



DÉBORA PATRÍCIA MIGUEL DIAS

Master in Computer Science and Engineering

GENDER-INCLUSIVE INTERFACES

THE CASE OF SOFTWARE DEVELOPMENT TOOLS

MASTER IN COMPUTER SCIENCE AND ENGINEERING
SPECIALIZATION IN SOFTWARE ENGINEERING

NOVA University Lisbon
September, 2024



DEPARTMENT OF
COMPUTER SCIENCE

GENDER-INCLUSIVE INTERFACES

THE CASE OF SOFTWARE DEVELOPMENT TOOLS

DÉBORA PATRÍCIA MIGUEL DIAS

Master in Computer Science and Engineering

Adviser: Miguel Goulão

Associate Professor, NOVA University Lisbon

MASTER IN COMPUTER SCIENCE AND ENGINEERING
SPECIALIZATION IN SOFTWARE ENGINEERING

NOVA University Lisbon

September, 2024

Gender-Inclusive Interfaces

The case of Software Development Tools

Copyright © Débora Patrícia Miguel Dias, NOVA School of Science and Technology, NOVA University Lisbon.

The NOVA School of Science and Technology and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.

ABSTRACT

The underrepresentation of women in the Information Technology field, where female participation is below 30% and even lower (3-5%) in Open-source software (OSS) communities, significantly impacts both diversity in software development and the gender inclusiveness of software tools. This disparity underscores the need for interfaces that cater to a broader range of users. The GenderMag method offers a systematic approach to identifying and addressing gender biases in software tools.

This study used the GenderMag method to evaluate and improve the inclusiveness of the piStar tool by comparing its original version with a modified version aimed at enhancing usability. A controlled experiment with 24 participants, divided into two groups balanced by GenderMag personas, tested either the original or modified tool. Key variables measured included task completion time, usability, cognitive load, and accuracy, with qualitative feedback gathered to explore user experiences in depth.

The results revealed no statistically significant differences between the two versions. However, qualitative feedback showed that the multi-selection feature in Version 2 improved user experience.

The study's limitations include a small and homogeneous sample, with participants sharing similar academic backgrounds, potentially introducing bias and limiting the generalizability of the results. Additionally, the tasks were relatively simple, which may have underestimated the tool's usability challenges in more complex, real-world scenarios.

In conclusion, while statistical analysis did not show clear superiority of Version 2, qualitative insights suggest that modifications, especially the multi-selection feature, improved usability. This study underscores the significance of gender inclusiveness in software design, demonstrating how systematic methods like GenderMag can effectively identify and rectify gender biases. By focusing on the developer experience, the research contributes to creating more inclusive software development tools, thereby promoting diversity and equality in the technology field.

Keywords: Developer experience, Gender inclusion, User interface, GenderMag, Gender-Mag Design Catalog

RESUMO

A sub-representação das mulheres no campo da Tecnologia da Informação (TI), onde a participação feminina é inferior a 30%, e ainda menor (3-5%) nas comunidades OSS, impacta significativamente tanto a diversidade no desenvolvimento de software quanto a inclusão de gênero nas ferramentas de software. Esta disparidade ressalta a necessidade de interfaces que atendam a uma gama mais ampla de utilizadores. O método GenderMag oferece uma abordagem sistemática para identificar e corrigir *bugs* de inclusão de gênero em ferramentas de software.

Este estudo aplicou o método GenderMag para avaliar e melhorar a inclusão da ferramenta piStar, comparando a versão original com uma versão modificada para aumentar a usabilidade. Foram testadas 24 pessoas, divididas em dois grupos equilibrados pelas personas do GenderMag. As principais variáveis foram o tempo de conclusão, percepção de usabilidade, carga cognitiva e níveis de correção, com feedback qualitativo recolhido para aprofundar a análise da experiência dos utilizadores.

Os resultados não revelaram diferenças estatisticamente significativas entre as duas versões. No entanto, o *feedback* qualitativo mostrou que a multi-seleção na Versão 2 melhorou a experiência do utilizador.

As limitações do estudo incluem uma amostra pequena e homogênea, com participantes com formações académicas semelhantes, o que pode ter limitado a generalização dos resultados. Além disso, as tarefas eram relativamente simples, o que pode ter subestimado os desafios de usabilidade em cenários mais complexos.

Em conclusão, embora a análise estatística não tenha mostrado clara superioridade da Versão 2, a avaliação qualitativa sugere que as modificações, especialmente a multi-seleção, melhoraram a usabilidade. Este estudo destaca a importância da inclusão de gênero no *design* de software, demonstrando como métodos como o GenderMag mitigam os *bugs* de inclusão de gênero. O foco na experiência do desenvolvedor contribui para software mais inclusivo, promovendo a diversidade e a igualdade no campo da tecnologia.

Palavras-chave: Experiência do desenvolvedor, Inclusão de gênero, Interface do utilizador, GenderMag, Catálogo de Desenho GenderMag

CONTENTS

List of Figures	vii
List of Tables	x
Glossary	xi
Acronyms	xiii
1 Introduction	1
1.1 Description and context	1
1.2 Motivation	1
1.3 Goals and expected results	2
1.4 Structure of the document	2
2 State of the Art	4
2.1 Theoretical foundation	4
2.1.1 The GenderMag method	4
2.1.2 Software Development Tools	13
2.1.3 Other usability practices and techniques	15
2.2 Relevance of the question	17
2.3 Focus	18
3 Background	20
3.1 Categories of design changes	20
3.1.1 Ambiguous option	20
3.1.2 Big wall of text	21
3.1.3 Difficult first task	22
3.1.4 Incomplete instructions	22
3.1.5 Login wall	23
3.1.6 Long list	24
3.1.7 Many options to choose from	24

