. "Analysing gender differences in building social goal models: a quasi-experiment". In: *2019 IEEE 27th International Requirements Engineering Conference (RE)*. IEEE. 2019, pp. 165–176 (cit. on pp. 27, 29).

[67] L. Gren. "On gender, ethnicity, and culture in empirical software engineering research". In: *2018 IEEE/ACM 11th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE. 2018, pp. 77–78 (cit. on p. 32).

[68] V. Grigoreanu et al. "Can feature design reduce the gender gap in end-user software development environments?" In: *2008 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE. 2008, pp. 149–156 (cit. on pp. 29, 33, 37, 62).

[69] E. O. Gyamfi et al. "deb2viz: Debiasing gender in word embedding data using subspace visualization". In: *Eleventh International Conference on Graphics and Image Processing (ICGIP 2019)*. Vol. 11373. International Society for Optics and Photonics. 2020, 113732F (cit. on pp. 26, 28).

[70] T. Hartmann and C. Klimmt. "Gender and computer games: Exploring females' dislikes". In: *Journal of Computer-Mediated Communication* 11.4 (2006), pp. 910–931 (cit. on p. 29).

[71] K. Hartzel. "How self-efficacy and gender issues affect software adoption and use". In: *Communications of the ACM* 46.9 (2003), pp. 167–171 (cit. on pp. 28, 59).

[72] O. Hazzan and Y. Dubinsky. "Can diversity in global software development be enhanced by agile software development?" In: *Proceedings of the 2006 international workshop on Global software development for the practitioner*. 2006, pp. 58–61 (cit. on p. 41).

[73] C. Hilderbrand et al. "Engineering Gender-Inclusivity into Software: Tales from the Trenches". In: *arXiv preprint arXiv:1905.10361* (2019) (cit. on pp. 27, 28, 32).

[74] C. G. Hill et al. "Gender-Inclusiveness Personas vs. Stereotyping: Can we have it both ways?" In: *Proceedings of the 2017 chi conference on human factors in computing systems*. 2017, pp. 6658–6671 (cit. on pp. 26, 28).

[75] C. C. Hsu. "Comparison of gender differences in young people's blog interface preferences and designs". In: *Displays* 33.3 (2012), pp. 119–128 (cit. on p. 37).

[76] Y. M. Hwang and K. C. Lee. "Using an eye-tracking approach to explore gender differences in visual attention and shopping attitudes in an online shopping environment". In: *International Journal of Human–Computer Interaction* 34.1 (2018), pp. 15–24 (cit. on pp. 26, 28, 57).

[77] R. Ibrahim. "A conceptual framework for supporting gender inclusivity in games". PhD thesis. University of Southampton, 2011 (cit. on pp. 3, 27, 29, 31–33, 37, 38, 61, 66).

[78] V. Ilie et al. "Gender differences in perceptions and use of communication technologies: A diffusion of innovation approach". In: *Information Resources Management Journal (IRMJ)* 18.3 (2005), pp. 13–31 (cit. on p. 29).

[79] N. Imtiaz et al. "Investigating the effects of gender bias on GitHub". In: *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE. 2019, pp. 700–711 (cit. on pp. 26, 29, 34, 60, 67).

[80] S. Jalali and C. Wohlin. "Systematic literature studies: database searches vs. backward snowballing". In: *Proceedings of the 2012 ACM-IEEE international symposium on empirical software engineering and measurement*. IEEE. 2012, pp. 29–38 (cit. on p. 13).

[81] B. Jesiek. "Democratizing software: Open source, the hacker ethic, and beyond". In: *First Monday* (2003) (cit. on p. 51).

[82] G. Kannabiran. "Gender and the design of technology-A critical analysis". In: (2012) (cit. on pp. 32, 34, 37).

[83] T. Katerina and P. Nicolaos. "Examining gender issues in perception and acceptance in web-based end-user development activities". In: *Education and Information Technologies* 23.3 (2018), pp. 1175–1202 (cit. on pp. 29, 33, 56–58).

[84] M. Kay, C. Matuszek, and S. A. Munson. "Unequal representation and gender stereotypes in image search results for occupations". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 2015, pp. 3819–3828 (cit. on pp. 26, 28, 49).

[85] S. Keele et al. *Guidelines for performing systematic literature reviews in software engineering*. Tech. rep. Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007 (cit. on p. 12).

[86] O. Keyes. "The misgendering machines: Trans/HCI implications of automatic gender recognition". In: *Proceedings of the ACM on Human-Computer Interaction* 2.CSCW (2018), pp. 1–22 (cit. on pp. 27, 28).

[87] B. Kitchenham and S. Charters. "Guidelines for performing systematic literature reviews in software engineering". In: (2007) (cit. on pp. 12, 14, 15).

[88] G. Kotonya and I. Sommerville. *Requirements Engineering: Processes and Techniques*. 1st. Wiley Publishing, 1998. ISBN: 0471972088 (cit. on pp. 9–11).

[89] T. Koulouri. "Gender differences in navigation dialogues with computer systems". PhD thesis. Brunel University, School of Information Systems, Computing and Mathematics, 2013 (cit. on p. 28).

[90] V. Kuechler, C. Gilbertson, and C. Jensen. "Gender differences in early free and open source software joining process". In: *IFIP International Conference on Open Source Systems*. Springer. 2012, pp. 78–93 (cit. on pp. 26, 29, 37).

[91] I. Latz et al. *Equal rights for women and girls in the world's constitutions*. 2014 (cit. on p. 1).

[92] S. Leavy. "Gender bias in artificial intelligence: The need for diversity and gender theory in machine learning". In: *Proceedings of the 1st international workshop on gender equality in software engineering*. 2018, pp. 14–16 (cit. on pp. 26, 28, 37).

[93] A. S. Lee. "Free/libre open source software contributors: one-time contributors, gender, and governance". PhD thesis. University of Alabama Libraries, 2019 (cit. on pp. 26, 29).

[94] B. Lenz, V. Kolarova, and K. Stark. "Gender Issues in the Digitalized 'Smart'Mobility World–Conceptualization and Empirical Findings Applying a Mixed Methods Approach". In: *International Conference on Human-Computer Interaction*. Springer. 2019, pp. 378–392 (cit. on pp. 26, 29, 31–34, 39, 56, 58, 65).

[95] P.-H. Lin and S.-C. Yeh. "How motion-control influences a VR-supported technology for mental rotation learning: from the perspectives of playfulness, gender difference and technology acceptance model". In: *International Journal of Human–Computer Interaction* 35.18 (2019), pp. 1736–1746 (cit. on pp. 29, 58).

[96] P.-C. Lin and L.-W. Chien. "The effects of gender differences on operational performance and satisfaction with car navigation systems". In: *International journal of human-computer studies* 68.10 (2010), pp. 777–787 (cit. on p. 28).

[97] X. Lin et al. "Can social role theory explain gender differences in Facebook usage?" In: *2013 46th Hawaii International Conference on System Sciences*. IEEE. 2013, pp. 690–699 (cit. on pp. 29, 57).

[98] Y. Lin. "Women in the free/libre open source software development". In: *Encyclopedia of gender and information technology*. IGI Global, 2006, pp. 1286–1291 (cit. on pp. 26, 29).

[99] D. Liu and X. Guo. "Exploring gender differences in acceptance of mobile computing devices among college students". In: *Information Systems and e-business Management* 15.1 (2017), pp. 197–223 (cit. on pp. 29, 33, 34, 56–58, 66).

[100] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: `https://github.com/joaomlourenco/novathesis/raw/master/template.pdf` (cit. on p. ii).

[101] R. K. Mark West and H. E. Chew. *I'd blush if I could: Closing gender divides in digital skills through education*. EQUALS and UNESCO, 2019 (cit. on pp. 2, 51, 62).

[102] N. Marsden and M. Haag. "Evaluation of gendermag personas based on persona attributes and persona gender". In: *International Conference on Human-Computer Interaction*. Springer. 2016, pp. 122–127 (cit. on pp. 27, 28).

[103] N. Marsden, J. Hermann, and M. Pröbster. "Developing personas, considering gender: a case study". In: *Proceedings of the 29th Australian Conference on Computer-Human Interaction*. 2017, pp. 392–396 (cit. on pp. 27, 28, 30, 32, 38).

[104] D. P. Martin. "Gender Equality With Agile In Software Engineering". In: (2011) (cit. on pp. 59, 67).

[105] D. Martin and K. Ellis. "Playing the game: Effective gender role analysis techniques for computer games". In: *Proceedings of the 23rd Australian Computer-Human Interaction Conference*. 2011, pp. 185–193 (cit. on pp. 27, 29).

[106] *Masculinities*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1285 (cit. on p. 17).

[107] A. May, J. Wachs, and A. Hannák. "Gender differences in participation and reward on Stack Overflow". In: *Empirical Software Engineering* 24.4 (2019), pp. 1997–2019 (cit. on pp. 26, 29, 34, 37, 63, 66, 67).

[108] M. McDonnell and D. Baxter. "Chatbots and gender stereotyping". In: *Interacting with Computers* 31.2 (2019), pp. 116–121 (cit. on p. 28).

[109] T. McGill and N. Thompson. "Gender differences in information security perceptions and behaviour". In: (2018) (cit. on pp. 29, 57).

[110] C. Mendez et al. "From GenderMag to InclusiveMag: An Inclusive Design Meta-Method". In: *2019 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE. 2019, pp. 97–106 (cit. on pp. 27, 28).

[111] D. A. Menéndez. *Damegender: Writing and Comparing Gender Detection Tools*. Tech. rep. EasyChair, 2020 (cit. on p. 26).

[112] D. Metaxa-Kakavouli et al. "Gender-inclusive design: Sense of belonging and bias in web interfaces". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–6 (cit. on pp. 26, 29, 33, 56, 59–62).

[113] E. Moon. "Gendered patterns of politeness in free/libre open source software development". In: *2013 46th Hawaii International Conference on System Sciences*. IEEE. 2013, pp. 3168–3177 (cit. on pp. 26, 29, 66).