3.1.8	Multi-step task	25
3.1.9	Paragraph of instructions	26
3.1.10	Reason to follow link unclear	26
3.1.11	Unexplained information	28
3.1.12	Help users collect, compare, and integrate information	28
3.2	An example of an empirically validated design change	29
3.2.1	Example of Unexplained information	29
4	Inclusivity Bugs' Detection and Implementation	32
4.1	Inclusivity Bugs' Detection	32
4.1.1	Application of the GenderMag Method to piStar	32
4.1.2	GenderMag application: sessions with João Pimentel	64
4.2	Implementation	66
4.2.1	Deletion of an item	67
4.2.2	Confirming to a seemingly permanent alteration	69
4.2.3	Editing fields not obvious	71
4.2.4	Multi-Selection for Multi-delete and Report Generation Feature	72
5	Testing and Result Analysis	75
5.1	Experiment Planning	75
5.1.1	Goals	75
5.1.2	Experiment Units	75
5.1.3	Experiment Material	76
5.1.4	Tasks	76
5.1.5	Hypotheses, parameters and variables	78
5.1.6	Design	79
5.1.7	Procedure	79
5.1.8	Analysis Procedure	79
5.2	Execution	80
5.2.1	Preparation	80
5.2.2	Deviations	80
5.3	Analysis	81
5.3.1	Descriptive Statistics	82
5.4	Discussion	88
5.4.1	Evaluation of Results and Implications	88
5.4.2	Threats to Validity	90
5.4.3	Inferences	92
5.4.4	Lessons Learned	92
6	Conclusions	95
6.1	Summary	95
6.2	Impact	96

6.3 Future Work	96
Bibliography	97
Appendices	
A Abi (Abigail/Abishek) Persona Foundations	103
B Pat (Patricia/Patrick) Persona Foundations	104
C Tim (Timothy/Timara) Persona Foundations	105
Annexes	
I Annex GenderMag Survey in Portuguese	106
II Annex GenderMag Survey in English	107
III Annex Test Guide in Portuguese	108
IV Annex Test Guide in English	109
V Annex Presentation to Test Participants	110

LIST OF FIGURES

2.1	Graph representing a MSc Computer Science and Engineering student's facet values analyzed through the GenderMag's survey [29].	8
2.2	Model representation of the GenderMag process [24].	10
2.3	Model representation of the InclusiveMag process [38].	12
2.4	Screenshot of the piStar interface [46]. On the left the menu item "Add" is selected and on the right the menu item "File" is selected.	13
2.5	i* 2.0 elements, adapted from [1].	15
2.6	Illustrative image for the developer experience concept [21].	19
3.1	Interfaces representing the before and after the "ambiguous option" design change, respectively, regarding scenario one.	20
3.2	Interfaces representing the before and after the "ambiguous option" design change, respectively, in regards to scenario one.	21
3.3	Interfaces representing the before and after the "big wall of text" design change, respectively.	21
3.4	Interfaces representing the before and after the "difficult first task" design change, respectively.	22
3.5	Interfaces representing the before and after the "incomplete instructions" design change, respectively.	23
3.6	Interfaces representing the before and after the "login wall" design change, respectively.	23
3.7	Interfaces representing the "long list" design change, respectively.	24
3.8	Interfaces representing the before and after the "many options to choose from" design change, respectively.	24
3.9	Interfaces representing the before and after the "many options to choose from" design change, respectively.	25
3.10	Interfaces representing the before and after the "multi-step task" design change, respectively.	25
3.11	Interfaces representing the before and after the "paragraph of instructions" design change, respectively.	26

3.12	Interfaces representing the before and after the “reason to follow link unclear” design change, respectively, regarding scenario one.	27
3.13	Interfaces representing the before and after the “reason to follow link unclear” design change, respectively, regarding scenario two.	27
3.14	Interfaces representing the before and after the “reason to follow link unclear” design change, respectively, in regards to scenario three.	28
3.15	Interfaces representing the before and after the “help users collect, compare, and integrate information” design change, respectively.	29
3.16	Screenshot of the original table. It seems contradictory since the "Rank" filter is selected but the numbers on the right side do not seem to be in order. However, the numbers on the right are actually the number of publications for each university and doesn't affect the rank. [23]	30
3.17	Screenshot of the new version [23]	31
4.1	Abi persona card with custom information.	33
4.2	Tim persona card with custom information.	34
4.3	Illustration depicting the envisioned model to attain through the goals and subgoals [27].	34
4.4	Illustration depicting the envisioned model at the time of creating the AND relationship between the children “Collect available dates” and “Merge available dates” to the parent “Meeting Participant”.	47
4.5	Illustration depicting the envisioned model at the time of creating the OR relationship between the children “Manually merge dates” and “Merge dates using scheduler” to their parent “Merge available dates”.	50
4.6	Illustration depicting the envisioned model at the time of creating the needed-by relationship between the children “Available dates” to its parent “Collect available dates”.	52
4.7	Illustration depicting the envisioned model at the time of creating the contribution links between “Merge dates using scheduler” and “Manually merge dates” with the quality “Minimal effort”.	56
4.8	Illustration depicting the envisioned model at the time of creating the right side of the model.	57
4.9	Illustration of the model to be uploaded into the tool to test the process of comparing more than two item's attributes.	59
4.10	Snippet of the text file saved when saving a model on the piStar tool [46]. . .	66
4.11	Screenshot of the piStar [46] interface, with a highlight on the “Delete” button before the implementation of a fix for the potential inclusivity bug. On the right, we can a zoomed-in version.	67
4.12	Screenshot of the piStar [46] interface, with a highlight on the “Delete” button after the implementation of a fix for the potential inclusivity bug. On the right, we can a zoomed-in version.	68

4.13	Screenshot of the piStar [46] interface, with a highlight on the alert message for deleting an actor. On the left, we can see the alert message before the implementation of a fix for the potential inclusivity bug. On the right, we can see after the implementation of the fix.	69
4.14	Screenshot of the piStar [46] interface, with a highlight on the alert message for creating a new model. On the left, we can see the alert message before the implementation of a fix for the potential inclusivity bug. On the right, we can see after the implementation of the fix.	70
4.15	Screenshot of the piStar [46] interface, with a highlight on the alert message for loading example models. On the left, we can see the alert message before the implementation of a fix for the potential inclusivity bug. On the right, we can see after the implementation of the fix.	70
4.16	Screenshot of the piStar [46] interface, with a highlight on the name editing field. On the right, we can see the underlined field when hovered over. . . .	71
4.17	Illustration depicting the piStar [46] interface for the properties editing design changes.	72
4.18	Illustration depicting the envisioned piStar [46] interface for multi-selection feature.	73
4.19	Illustration depicting the envisioned piStar [46] interface for report feature.	73
4.20	Illustration depicting the envisioned piStar [46] interface for general model report feature.	74
4.21	Illustration depicting the envisioned piStar [46] interface for general model report feature.	74
5.1	Histograms and a box plot to represent the distribution of the self-efficacy facet values through versions.	83
5.2	Histograms and a box plot to represent the distribution of the motivations facet values through versions.	83
5.3	Histograms and a box plot to represent the distribution of the learning style facet values through versions.	84
5.4	Histograms and a box plot to represent the distribution of the information processing style facet values through versions.	84
5.5	Histograms and a box plot to represent the distribution of the risk aversion facet values through versions.	84