[114] S. Morgan. "How are programming questions from women received on stack overflow? a case study of peer parity". In: *Proceedings Companion of the 2017 ACM SIGPLAN International Conference on Systems, Programming, Languages, and Applications: Software for Humanity*. 2017, pp. 39–41 (cit. on pp. 26, 29, 67).

[115] G. Moss, R. Gunn, and J. Heller. "Some men like it black, some women like it pink: consumer implications of differences in male and female website design". In: *Journal of Consumer behaviour* 5.4 (2006), pp. 328–341 (cit. on p. 29).

[116] N. L. Muscanell and R. E. Guadagno. "Make new friends or keep the old: Gender and personality differences in social networking use". In: *Computers in Human Behavior* 28.1 (2012), pp. 107–112 (cit. on pp. 29, 33, 63).

[117] D. Nafus. "'Patches don't have gender': What is not open in open source software". In: *New Media & Society* 14.4 (2012), pp. 669–683 (cit. on p. 67).

[118] G. Nesti. "Mainstreaming gender equality in smart cities: Theoretical, methodological and empirical challenges". In: *Information Polity* 24.3 (2019), pp. 289–304 (cit. on pp. 26, 29, 65).

[119] H. Nissenbaum. "How computer systems embody values". In: *Computer* 34.3 (2001), pp. 120–119 (cit. on p. 52).

[120] N. F. Noy, D. L. McGuinness, et al. *Ontology development 101: A guide to creating your first ontology*. 2001 (cit. on pp. 7, 42).

[121] B. Nuseibeh and S. Easterbrook. "Requirements engineering: a roadmap". In: *Proceedings of the Conference on the Future of Software Engineering*. 2000, pp. 35–46 (cit. on pp. 9, 11).

[122] U. N. H. R. Office. *Definitions*. URL: https://www.unfe.org/definitions/ (visited on 11/13/2020) (cit. on p. 48).

[123] J. Otterbacher et al. "Investigating user perception of gender bias in image search: the role of sexism". In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2018, pp. 933–936 (cit. on pp. 26, 28).

[124] R. Ottoni et al. "Ladies first: Analyzing gender roles and behaviors in pinterest". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 7. 1. 2013 (cit. on p. 29).

[125] N. Oudshoorn, E. Rommes, and M. Stienstra. "Configuring the user as everybody: Gender and design cultures in information and communication technologies". In: *Science, Technology, & Human Values* 29.1 (2004), pp. 30–63 (cit. on pp. 40, 51).

[126] K. Oyibo, Y. S. Ali, and J. Vassileva. "Gender difference in the credibility perception of mobile websites: a mixed method approach". In: *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. 2016, pp. 75–84 (cit. on pp. 29, 33, 56, 58, 60, 62).

[127] L. Palmquist and P. Wennberg. "SATIN 2: How to support self-efficacy and diversity in end-user development". In: *Equality, Growth and Innovation-In Theory and Practice: 09/10/2013-10/10/2013*. 2013 (cit. on pp. 26, 29, 30, 32, 38, 63).

[128] K. Petersen et al. “Systematic Mapping Studies in Software Engineering”. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*. EASE’08. Italy: BCS Learning & Development Ltd., 2008, pp. 68–77 (cit. on pp. 12, 14).

[129] M. Petticrew and H. Roberts. *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008 (cit. on pp. 13, 16).

[130] A. Popescu and G. Grefenstette. “Image tagging and search: a gender oriented study”. In: *Proceedings of second ACM SIGMM workshop on Social media*. 2010, pp. 9–14 (cit. on pp. 26, 28).

[131] Y. Qian et al. “Reducing gender bias in word-level language models with a gender-equalizing loss function”. In: *arXiv preprint arXiv:1905.12801* (2019) (cit. on p. 26).

[132] A. M. G. Ramos and T. Rojas-Rajs. “Inclusion of Gender Perspective in Design and IT Environments”. In: *Proceedings of the XVII International Conference on Human Computer Interaction*. 2016, pp. 1–4 (cit. on pp. 34, 37, 51, 54, 59).

[133] A. Rashid, A. Moreira, and J. Araújo. “Modularisation and composition of aspectual requirements”. In: *Proceedings of the 2nd international conference on Aspect-oriented software development*. 2003, pp. 11–20 (cit. on p. 11).

[134] N. Rekabsaz and M. Schedl. “Do Neural Ranking Models Intensify Gender Bias?” In: *arXiv preprint arXiv:2005.00372* (2020) (cit. on pp. 26, 28).

[135] S. Revate et al. “SWT Multi-Level Fingerprint Feature Combination for Gender Classification: SVM Approach.” In: (2017) (cit. on p. 26).

[136] N. Robson. “Diversity and decorum in open source communities”. In: *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2018, pp. 986–987 (cit. on pp. 26, 29).

[137] J. A. Rode. “A theoretical agenda for feminist HCI”. In: *Interacting with Computers* 23.5 (2011), pp. 393–400 (cit. on p. 49).

[138] J. A. Rode and E. S. Poole. “Putting the gender back in digital housekeeping”. In: *Proceedings of the 4th Conference on Gender & IT*. 2018, pp. 79–90 (cit. on pp. 27, 28, 37, 49, 51, 61).

[139] S. Rodgers and M. A. Harris. “Gender and e-commerce: An exploratory study”. In: *Journal of advertising research* 43.3 (2003), pp. 322–329 (cit. on p. 29).

[140] E. Rommes. “Gender senititve design practices”. In: *Encyclopedia of gender and information technology*. IGI Global, 2006, pp. 675–681 (cit. on p. 51).

[141] P. Rossetti. “Gender differences in e-mail communication”. In: *The Internet TESL Journal* 4.7 (1998), pp. 1–6 (cit. on pp. 26, 29).

[142] M. B. Rosson, H. Sinha, and T. Edor. “Design planning in end-user web development: gender, feature exploration and feelings of success”. In: *2010 IEEE Symposium on Visual Languages and Human-Centric Computing*. IEEE. 2010, pp. 141–148 (cit. on p. 29).

[143] M. Rowan and J. Dehlinger. “Observed gender differences in privacy concerns and behaviors of mobile device end users”. In: *Procedia Computer Science* 37 (2014), pp. 340–347 (cit. on pp. 29, 33, 37, 56).

[144] M. K. Scheuerman, J. M. Paul, and J. R. Brubaker. “How computers see gender: An evaluation of gender classification in commercial facial analysis services”. In: *Proceedings of the ACM on Human-Computer Interaction* 3.CSCW (2019), pp. 1–33 (cit. on pp. 27, 28).

[145] A. Schlesinger, W. K. Edwards, and R. E. Grinter. “Intersectional HCI: Engaging identity through gender, race, and class”. In: *Proceedings of the 2017 CHI conference on human factors in computing systems*. 2017, pp. 5412–5427 (cit. on pp. 27, 28, 32, 37, 49, 51).

[146] R. of the Secretary-General. *Special edition: progress towards the Sustainable Development Goals*. United Nations, 2019 (cit. on p. 1).

[147] P. Sengers. “The engineering of experience”. In: *Funology 2*. Springer, 2018, pp. 287–299 (cit. on p. 40).

[148] P. Sengers et al. “Reflective design”. In: *Proceedings of the 4th decennial conference on Critical computing: between sense and sensibility*. 2005, pp. 49–58 (cit. on p. 40).

[149] *Sex*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1361 (cit. on pp. 16, 48).

[150] *Sexism*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1367 (cit. on p. 17).

[151] N. (Sey A. & Hafkin. *Taking stock: Data and evidence on gender equality in digital access, skills and leadership*. 2019 (cit. on pp. 2, 36, 51, 66).

[152] Z. Sharafi et al. “Women and men—Different but equal: On the impact of identifier style on source code reading”. In: *2012 20th IEEE International Conference on Program Comprehension (ICPC)*. IEEE. 2012, pp. 27–36 (cit. on p. 29).

[153] S. Sharma and S. Pandey. “Revisiting requirements elicitation techniques”. In: *International Journal of Computer Applications* 75.12 (2013) (cit. on p. 9).

[154] H. Sharp, A. Finkelstein, and G. Galal. “Stakeholder identification in the requirements engineering process”. In: *Proceedings. Tenth International Workshop on Database and Expert Systems Applications. DEXA 99*. Ieee. 1999, pp. 387–391 (cit. on p. 54).

[155] N. Shekhawat, A. Chauhan, and S. B. Muthiah. "Algorithmic Privacy and Gender Bias Issues in Google Ad Settings". In: *Proceedings of the 10th ACM Conference on Web Science*. 2019, pp. 281–285 (cit. on pp. 27, 28).

[156] W. Shen. *Using machine learning and natural language processing to replace gender biased words within free-form text*. US Patent 10,467,339. 2019 (cit. on p. 26).

[157] D. Showkat and C. Grimm. "Identifying gender differences in information processing style, self-efficacy, and tinkering for robot tele-operation". In: *2018 15th International Conference on Ubiquitous Robots (UR)*. IEEE. 2018, pp. 443–448 (cit. on p. 28).

[158] S. Silessi, C. Varol, and M. Karabatak. "Identifying gender from SMS text messages". In: *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2016, pp. 488–491 (cit. on p. 26).

[159] S. J. Simon. "The impact of culture and gender on web sites: an empirical study". In: *ACM SIGMIS Database: the DATABASE for Advances in Information Systems* 32.1 (2000), pp. 18–37 (cit. on pp. 26, 29, 33, 62).

[160] P. Smith and K. Ricanek. "Mitigating algorithmic bias: Evolving an augmentation policy that is non-biasing". In: *Proceedings of the IEEE Winter Conference on Applications of Computer Vision Workshops*. 2020, pp. 90–97 (cit. on pp. 26, 28).

[161] I. Sommerville. "Software engineering 9th Edition". In: *ISBN-10* 137035152 (2011), p. 18 (cit. on p. 11).

[162] O. of the Special Adviser on Gender Issues and the Advancement of Women. *Important Concepts Underlying Gender Mainstreaming*. United Nations, 2001 (cit. on pp. 16, 17, 49, 64).