LIST OF TABLES

5.1	Distribution of participants' ages and experience with iStar (i*)	82
5.2	Descriptive Statistics for GenderMag values	83
5.3	Independent Samples Test of the distribution of the GenderMag values of participants in version 1 and 2.	83
5.4	Descriptive Statistics for dependent variables with Shapiro-Wilk Significance	85
5.5	Descriptive Statistics for independent variables with Shapiro-Wilk Significance	85
5.6	Independent Samples T-Test	85
5.7	Kendall's Tau-b and Spearman's Rho Correlations for all samples	86
5.8	Kendall's Tau-b and Spearman's Rho Correlations for samples from version 1	86
5.9	Kendall's Tau-b and Spearman's Rho Correlations for samples from version 2	87

GLOSSARY

A/B testing	Strategy for testing two different versions of a website or application to determine which one performs better. (p. 79)
cognitive walkthrough	Analytical evaluation method that focuses on ease of learning. ([7], [47] and [36] (pp. 5, 9, 10, 16, 33, 34)
developer	The term “developer” is used to refer to anyone who is engaged in the activity of developing software. [21] (p. 17)
developer experience	Developer experience is employed to refer to the concept that captures how developers think and feel about their activities within their working environments. This idea is heavily influenced by the concept of "user experience". [21] (pp. vii, 1, 2, 18, 19, 65)
experience	The term "experience" encompasses both the immediate perception of events and the recollection of those events, along with the knowledge acquired through interpretation and reflection on remembered experiences. [21] (p. 18)
female	The term "females" is employed to refer to individuals who identify as female. [25] (p. 4)
gender	Following articles and papers in the field, the term "gender" is used to refer to a socially constructed concept ([12] and [57] cited by [25]). This concept encompasses the identification, expression, and enactment of gender, which may or may not correspond to one’s biological sex. (pp. 2, 4, 18)

GenderMag	The GenderMag method is a systematic approach aimed at identifying and addressing gender biases in software, thereby fostering more inclusive and equitable technology solutions. [25] (pp. 1, 4, 5, 16)
male	The term "males" is employed to refer to individuals who identify as male. [25] (p. 4)
tenure	Amount of experience accumulated by each group member. [54] (pp. 2, 18)
user	The term "user" is used to refer to anyone who is engaged in the activity of using software. (pp. 5, 17)
user experience	User experience covers the resulting perceptions and responses of a person's use or anticipated use of a product, system or service. This includes perceived product properties such as value, desirability, and usefulness. [21] (p. 2)

ACRONYMS

HCI	Human-Computer Interaction (<i>p. 17</i>)
i*	iStar (<i>pp. x, 13, 14, 32, 76, 79, 82</i>)
IDE	Integrated Development Environment (<i>p. 13</i>)
NASA TLX	NASA Task Load Index (<i>pp. 78, 79, 81, 87, 89, 95</i>)
OS	Operating System (<i>p. 22</i>)
OSS	Open-source software (<i>pp. ii, iii, 1, 22</i>)
SD	Strategic Dependency (<i>pp. 14, 80</i>)
SR	Strategic Rationale (<i>pp. 14, 43–45, 47, 48, 50–52, 57, 80</i>)
SUS	System Usability Scale (<i>pp. 78, 79, 81, 87, 89, 95</i>)
UCD	User-Centered Design (<i>pp. 15, 16, 96</i>)
UX	User Experience (<i>p. 7</i>)

INTRODUCTION

1.1 Description and context

The evolution of software interfaces and development tools has motivated studies around technology use and interaction. In this context, the design and usability of software development tools, play a pivotal role in defining developer experience. This study delves into the realm of gender-inclusive interfaces, with a focus on evaluating and enhancing the inclusiveness of software development tools using the GenderMag method. The GenderMag method is a systematic approach aimed at identifying and addressing gender biases in software, thereby fostering more inclusive and equitable technology solutions [25].

1.2 Motivation

The motivation for this research is underlined by the crucial role of inclusivity in software design, emphasizing the need to close the existing gender gap. Specifically in the Information Technology field, women's participation is less than thirty percent, and three to five percent in OSS communities [51]. This alone leaves women lacking representation in software development tools and, therefore, more exposed to bias issues in software tools. Gender inclusivity not only broadens the developer experience to a wider audience but also fosters equality and mitigates inherent biases within technology.

Specifically in OSS, evident gender biases impact contribution acceptance rates, with women's contributions more likely to be rejected when their gender is known, despite being accepted more frequently than man's when gender is anonymized [51]. This discrepancy highlights a broader issue of underrepresentation of women in software development. The predominance of software solutions designed to suit male-preferred problem-solving styles [25] further exacerbates this gap. This study aims to address and bridge this divide by applying the GenderMag method to systematically identify and correct gender inclusiveness issues in software development tools, supplemented by controlled experiments for thorough evaluation and validation of the implemented

modifications.

This study is driven by the need to bridge this gap to allow the embrace of gender diversity within software development teams. Diversity significantly reduces the likelihood of gender bias, as team members are more attuned to challenges they personally encounter. A diverse team, encompassing a range of genders and experiences, is adept at identifying and addressing a variety of issues, enhancing problem-solving capabilities. Moreover, gender and tenure diversity is linked to heightened productivity [54]. Research supports that gender-inclusive software not only benefits those directly affected by biases but also improves the overall user experience, underscoring the universal value of inclusivity in technology design [25, 28, 55].