[163] K. Spiel, O. L. Haimson, and D. Lottridge. "How to do better with gender on surveys: a guide for HCI researchers". In: *interactions* 26.4 (2019), pp. 62–65 (cit. on p. 49).

[164] B. Spieler and W. Slany. "Game development-based learning experience: Gender differences in game design". In: *arXiv preprint arXiv:1805.04457* (2018) (cit. on p. 29).

[165] *Structural Inequality*. 2021. URL: https://eige.europa.eu/thesaurus/terms/1399 (cit. on p. 17).

[166] S. Stumpf et al. "Gender-Inclusive HCI Research and Design: A Conceptual Review". In: *Foundations and Trends® in Human–Computer Interaction* 13.1 (2019) (cit. on pp. 27, 28).

[167] N. Subrahmaniyan et al. "Testing vs. code inspection vs. what else? Male and female end users' debugging strategies". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2008, pp. 617–626 (cit. on p. 29).

[168] A. Sullivan. “Gender-inclusive quest design in massively multiplayer online role-playing games”. In: *Proceedings of the 4th international conference on foundations of digital games*. 2009, pp. 354–356 (cit. on p. 29).

[169] M. Szell and S. Thurner. “How women organize social networks different from men”. In: *Scientific reports* 3 (2013), p. 1214 (cit. on pp. 29, 66).

[170] R. Tang et al. “Mitigating Gender Bias in Captioning Systems”. In: *arXiv preprint arXiv:2006.08315* (2020) (cit. on p. 26).

[171] B. Tay, Y. Jung, and T. Park. “When stereotypes meet robots: the double-edge sword of robot gender and personality in human–robot interaction”. In: *Computers in Human Behavior* 38 (2014), pp. 75–84 (cit. on pp. 26, 28).

[172] *Transforming our world : the 2030 Agenda for Sustainable Development*. 2015 (cit. on p. 2).

[173] K. Tzafilkou et al. “Gender-based behavioral analysis for end-user development and the ‘RULES’attributes”. In: *Education and Information Technologies* 22.4 (2017), pp. 1853–1894 (cit. on pp. 3, 29, 30, 32, 33, 38, 56–59, 63).

[174] I. w. o. f. UNESCO. Director-General 2009-2017 (Bokova. *Cracking the code: Girls’ and women’s education in science, technology, engineering and mathematics (STEM)*. United Nations Educational, Scientific and Cultural Organizations, 2017. ISBN: 978-92-3-100233-5 (cit. on p. 2).

[175] *Universal Declaration of Human Rights*. 1948 (cit. on p. 1).

[176] M. Uschold, M. Gruninger, et al. “Ontologies: Principles, methods and applications”. In: *TECHNICAL REPORT-UNIVERSITY OF EDINBURGH ARTIFICIAL INTELLIGENCE APPLICATIONS INSTITUTE AIAI TR* (1996) (cit. on pp. 43, 46).

[177] A. Van Lamsweerde. “Goal-oriented requirements engineering: A guided tour”. In: *Proceedings fifth ieee international symposium on requirements engineering*. IEEE. 2001, pp. 249–262 (cit. on p. 11).

[178] B. Vasilescu et al. “Gender and tenure diversity in GitHub teams”. In: *Proceedings of the 33rd annual ACM conference on human factors in computing systems*. 2015, pp. 3789–3798 (cit. on p. 40).

[179] M. Vorvoreanu et al. “From gender biases to gender-inclusive design: An empirical investigation”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 2019, pp. 1–14 (cit. on pp. 26, 28, 32, 54, 61).

[180] X. Wang et al. “Software as a social artifact: a management and evolution perspective”. In: *International Conference on Conceptual Modeling*. Springer. 2014, pp. 321–334 (cit. on p. 51).

[181] X. Wang and Y. Lespérance. “Agent-oriented requirements engineering using ConGolog and i*”. In: *Agent-Oriented Information Systems Workshop (AOIS-2001). Montreal, Canada*. 2001, pp. 59–78 (cit. on p. 11).

[182] Y. Wang. "Understanding the Reputation Differences between Women and Men on Stack Overflow". In: *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*. IEEE. 2018, pp. 436–444 (cit. on pp. 26, 29, 34).

[183] M. Warkentin and N. Malimage. "Overcoming mixed-gender requirements misspecification with the Modified Coherence Method". In: (2012) (cit. on pp. 29, 31, 32, 38).

[184] D. Williams et al. "The virtual census: Representations of gender, race and age in video games". In: *New Media & Society* 11.5 (2009), pp. 815–834 (cit. on p. 29).

[185] G. Williams. "Are you sure your software is gender-neutral?" In: *Interactions* 21.1 (2014), pp. 36–39 (cit. on pp. 4, 26, 28, 37, 38, 51, 54).

[186] R. Williams and D. Edge. "The social shaping of technology". In: *Research policy* 25.6 (1996), pp. 865–899 (cit. on p. 50).

[187] J. Winn and C. Heeter. "Gaming, gender, and time: Who makes time to play?" In: *Sex roles* 61.1-2 (2009), pp. 1–13 (cit. on pp. 29, 34, 65).

[188] C. Wohlin. "Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering". In: *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*. EASE '14. London, England, United Kingdom: Association for Computing Machinery, 2014. ISBN: 9781450324762. DOI: 10.1145/2601248.2601268. URL: https://doi.org/10.1145/2601248.2601268 (cit. on p. 22).

[189] C. Wohlin et al. *Experimentation in software engineering*. Springer Science & Business Media, 2012 (cit. on pp. 35, 84, 105).

[190] Z. Yang and J. Feng. "A Causal Inference Method for Reducing Gender Bias in Word Embedding Relations." In: *AAAI*. 2020, pp. 9434–9441 (cit. on pp. 26, 28).

[191] E. S. Yu. "Social Modeling and i". In: *Conceptual modeling: Foundations and applications: Essays in honor of John Mylopoulos*. 2009, pp. 99–121 (cit. on p. 50).

[192] P. Zave. "Classification of research efforts in requirements engineering". In: *ACM Computing Surveys (CSUR)* 29.4 (1997), pp. 315–321 (cit. on p. 9).

[193] S. Zdenek. "Rising up from the MUD: Inscribing gender in software design". In: *Discourse & Society* 10.3 (1999), pp. 379–409 (cit. on pp. 27, 28).

[194] H. Zhang, M. A. Babar, and P. Tell. "Identifying relevant studies in software engineering". In: *Information and Software Technology* 53.6 (2011), pp. 625–637 (cit. on p. 13).

A

Data extraction form

Paper Information	
ID	
Title	
Conference / Journal / Workshop	
Year	
Quality Assessment	
Number of citations	
Digital Library	
Core Rank	
Data Extraction	
Gender conceptualization	
Gender issues conceptualization	
Software application domain	
Characterization of users	
Description of the gender issues found / addresssed	
Description of the proposed approach to solve the gender issues	
Software development phase of the approach	
Validation of approach	
Additional information	

Figure A.1: Form used for the data extraction of the systematic mapping study

B

Evaluation Questionnaire

Conceptual Model of Gender-Inclusive Requirements

The purpose of this questionnaire is to evaluate a conceptual model on gender-inclusive requirements developed in the context of the MSc dissertation in Computer Science and Informatics of Inês Nunes, student at NOVA School of Science of Technology.

The conceptual model aims to be a starting point for supporting the elicitation of gender-inclusive requirements.

You will find the guide of the conceptual model in the following link:
https://drive.google.com/file/d/1_GOcBM-ronXlTxduFLaa8FTIaSvwLE2P/view?usp=sharing

Please read the guide before answering this questionnaire.

Your answers are anonymous, and any personal information will be kept strictly confidential.
The final MSc dissertation may present anonymised answers.

This questionnaire will not take longer than 15 minutes to answer. If you are interested in the results of this work, please contact me.

Thank you so much for your help and time to participate in this survey. Your availability is very much appreciated.

Inês Nunes (ir.nunes@campus.fct.unl.pt)
August 2021

*Obrigatório

1. Email

___

Pre-survey. Profile

2. What is your age? *

___

3. What is your gender? *

If you prefer to self-describe, please fill the option "Other".

Marcar apenas uma oval.

- Woman
- Man
- Non-binary
- Prefer not to disclose
- Outra: ____________________

4. What is your level of academic training? *

Marcar apenas uma oval.

- Bachelor
- Master's Degree
- PhD
- Outra: ____________________

5. What is the scientific domain of your degree? *

E.g., Computer Science

____

6. What is your current occupation? *

Marcar apenas uma oval.

- Practitioner (e.g., programmer, software engineer, analyst, etc.)
- Academic (e.g., Professor, lecturer, etc.)
- Researcher (e.g., Post-doc, PhD candidate, etc.)
- Student (e.g., MSc, bachelor, etc.)
- Outra: ____________________

7. What is your experience with conceptual modeling? *

Marcar apenas uma oval.

- 0-2 years
- 3-5 years
- 6-10 years
- 10+ years

8. How important is gender inclusion in software development? *

Marcar apenas uma oval.

	1	2	3	4	5	
Not Important						Very Important

9. Which best describes your perceptions of gender issues in software development? *

Marcar apenas uma oval.

- Persistent structural inequalities with statistical evidence.
- Inequalities are mainly due to culture and education.
- There are no barriers because there is female presence and inequalities are circumstantial.
- I don't have an opinion.

10. In which situations is gender inclusion relevant in your professional activities? *

Please check all that apply.

Marcar tudo o que for aplicável.