1.3 Goals and expected results

The primary goal of this research is to apply the GenderMag method to assess and improve the gender inclusiveness of a software development tool, thereby contributing to a more equitable and productive developer experience. The GenderMag method was developed with the concept of user experience in mind, but in this case the method will be leveraged around developer experience. By conducting a thorough analysis using the GenderMag framework, this study aims to identify key gender inclusiveness issues within the piStar tool [46] and propose actionable solutions to address them.

The expected results include a report of the identified issues, proposed design changes validated through empirical research, and an evaluation of the impact of these changes on user experience and inclusivity.

1.4 Structure of the document

It is also relevant to discuss the structure of this document. Apart from this chapter, the document is organized as follows:

- **Chapter 2 - State of the Art:** This chapter lays the foundational groundwork for the study, elaborating on theoretical principles and defining crucial concepts and elements critical to the discourse on gender-inclusive interfaces. It explores existing methodologies, including the GenderMag method, to establish a comprehensive backdrop for the research.
- **Chapter 3 - Related Work:** This chapter situates the current study within the broader research landscape, presenting an overview of relevant works and contributions in the field. It delves into the GenderMag Design Catalog, examining case studies where the GenderMag method and the Design Catalog have been applied, showcasing their outcomes and implications for software inclusivity.

- **Chapter 4 - Inclusivity Bugs' Detection and Implementation:** This chapter discusses the detection of gender inclusivity issues in the piStar tool using the GenderMag method. It documents the process of identifying potential inclusivity bugs, focusing on how gender biases within the tool's interface could hinder inclusivity. The second half of the chapter provides an explanation of the implementation process, describing the code changes made to fix the detected inclusivity issues.
- **Chapter 5 - Testing and Result Analysis:** This chapter presents the results of the user tests conducted after implementing the inclusivity fixes. It covers the testing methodology and provides an analysis of the results. The focus is on determining whether the inclusivity improvements effectively addressed the identified bugs, along with user feedback on the enhanced interface's overall inclusivity and usability.
- **Chapter 6 - Conclusion:** This chapter is divided into three sections. The Summary section recaps the key findings from the research, focusing on the inclusivity bug detection process, the implementation of fixes, and the effectiveness of using the GenderMag method in the piStar tool. The Impact section evaluates the broader implications of this research on gender-inclusive software design, discussing how the improvements contribute to making software tools more accessible and equitable for diverse users. Finally, the Future Work section explores potential opportunities for expanding this work.

STATE OF THE ART

This chapter is particularly important in this field of study, since it allows for a better understanding of the overall context involving gender-inclusive interfaces and the GenderMag method as well as other methods and techniques proposed and studied before.

It is of great importance that prior research, papers and methods are studied and understood so that a solid foundation is created to work and build upon. To do this, it makes sense that this chapter is divided into a few sections. First, we will begin by laying the theoretical groundwork to ensure a basic understanding of the study's core concepts. Next, we will explore why the question at hand is significant, helping to grasp its importance. Finally, we'll conclude with a section that outlines the main focus of the study.

2.1 Theoretical foundation

Individual differences regarding how people problem-solve, tend to cluster by gender [25]. This has great implications, especially for software tools that aim to support diverse teams or individuals with diverse characteristics with problem-solving activities. Most software tools mainly support problem-solving styles and processes favored statistically by males, who tend to be better represented in software development teams, leaving females along with their problem-solving styles unattended to [25]. As alluded to in Chapter 1, Section 1.2, the involvement of women in the Information Technology sector is considerably lower than that of men [51], leading to projects suffering from a lack of diversity of perspectives [52], which results in the underrepresentation of women's needs and preferences.

This matter has been becoming increasingly more popular in recent studies, suggesting that there is in fact a need for one or more software practices or methods that include a gender perspective [60].

2.1.1 The GenderMag method

The GenderMag method offers an answer to the question: "How can ordinary practitioners, with no background in gender research, identify which aspects of their software have

gender-inclusiveness issues, realize why those issues are issues, and thereby know what they should change” [25].

The GenderMag method is an evaluating method with which software practitioners can evaluate the software they are producing or have produced.

The GenderMag method relies heavily on the following five facets to characterize people’s preferred problem-solving styles: motivation, information-processing style, attitude towards risk, computer self-efficacy and learning style [25]. The method brings these facets to life by assigning them into personas that are relatable, and embeds them in a systematic process through a gender specialization of the cognitive walkthrough, which we will discuss in the next few sections.

It is important to note that although the GenderMag is tied to the concept of gender, this concept is not the core of the method. The method can be used to detect and fix inclusiveness bugs and issues without ever mentioning gender, since the facet values are the most important resource to achieve a more inclusive software interface, not gender.

The GenderMag method, rather than focusing on gender, focuses on individual differences within genders, embracing the notion of pluralism as an alternative to universality, allowing the method to support beyond binary gender, extending its support to non-binary gender identifications [55].

2.1.1.1 The GenderMag facets

As mentioned before, at the core of GenderMag are the five facets of gender differences. These facets allow for a characterization of people’s differences at a foundational level and hold solid value in multiple fields of study, such as in psychology, education and communications [25]. The GenderMag method then encapsulates the five facets into personas.

The five facets are [25]:

- **Motivation:** this facet represents the motivation that led the user to procure the problem-solving tool. Statistically, most females tend to be motivated to use technology for what it enables them to achieve [25, 37]. In contrast, most males are motivated by the joy of using technology itself [25, 37]. Logically, this difference affects the choice-making process that is included in problem-solving software. People who find joy in using new tools and learning their functionalities will face this activity differently from people who are motivated to use new tools for what their functionalities enable them to achieve.
- **Information processing styles:** when dealing with solving a problem, the idea of processing new information is inherent. Females are statistically more likely to use comprehensive information processing styles, starting by gathering fairly complete information before proceeding. On the other hand, males are statistically inclined to

use selective styles, starting by following the first promising information, and then potentially backtracking, in “depth first” order ([39] cited by [40]).

- **Computer self-efficacy:** a person’s judgment of their capability to organize and execute actions to get to a certain desired performance [3]. In other words, self-efficacy is a person’s confidence about succeeding given a specific task. Self-efficacy holds significance in the realm of problem-solving, as an individual’s self-efficacy directly shapes their use of cognitive strategies, the extent of effort they invest, their perseverance, and their approaches to overcoming obstacles [3, 64]. Computer self-efficacy is a person’s confidence about succeeding given a specific task in a computer. Females tend to statistically have lower computer self-efficacy than males [4], and it is heavily tied to the confidence that others expressed in the female’s capabilities [64].
- **Risk aversion:** Females are statistically more prone to be risk-averse than males [19]. The variation in facet values carries significant weight in influencing females’ choices regarding which sets of features to utilize and which ones to steer clear of.
- **Tinkering:** Females are statistically inclined to playfully experiment less with features new to them, when compared to males [5]. Nonetheless, research also indicates that when females engage in tinkering, they do so in a more productive way than males and they tend to exhibit a higher propensity for reflective thinking throughout the process, leading to occasional greater benefits than their male counterparts [5]. Male’s tinkering tends to be more counterproductive [5]. This might be tied to the fact that females tend to pause while tinkering more than males, and pausing allows for reflection [5]. The divergence in tinkering behaviors between females and males yields an effect on the selection of software features they choose to use, particularly when a fundamental design aspect of the software involves users learning new features through exploration and tinkering.

2.1.1.2 The GenderMag personas

Personas are fictional, detailed profiles of individuals, encompassing a range of attributes akin to those of real people [53]. These profiles typically include names, gender, age, and professions, as well as a broader spectrum of characteristics such as friendships, ethnic background, personal preferences, educational achievements, and various indicators of socioeconomic status. These composite descriptions are crafted to represent a diverse array of human traits and experiences. The GenderMag personas are the personification of the facet values. These facet values match some of those encountered on gender theory.

Considering the GenderMag method’s usability, it was decided that although the literature on gender theory states there are more than five facets, the method would settle on five, to lessen the mental load on the method’s users. In determining the five facets to include, the criteria demanded that they’re well-researched, usable by non-gender experts

in software development or User Experience (UX), and impactful on software usage [25]. As expected, this decision does have an impact on the method's facet range and creates limitations. The GenderMag method tries to ameliorate this by making the personas somewhat customizable, which we will discuss in this section.

For each facet, the facet values were considered, and each of the personas in the persona set has a value for each of the five facets, so as to ensure that the personas cover all the facet values.

The persona set is composed of the following personas, ranging from the persona with the most amount of facet values frequently seen in males, to the persona with the most amount of facet values frequently seen in females: Tim (Timothy/Tamara), Pat (Patrick/Patricia) and Abi (Abigail/Abishek).

The persona's name in itself also has room for gender. Any persona can be attached to any gender, further highlighting the approach's commitment to considering diverse user needs and perspectives, ensuring that systems are accessible and usable by people of all genders, which is the focus of the GenderMag method.

The identical personas, both Pats, Pat(trick) and Pat(ricia), are not only middle-ground personas but also a way to battle against inappropriate stereotypes. They highlight the duality within the same person's problem-solving style, while also showcasing that the true concept at the core of GenderMag are actually the facet values and not gender. To reiterate, these two personas accentuate the idea that, although individual differences often cluster by gender, as mentioned before, the most effective way to ensure gender inclusiveness is for the evaluators to ask themselves: "is this feature effective for people who have a comprehensive information processing style?" [25], not allowing the gender label to create inappropriate stereotypes or even to limit their understanding of the GenderMag essence.

The set of personas can also be seen as a set of four personas [25]: Tim (Timothy/Tamara), Pat(trick), Pat(ricia) and Abi (Abigail/Abishek). Opting to present four personas instead of three underscores the significance of the Pat persona. Within this quartet, the presence of both Patrick and Patricia as the two Pats underscores that these personas are identical in all aspects except for their names and photographs, thereby establishing that the personas are designed to reflect facet values rather than gender.

The personas emphasize individual distinctions, encompassing facet values documented in the literature [25]. For illustrative purposes, consider the graph in **Figure 2.1** representing the facet values of a MSc Computer Science and Engineering student at Nova School of Science and Technology. The graph in **Figure 2.1** intent is to further emphasize that while pure Abis and Tims do exist, it is common for users to share facet values with all the GenderMag's personas, highlighting how gender, if not for the fact that people's problem-solving styles tend to cluster by gender, would not be relevant when choosing a persona. Being closer to Abi or closer to Tim does not present an advantage in itself, it is only an advantage when the user identifies with a persona whose needs were taken into consideration when building interfaces. This puts all the weight on how the GenderMag method aims to support diverse problem-solving styles, regardless of gender.

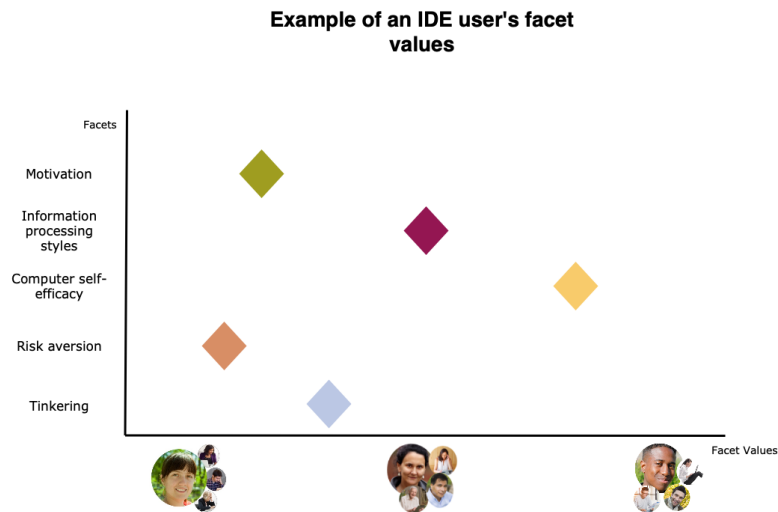


Figure 2.1: Graph representing a MSc Computer Science and Engineering student's facet values analyzed through the GenderMag's survey [29].

To better understand and further explore the notion of the defined personas, **Appendixes A, B and C** were added. In **Appendixes A, B and C** it is shown the background knowledge and skills, motivations and strategies and attitude to technology for the Abi persona, the Pat persona and the Tim persona, respectively. In gray is the persona's characteristics that can not easily be changed, but this will be further explored in this section. In black is the text about the persona that is fully customizable.