- [ ] Management (e.g., career opportunities, leadership roles, working conditions)
- [ ] Selection of project team members
- [ ] Selection of the approaches for the elicitation of requirements
- [ ] Selection of the population for a project/product/prototype evaluation
- [ ] Data collection, report and analysis
- [ ] Selection of research, curricula or studying materials

Outra: [ ] ______________________________

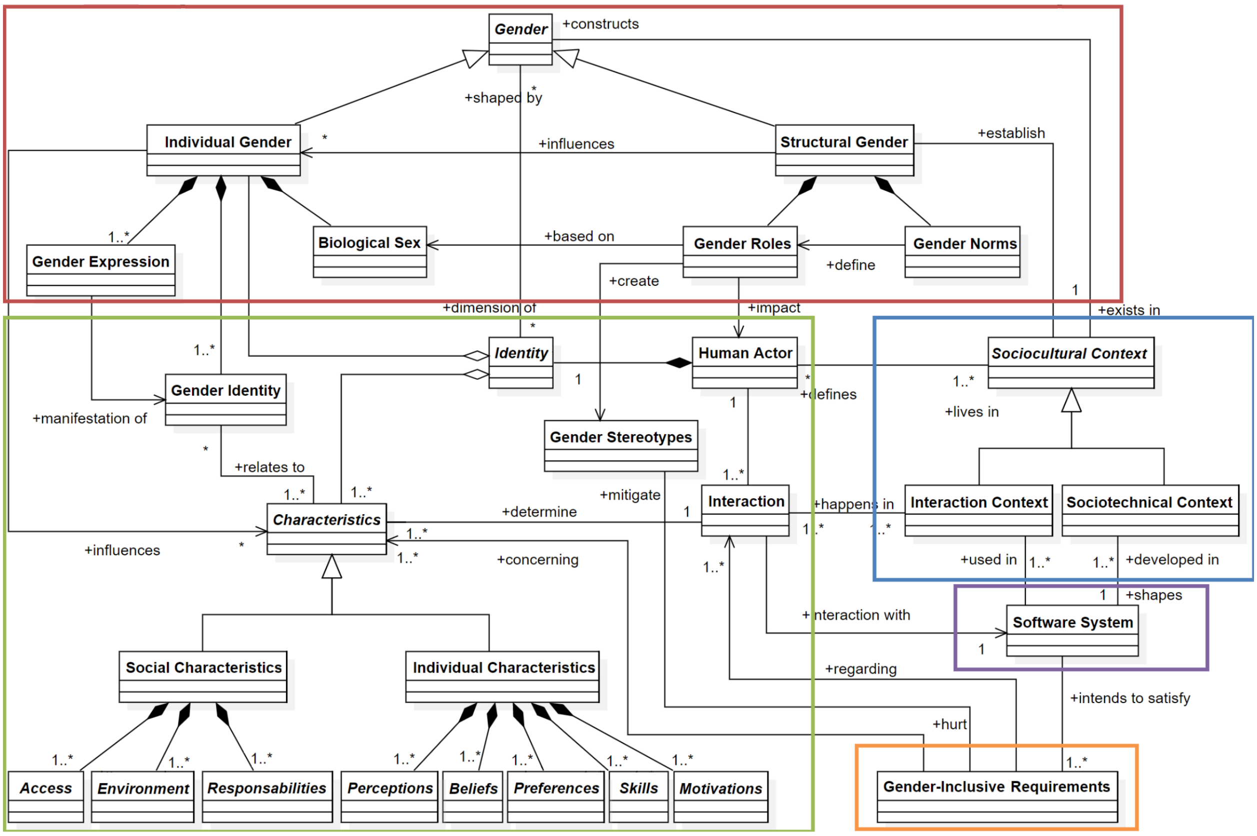
11. What approaches for addressing gender issues in software development do you know?

____

Overview of the Gender-Inclusive Conceptual Model

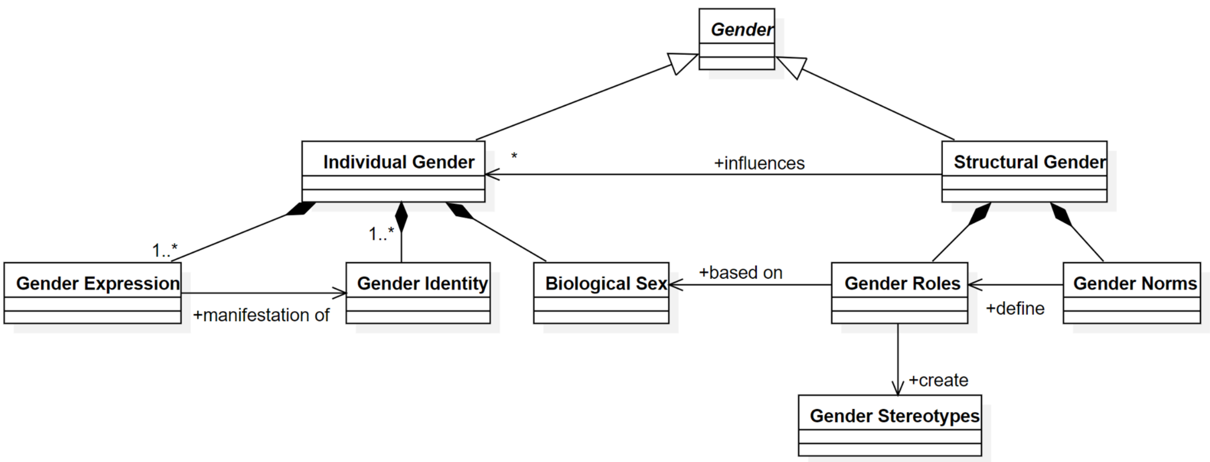
The gender-inclusive conceptual model provides a representation of how gender issues may arise in the software development process and how they can be mitigated by taking a gender perspective in requirements elicitation.

It is composed of four components: Gender, Sociocultural Context, Human Actor, and Software System. The questions in the next sections refer to the components of this model.



Part 1. Gender Component

The Gender component of the model describes gender as a structural feature of society and as a complex part of a Human Actor's identity.



12. Does this model capture the concept of gender as perceived by individuals (self-identity) and society? *

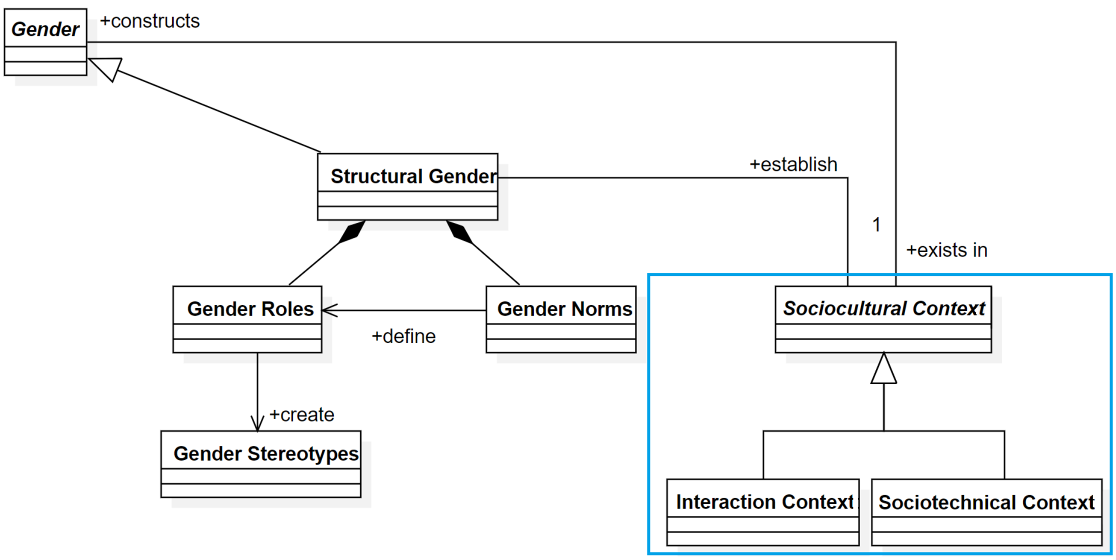
Marcar apenas uma oval.

- Yes
- No

13. If your answer is “No”, please list the concepts (or relationships) you would change or add:

Part 2. Sociocultural Context Component

The Sociocultural Context component model defines the social and cultural factors and events of a particular time period that influence those who live in it.



14. Are these concepts enough for capturing the relation of gender to the sociocultural contexts of software use and development? *

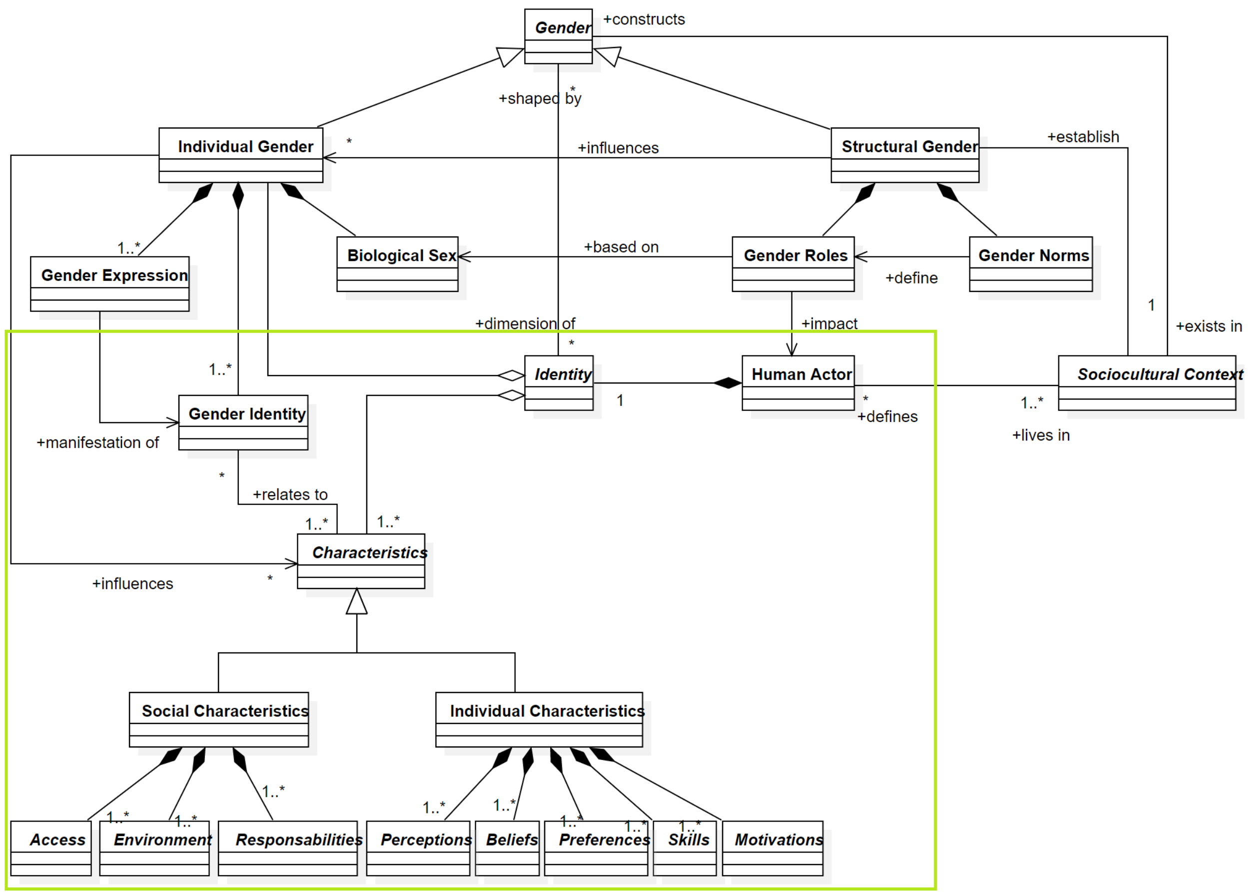
Marcar apenas uma oval.

- Yes
- No

15. If you answered “No”, please list the concepts (or relationships) you would change or add:

Part 3. Human Actor Component

The Human Actor component model represents the individuals involved with a software system in a sociocultural context. These have a unique identity and characteristics and represent the users' attitudes towards software that were empirically found to be influenced by gender.



16. Are these concepts enough for capturing a person with a unique identity of individual and social characteristics influenced by their gender? *

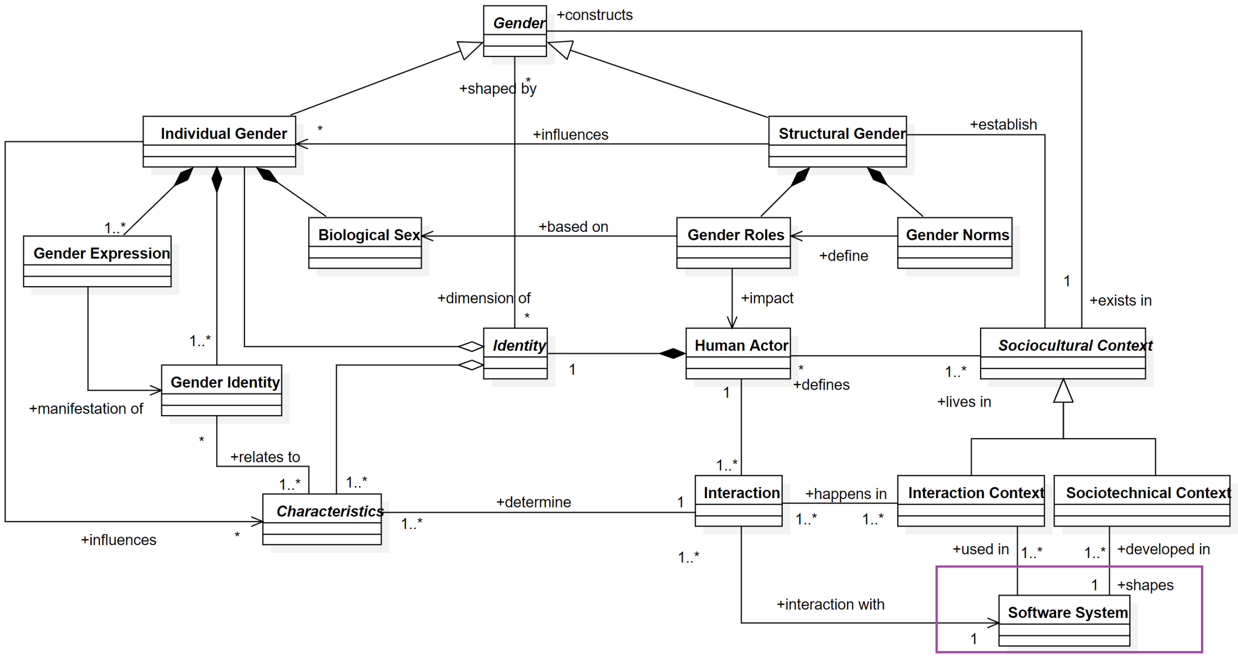
Marcar apenas uma oval.

- Yes
- No

17. If you answered “No”, please list the concepts (or relationships) you would change or add:

Part 4. Software System Component

The Software System component model describes the software system that intends to satisfy gender-inclusive requirements.



18. Are these concepts enough for capturing a software system as the product of human-based decisions in a sociocultural context influenced by gender? *

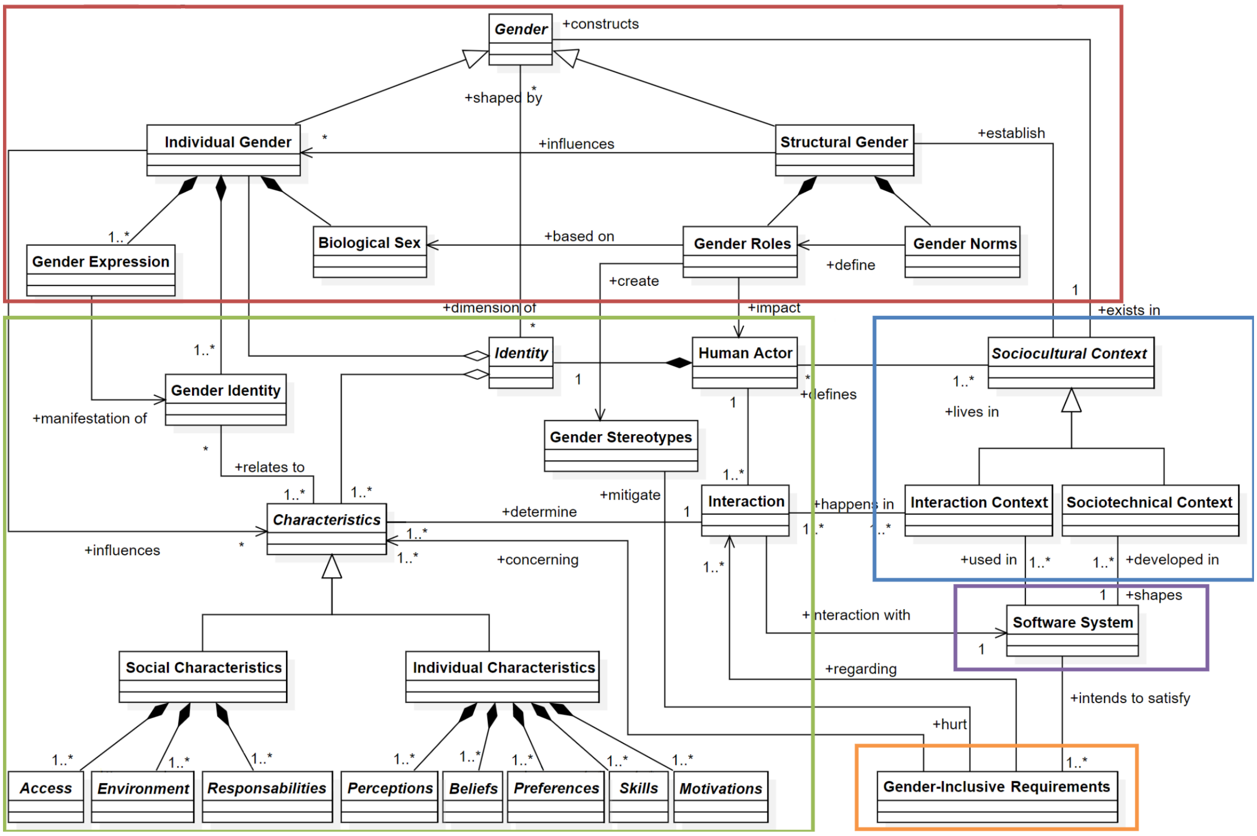
Marcar apenas uma oval.

- Yes
- No

19. If you answered "No", please list the concepts (or relationships) you would change or add:

Overall opinion about the conceptual model

For each of the following questions, please provide the answer that better reflects your opinion about this model.



20. How useful is this conceptual model? *

Marcar apenas uma oval.

	1	2	3	4	5	
Not useful	◯	◯	◯	◯	◯	Very useful

21. How difficult is it to understand this conceptual model? *

Marcar apenas uma oval.

	1	2	3	4	5	
Not Difficult	◯	◯	◯	◯	◯	Very Difficult

22. Would you use this conceptual model? *

Marcar apenas uma oval.

- Yes
- No
- Perhaps

23. Would you recommend this conceptual model? *

Marcar apenas uma oval.

- Yes
- No
- Perhaps

24. To whom can this conceptual model be useful?

Marcar tudo o que for aplicável.

- [ ] Project Managers
- [ ] Software Analysts
- [ ] Software Architects
- [ ] Software Engineers
- [ ] Software Developers
- [ ] Researchers

Outra: [ ] ______________________

25. What are the strongest points about this conceptual model?

26. What are the weakest points of this conceptual model?

Guide evaluation

Pease choose the option that better suits your opinion about the guide provided for the completion of this questionnaire.

27. Was the guide useful? *

Marcar apenas uma oval.

- Yes
- No

28. Was the guide easy to understand? *

Marcar apenas uma oval.