To an extent, personas must be customizable. In order for the software development team using the GenderMag method to connect and empathize with the personas, some of the personas portions may have to be customized. For example, a product aimed at professional chefs may need a professional chef instead of an accountant representing the user [25].

It is important that the personas, when customized, do not lose their essence and keep cohesion all throughout the method. All substantial changes made to a persona, in other words, changes made to any part of the gray text in **Appendixes A, B and C**, are changes that alter the overall functioning of the personas and their facet values, and they have to be proportionally adjusted in all other personas [25].

When delving into the realm of personas, it becomes imperative to discuss stereotypes. Given that personas represent a group of users that contain the same facet values [53], there is high risk for inappropriate gender stereotyping.

The GenderMag's team has come up with several different ways to ameliorate this risk. Firstly, removing gender from the equation can, at first, seem like it would solve the problem, but soon the question would surround "users", which in itself leads males to envision males (associating the terms "person" and "participant" as male) [9], which goes against the GenderMag's goal of encouraging developers to consider gender, male and female ([35] cited by [25]). So the GenderMag method focuses on keeping faceted females and males in the center of the method, with the purpose of encouraging empathy towards

the different personas and their facet values.

In addition to placing faceted personas at the core of the evaluation process, three steps were implemented to reduce the risk of inappropriate stereotyping. Firstly, the four personas highlight subtle differences between genders, mirrored in their complex facet values. Secondly, the two Pats share identical facet values but differ in gender, underscoring that inclusivity concerns exist within the range of possible values for each facet. Lastly, the personas actively challenge several unfounded assumptions, as evidenced by data [15]. This is exemplified in the equal proficiency of the four personas in accounting-level mathematics. Their identical background, job title, responsibilities, math skills, domain knowledge, and proficiency with technology further reinforce this equality. Additionally, research indicates a similar prevalence of computer game enjoyment among both males and females, a shared trait among all four personas.

Adhering to these steps, GenderMag ensures that personas sharing the same job title and responsibilities remain entirely identical in all aspects unless empirically verified distinctions exist. Any differences beyond the five facets must meet three criteria: (1) empirical support, (2) absence of implications regarding intelligence or education differences, and (3) alignment with the respective persona's facet values or skill level.

2.1.1.3 The GenderMag process

The GenderMag method is an evaluating method, based partly on the cognitive walkthrough, as mentioned in Section 2.1.1, with which software practitioners can evaluate the software they are producing or have produced. The GenderMag method is based partly on a gender-specialized cognitive walkthrough. The cognitive walkthrough is an analytical evaluation method and focuses on ease of learning [7], along with the basic principles of usability [36]. However, distinctly from other evaluation methods, cognitive walkthrough also accounts for users' cognitive activities, prioritizing their goals and knowledge when executing a specific sequence of steps and tasks [36]. In a cognitive walkthrough session, the team of evaluators walks through the interface step by step, while keeping in mind the goal of this method: to evaluate the interface's usability and learnability. This team tries to accomplish each task from a list of tasks, step-by-step, in the sequence a user would be likely to use when using the system for the first time [25].

The cognitive walkthrough method exhibits both advantages and disadvantages. Notably, it is characterized by its robustness, which is a significant strength ([47] cited by [25]). Other strengths rely on uncovering issues that would affect not only novices but the interface's overall usability [36]. This method can be used in the earlier phases of the design and development with prototypes to detect errors early on [50] and also in the later phases [58]. In regards to the user, this method can shed light on the prerequisite background knowledge that users need to have in order to successfully complete tasks [58], and it can unveil any assumptions and misconceptions about the user that the designer might have unintentionally incorporated into the system [36].

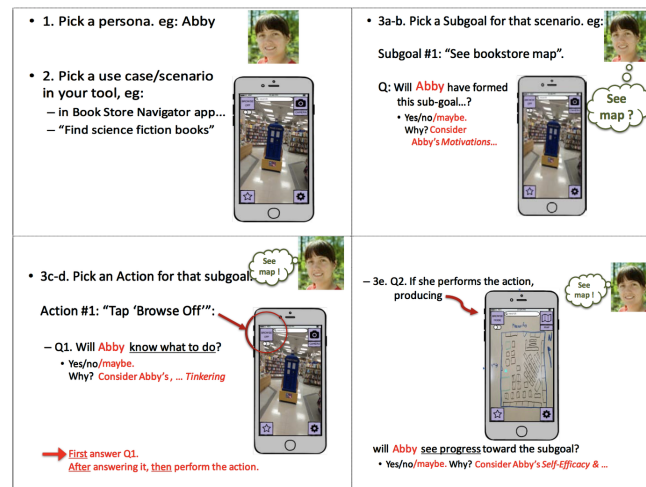


Figure 2.2: Model representation of the GenderMag process [24].

As previously mentioned, the cognitive walkthrough method also has some weaknesses. To start, choices made in task selection impact significantly the issues’ detection, since it is an essential concern for any user task’s methodology [59]. Also, this method can lead to “tedium” since the same questions are asked frequently and it becomes repetitive [36]. Finally, the cognitive walkthrough seems to fall short when it comes to actually making the evaluators feel and relate to the users since the evaluators seem to imagine users all too similar to themselves. This can give rise to two cognitive biases: Anchoring and Stereotyping. Anchoring bias occurs when individuals overly rely on pre-existing or initial information when making decisions. In the context of the cognitive walkthrough, this bias manifests when, despite their best efforts, evaluators unconsciously assess the interface from the viewpoint of users who resemble them, rather than accurately representing the diverse range of actual users. Finally, there is a high risk of describing users in broad or high-level terms, which unavoidably leads to stereotyping, since there is no room for the diversity of the actual group of users of the system being evaluated [30]. This weakness, however, is mitigated by the GenderMag faceted personas.