- Yes
- No

C

Presentation Guide

A Conceptual Model for Gender-Inclusive Requirements

Motivation

- Globally, gender inequality, gender-based stereotypes, social norms and practices persist
- Technology plays a fundamental role in shaping cultural and social beliefs and attitudes

Main gender issues

- **Software is assumed to be gender-neutral**
  Predominantly male development teams assume software technologies as exclusively technical and gender-neutral. They assume their needs and values are universally applicable

- **Gender is conceptualized as binary**
  If gender is addressed, it is simplified to a statistical binary category associated with biological sex

- **Lack of approaches**
  When awareness is raised, few support exists to address gender issues in the development process

Goal of this work

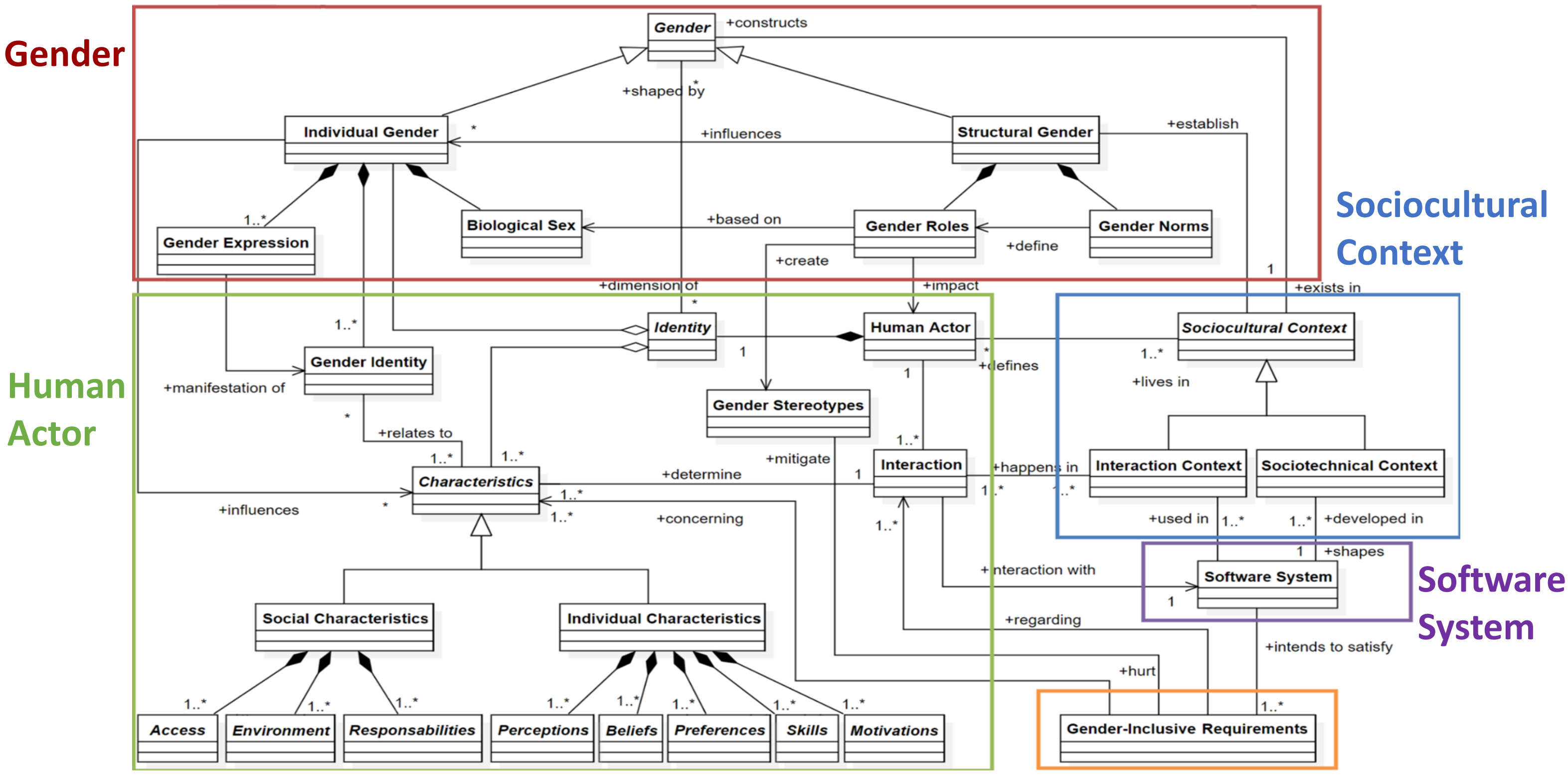
Define a gender inclusive conceptual model to assist in the construction of software systems that satisfy gender-inclusive requirements

- A common structured vocabulary for describing key gender-related concepts
  - intersection between gender and software technologies exploring their mutually influencing relationships
  - gender as a social construct and in software development and use

A gender inclusive conceptual model

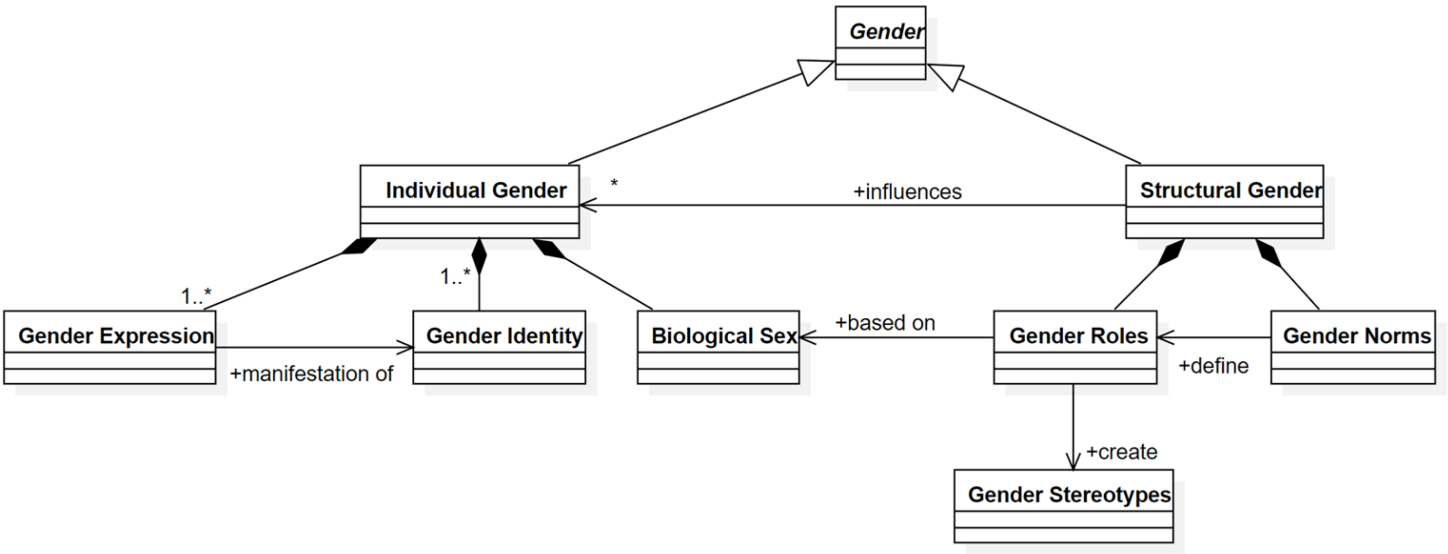
- Composed of four components (or parts):
  - Gender
  - Sociocultural Context
  - Human Actor
  - Software System

The 4 components of the gender inclusive conceptual model



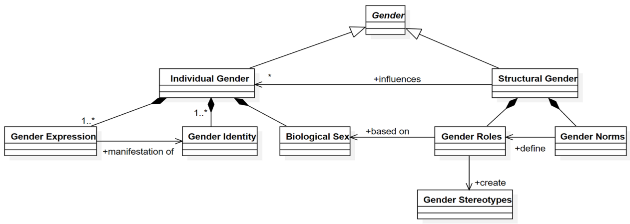
The Gender component model

- Defines two complementary perspectives of gender:
  - a subjectively held self-identity, biological sex, and a self-expression to others
  - a set of norms and roles imposed by society, resulting in gender-stereotypes



Gender component concepts defined

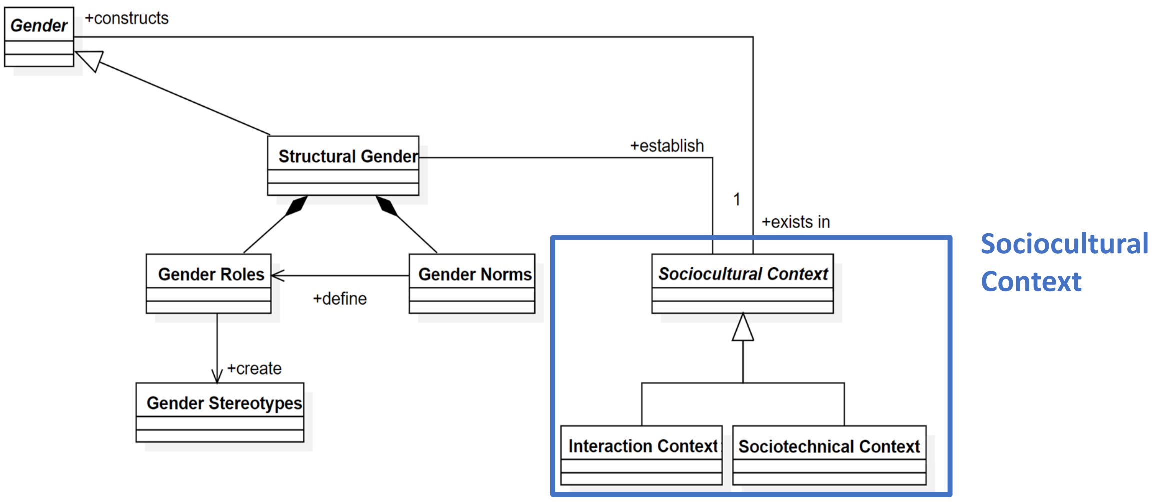
- **Gender:** structural feature of society and a complex part of a Human Actor's identity
- **Structural Gender:** social and culturally constructed norms, opportunities, and expectations imposed on individuals based on their perceived sex



- **Gender Norms:** standards and expectations to which women and men generally conform, within a range that defines a particular society, culture, and community at that point in time
- **Gender Roles:** internalized and embodied social expectations learned through socialization processes in which women and men behave accordingly
- **Gender stereotypes:** preconceived ideas whereby females and males are arbitrarily assigned characteristics and roles determined and limited by their gender
- **Individual Gender:** person's self-identified gender, which is one part of their identity. It is a continuum where the Human Actors can freely align themselves and construct their fluid and complex identities
- **Gender Identity:** person's individual experience of gender, independent of sex
- **Gender Expression:** individual choice about how a person wants to communicate their gender identity, and it can vary freely at any time
- **Biological Sex:** biological characteristics that determine whether an individual is female, male, or intersex

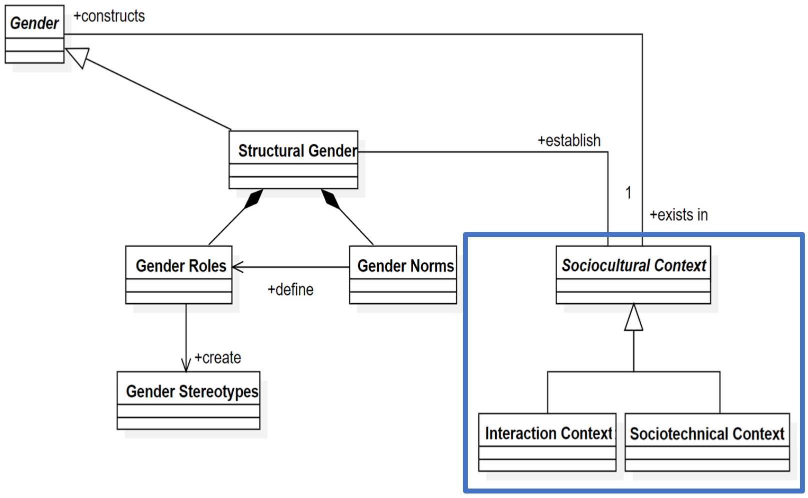
The Social Context component model

- Expresses the **sociocultural** asymmetry between the context where software is developed and where it is used, which includes:
  - a highly diverse **interaction context** (behaviours, preferences, needs)
  - **sociotechnical context** that can be a very narrow subset of the population: mostly men with high levels of education and income



Sociocultural Context component concepts defined

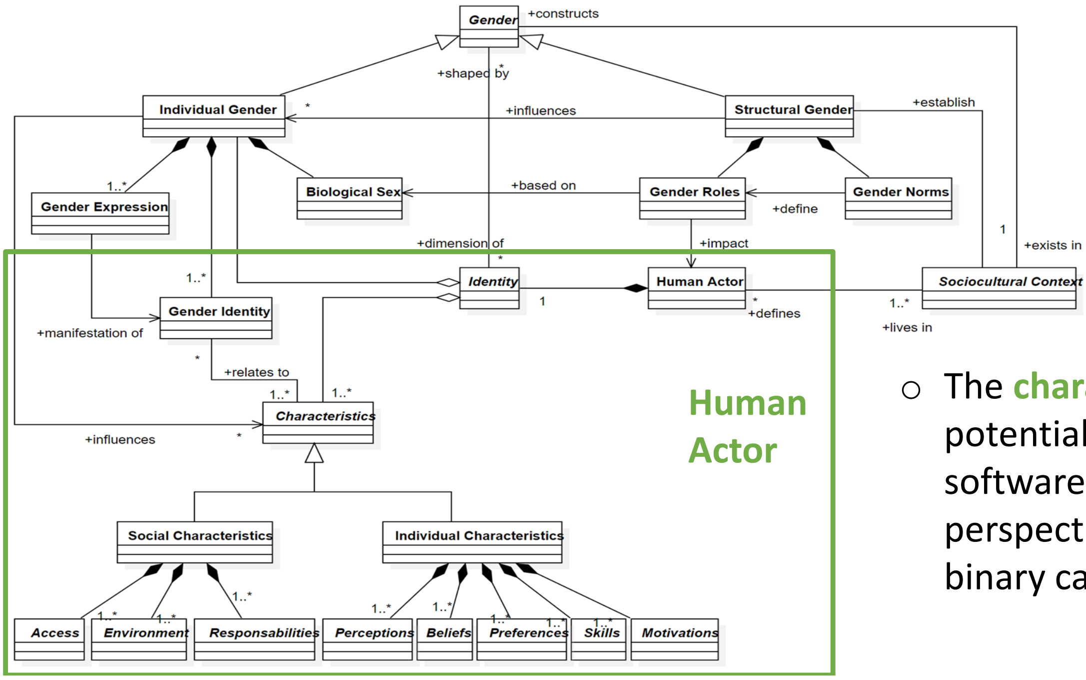
- **Sociocultural Context:** social and cultural factors and events of a particular time period that influence those who live in it
- **Interaction Context:** range of sociocultural factors and contextual settings where users interact with the software system



- **Sociotechnical Context:** range of social factors and contextual influences, such as the needs and values of development teams, stakeholders, organizations, and communities.

The Human Actor component model

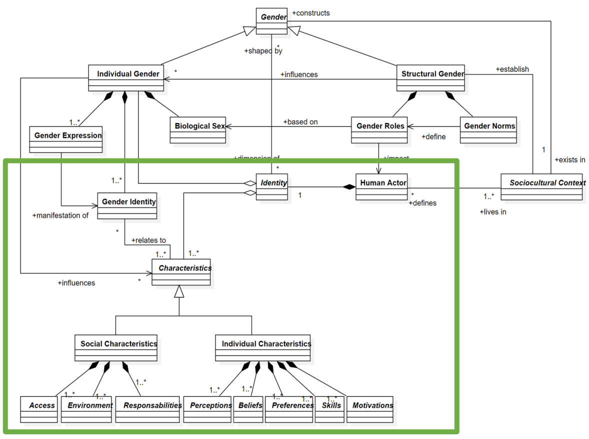
- Expresses the unique identities of users composed of individual and social characteristics that are influenced by their gender



- The **characteristics** explore the potential attitudes towards software through a gender perspective without resorting to binary categories of users.

Human Actor component concepts defined

- **Human Actor:** individuals with a unique identity that are involved with the software system in a sociocultural context
- **Identity:** distinguishing and unique character and personality of a human actor
- **Characteristics:** individual and social traits of the identity of a Human Actor that are influenced by their individual gender and determine their interaction with a software system



- **Individual Characteristics**: personal attributes of a human actor (perceptions, beliefs, preferences, skills, and motivations)
- **Social Characteristics:** attributes of a human actor in relation to other human actors (access to technological resources, their social environment, and their responsibilities)

Human Actor's Characteristics concepts defined *

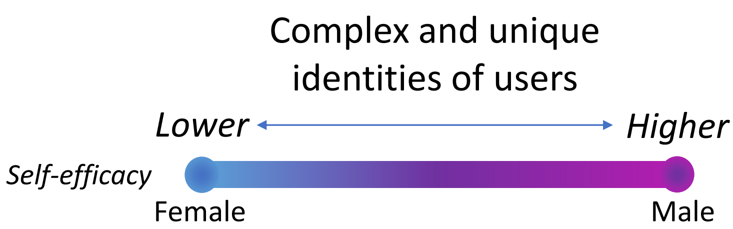
	Characteristic	Description
Perceptions	Risk	Perception of possible outcomes when using a software
	Financial Cost	Perception regarding the financial cost of a software
	Ease of Use	Perception that using the technology will be effort-free
	Usefulness	Perception that using the technology will provide utility
Beliefs	Credibility	Level of credibility attributed to a software
	Trust	Belief in the reliability and trustworthiness of a software
	Privacy	Concerns and behaviour for privacy when using a software
	Cost-benefit	Earned benefit compared to the cost of trying a software
	Self-efficacy	Belief in the ability to use software in varied situations
	Sense of Belonging	Feeling of fitting in with an online culture or community
Preferences	Linguistic & Comm. Style	Linguistic and communication styles in an online community / website interface / software
	Visual Design	Aesthetics of the software interface (e.g., imagery, colorfulness, complexity, fonts)
Skills	Cue Detection	Cue detection in interface design, language, and community norms of a software
	Information Processing Style	Strategies for processing new information and solving problems in a software task
	Awareness	Previous experiences and knowledge about a software
Motivations	Willingness to Learn	Desire to acquire knowledge about a software
	Motivation	Reasons behind one's behaviors towards software
	Tinkering	Exploratory behavior when using a software
Responsibilities	Time Commitment	Time one has available for using a software
	Routine Integration	Software compatibility with one's habits, behavior, patterns, and environments
Environment	Social Interaction	Type and quantity of social interaction provided by an online community/software
	Environment	Conditions for interacting with the software
Access	Access	Access to technological resources

[References given in final slide]*

Human Actor’s Characteristics

- Each characteristic is a range with two ends: one represents the observed statistically in female users and the other in male users. In between, there is a range of variability and diversity