When it comes to the GenderMag method process to evaluate a particular software product, the team of evaluators selects one or more personas from the persona set and tailors the customizable portions of the persona if deemed as needed. Then, the Preparatory Phase takes place, and this is where the team defines the list of tasks to be completed as well as the ideal sequence of each specific task. After the Preparatory Phase, comes the Analysis Phase where the team evaluates their own software product through a gender-specialized cognitive walkthrough with each persona they have selected [25]. In Figure 2.2, we can see an example of how the process would go in a specific situation. Firstly, a persona would be chosen to carry out the user activities. After this, a use case/scenario needs to be chosen to create the user action context. The next step would be to pick a subgoal for the specific scenario chosen in the previous step, and then an action for this subgoal. After this point, the evaluator would carry out the many

actions in a iterative way, while always asking themselves questions from the persona's perspective.

What the GenderMag method offers is a systematic method that software practitioners can use during or after their problem-solving software conception. It proposes a way to detect and fix inclusiveness bugs early on in the project when it is cheaper to do so. This method allows the community of software developers to encounter GenderMag moments during their developing sessions. GenderMag moments are portions of a full development session where the team discusses in-the-moment design decisions taking into account the facet values. These moments create a safe space to discuss potential solutions and ideas, while reducing the need for full GenderMag sessions, lessening the effort required to incorporate this method into their own existing software development process. Getting teams to relate to and impersonate the personas, is a step in the direction of changing the gender perspective with which software is developed [28].

2.1.1.4 The InclusiveMag

Taking a step back, creating room for a more abstract analysis, one can come to the conclusion that gender-inclusiveness issues in software belong to a big pool of inclusiveness issues that affect all types of diverse populations.

Although many existing tools and resources can be applied to the software development process, such as the inclusion of diverse users in design sessions with designers [8, 41] and conducting user testing to gain valuable insights [45], these approaches can prove to be expensive both in terms of financial costs and time investment. Due to this, the need for processes that do not require user involvement or dependence on experts to implement them is becoming increasingly apparent.

To evaluate software's support for diverse populations, concerning inclusivity issues in general, InclusiveMag was introduced. InclusiveMag (Inclusiveness Magnifier) is a generalization of the GenderMag method, built on its principles and processes [38].

InclusiveMag is a method that aims to allow inclusivity researchers to new design methods that are focused on inclusivity, leaving less room for inclusiveness bugs. These methods are intended to be used by software developers so they can evaluate the software they are building or built, having the clear goal of not only making the software more inclusive to an under-served population but also more usable to all users [38].

The InclusiveMag has three steps, as represented in **Figure 2.3** [38]:

- Step 1: In this step, inclusivity researchers define the scope by choosing a specific software category and diversity dimension, and conduct research to investigate potential factors influencing how different populations within that dimension interact with the chosen software type. The activities that compose this activity are iterative and often intertwined. This step yields a collection of facet categories and facet values that exhibit variations between the under-served group and the mainstream population. This step is demanding in terms of effort, but its quality is of great

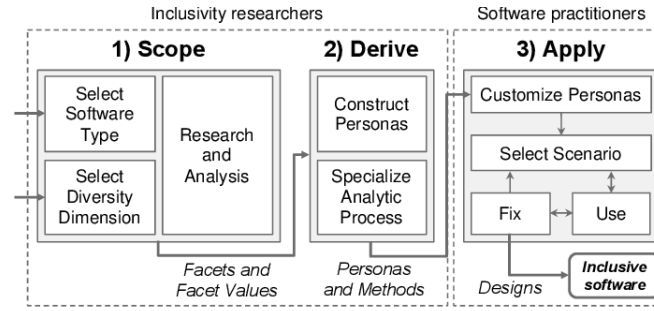


Figure 2.3: Model representation of the InclusiveMag process [38].

importance, as the resulting facets heavily rely on it. The objective is to construct high-quality, well-defined facets where individual differences exhibit distinct patterns within the under-served population compared to the mainstream population, and these facets are pertinent to the chosen software type. The output of this step is a small number of facets, so as to not overwhelm the practitioners with a big set and keep the method feasible.

- Step 2: In this step, inclusivity researchers use the output from step 1 and extract personas and an analytical process specialized to the pertaining diversity dimension. This step begins with "projecting the values of each facet onto a linear scale for that facet" [38]. The outcome of this step is a generated method built on top of the facets selected in Step 1.
- Step 3: This step is performed by software researchers, where they customize the personas to the appropriate background, demographic and skill characteristics for the software in question. Then, the software development team chooses on the personas they customized and for each one they choose a scenario to analyze for their software. This step produces an output of a list of issues to fix. Some will be usability issues and some will be inclusiveness issues. For these last type of issues, inclusive fixes can be oriented by the facets that helped raise the issue in the first place. The effectiveness and success of this generated method are greatly contingent on the quality of the facets, which, in turn, relies on the researcher's capability to acquire or generate a sufficient amount of high-quality evidence for deriving these facets.

InclusiveMag is, as stated, composed of two steps intended to be conducted by industrial (or academic) researchers, and only the last step is supposed to be conducted by the software developers themselves.

The process of validating this method is still ongoing and this method is still very much in its infancy but, we can look at the InclusiveMag method as a generalization of the GenderMag method.

2.1.2 Software Development Tools

The two main software development tools at hand in this study are Integrated Development Environment (IDE) and other multiple types of Software Development Tools.

An IDE is a software application or a suite of tools that provides a comprehensive environment for computer programmers and developers to write, edit, debug, and manage code for software development. They often include features like code editors, debugging tools, version control integration, and project management capabilities to streamline the development process.

IDEs are instances of problem-solving software, containing features in which users are actively engaged in working out a solution to some kind of problem or task [25]

In addition to IDEs, there are also web-based software development tools that do not require installation. These tools also provide features and functionalities that support developers in their problem-solving tasks.

Software development tools are crafted by software developers, making them susceptible to inclusivity issues, specifically those tied to gender. The root cause lies in the fact that these tools are products of human development, allowing for the unintentional integration of gender biases and personal preferences into the software. As mentioned in Section 1.2, the participation of men in the Information Technology field is much greater than women's (women's participation in Information Technology is not even thirty percent), leaving the space for the introduction of personal preferences and choices into software that may not represent the needs and preferences of all genders, and specifically of women.

2.1.2.1 piStar tool and i* 2.0 language and concepts

piStar is an open-source, online goal modeling tool for i* 2.0, with no installation required. This tool allows for developers to build i* models with ease.