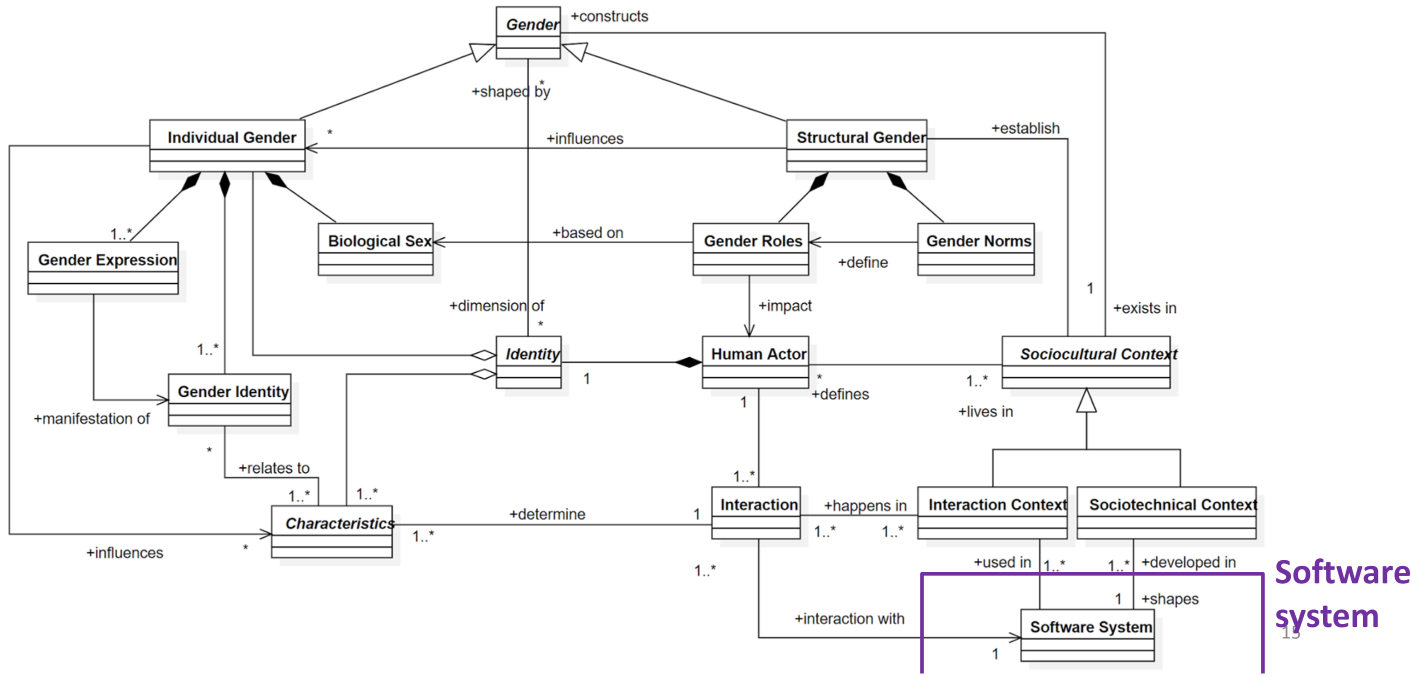
- Example for self-efficacy belief: female users are statistically more likely to have lower self-efficacy while male users are more likely to have it higher [1]



[1] M. Burnett, GenderMag: A method for evaluating software's gender inclusiveness,Interacting with Computers 28(6), 760-787, 2016

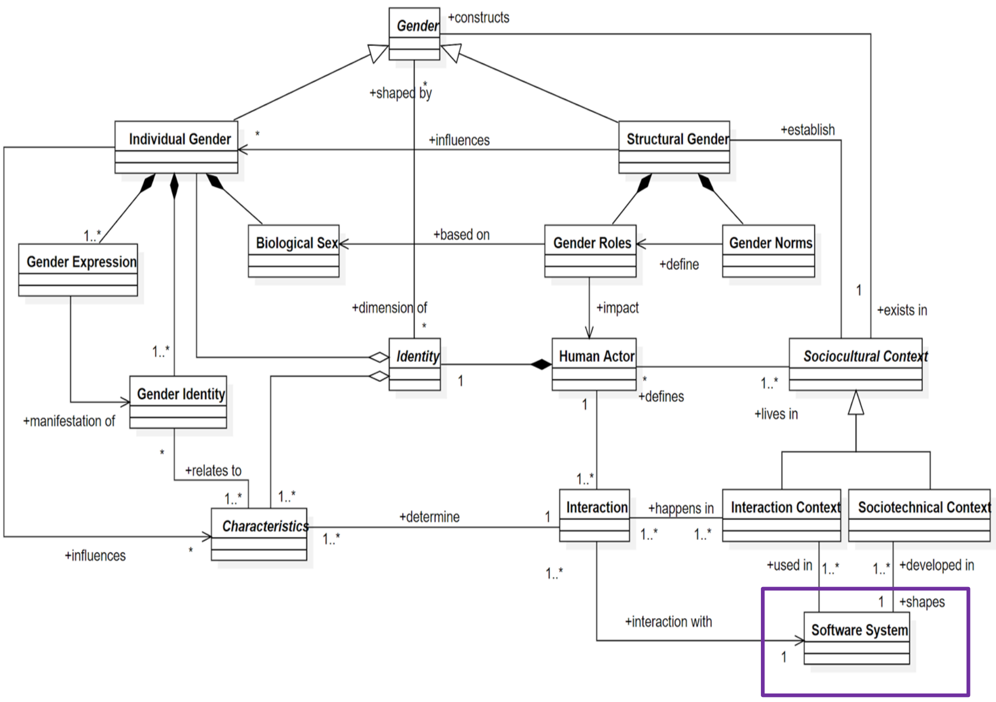
The Software System component model

- Expresses the software system as the result of human-based decisions in a particular sociocultural context

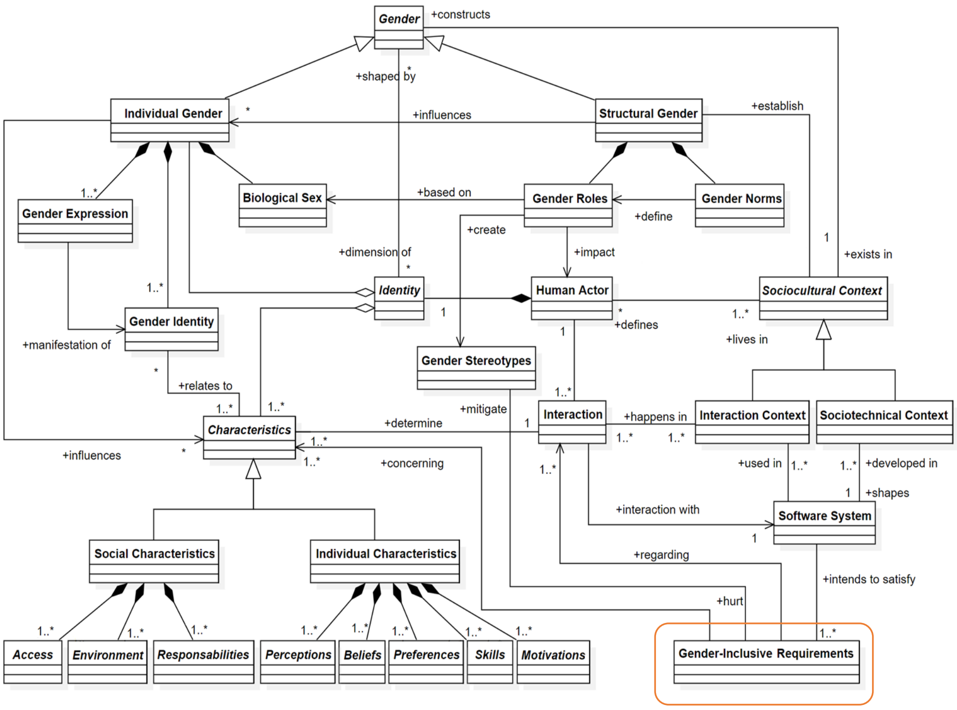


Software system component concepts defined

Software System: the software product developed in a sociotechnical context and used in an interaction context, intended to be Gender-Inclusive



The conceptual model of gender-inclusive requirements



Conceptual Model of Gender-Inclusive Requirements

- Gender-Inclusive Requirements capture the needs and perspectives of diverse users by exploring their range of potential attitudes, integrating them into the requirements of the system

The conceptual model of gender-inclusive requirements

- The formulation of Gender-Inclusive Requirements should satisfy the following criteria:
  - Address historical patterns of gender discrimination
  - Identify and mitigate gender stereotypes
  - Incorporate multiple perspectives, preferences, and needs
  - Account for behavioral diversity within groups of different genders

Human Actor characteristics References

1. *Burnett, M., et al.: GenderMag: A method for evaluating software's gender inclusiveness,Interacting with Computers 28(6), 760-787 (2016)*
2. *Liu, D., Guo, X.: Exploring gender differences in acceptance of mobile computing devices among college students. ISeB15(1), 197–223 (2017)*
3. *Katerina, T., Nicolaos, P.: Examining gender issues in perception and acceptance in web-based end-user development activities. Edu&IT23(3), 1175–1202 (2018)*
4. *Oyibo, K., et al.: Gender difference in the credibility perception of mobile websites:a mixed method approach. In: UMAP. pp. 75–84 (2016)*
5. *Rowan, M., Dehlinger, J.: Observed gender differences in privacy concerns and behaviors of mobile device end users. Procedia CS37, 340–347 (2014)*
6. *Tzafilkou, K., et al.: Gender-based behavioral analysis for end-user development and the 'rules' attributes. E&IT22(4), 1853–1894 (2017)*
7. *Metaxa-Kakavouli, D., et al.: Gender-inclusive design: Sense of belonging and biasin web interfaces. In: HCI. pp. 1–6 (2018)*
8. *Beckwith, L., et al.: Effectiveness of end-user debugging software features: Are there gender issues? In: SIGCHI Conf. HCI. pp. 869–878 (2005)*
9. *Ford, D., et al.: Paradise unplugged: Identifying barriers for female participation on stack overflow. In: ACM SIGSOFT Intl. Symposium on FSE. pp. 846–857 (2016)*
10. *Gao, G., et al.: Gendered design bias: Gender differences of in-game character choice and playing style in league of legends. In: CHI. pp. 307–317 (2017)*
11. *Lenz, B., et al.: Gender issues in the digitalized 'smart' mobility world–conceptualization and empirical findings applying a mixed methods approach. In:HCI. pp. 378–392. Springer (2019)*
12. *Beckwith, L., Burnett, M.: Gender: An important factor in end-user programming environments? In: VL/HCC. pp. 107–114. IEEE (2004)*

Thank you.

This illustrative guide was produced in the context of evaluating the Conceptual Model of Gender-Inclusive Requirements, a work developed during the Master Thesis in Computer Science and Informatics of student Inês Nunes at NOVA School of Science and Technology.

You can find the evaluation survey in this link.

If you have any further questions, please contact ir.nunes@campus.fct.unl.pt

2021 A CONCEPTUAL MODEL FOR GENDER-INCLUSIVE REQUIREMENTS Inês Nunes

NOVA SCHOOL OF