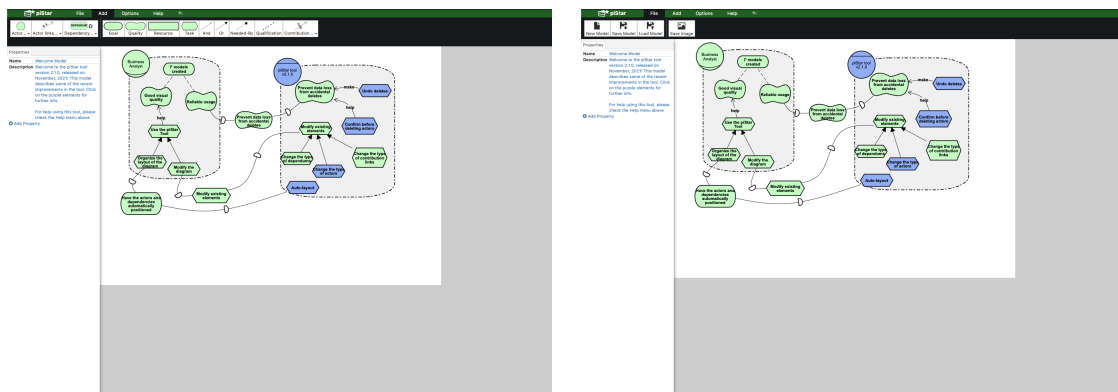


Figure 2.4: Screenshot of the piStar interface [46]. On the left the menu item “Add” is selected and on the right the menu item “File” is selected.

As illustrated in **Figure 2.4**, the interface includes a toolbar at the top of the page, offering basic file management and platform options. Directly beneath the toolbar, there

are elements specific to the tool, enabling the creation of actors, links, dependencies, and more.

The piStar tool was selected for its open-source nature and because its author is associated with NOVA School of Science and Technology, which facilitates a smooth process for editing and contributing to the project on GitHub, should the outcomes of this research turn out to be beneficial for the tool. As it will be further discussed in the next paragraphs, this tool is used by many developers, however, mastering its complexities can pose a challenge. Therefore, enhancing the piStar tool to facilitate newcomers' grasp of the i* language holds significant potential in fostering inclusivity and improving the tool's usability.

The i* language emerged in the 1990s [62] as a framework for modeling intentional, strategic actor relationships, prioritizing goals and actors [63]. It consists of a modeling language along with reasoning techniques for analyzing created models. Embraced by academia in fields like requirements engineering and business modeling [31] and in many areas, like healthcare [20], security [33] and e-commerce [14], i* underwent adaptations, clarifying semantics and introducing new constructs for specific domains. The i* framework's flexibility has been beneficial for researchers, allowing for tailored models and methods. Yet, this flexibility presents challenges in broadening its use beyond experts due to difficulties in learning, teaching, and applying the framework, as well as technology providers' uncertainty about its core elements [17]. To address these issues and balance openness with accessibility, the i* research community initiated a standardization effort, resulting in the establishment of core principles within the i* language, known as i* 2.0. This initiative aims to maintain the framework's adaptability while providing a common foundation for its users [17].

Within the i* framework, stakeholders are depicted as actors relying on one another to accomplish objectives, execute tasks, and allocate resources. Goals are analyzed from the perspective of each actor, yielding a network of dependencies between actor pairs. Elements in i* 2.0 are categorized into Intentional Elements (Goals, Tasks, Resources and Qualities), Actors (General Actors, Roles and Agents), and Links (Refinement, Contribution, Qualification and Needed-By), as seen on figure **Figure 2.5** [17]. These components are manifested in two models: Strategic Dependency (SD) and Strategic Rationale (SR). The SD model describes a business organization in terms of the dependencies that actors have on each other in accomplishing their work [63], delineating links and external dependencies among organizational actors [26]. On the other hand, the SR model describes the reasoning that actors have about the different possible ways of organizing work [63], facilitating an examination of how goals can be achieved through collaborative contributions from multiple actors [26].

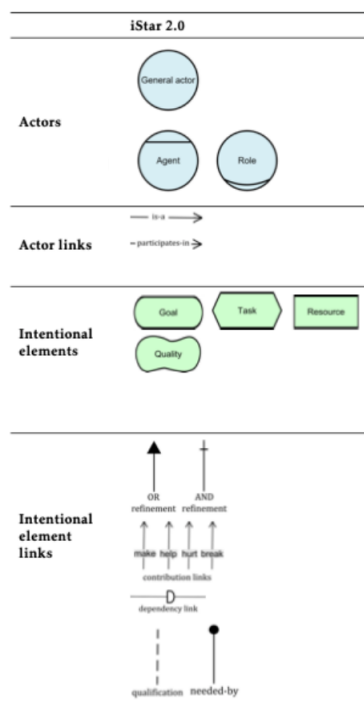


Figure 2.5: i* 2.0 elements, adapted from [1].

2.1.3 Other usability practices and techniques

Building and achieving good interfaces is one of the many goals of software development, but this goal can often prove to be more complex than it seems. With diverse variants and target users the evaluation of what is a good interface can often be subjective, adding to the already strenuous activity of software development. Thus, constructive and evaluative techniques, methods and tools have been developed and implemented in many software development processes.

When planning, building and developing interfaces, constructive tools are used to help navigate the decision-making process in order to build good interfaces. An example of this type of a constructive technique is User-Centered Design (UCD). UCD is a design philosophy and process in which the needs, wants, and limitations of end users of a product, service, or process are given extensive attention at each stage of the design process. In the context of interface design, this approach focuses on making interfaces more usable and user-friendly. UCD goes along with User and Task analysis, allowing for a full context analysis.

- User Analysis: Who are the users?
- Task Analysis: What does the user need to do?

Other techniques used in UCD are user observation, such as users thinking aloud to guide observers through their thought process, audio recording, video recording, computer logging and user notebooks [18], and interviews [18] or contextual interviews

which combine the previous two, observation and interviews. Finally, participatory design is particularly effective in the field of UCD, allowing developers to have access to a pool of representative users, including them directly in the design team [18]. Furthermore, there are golden rules and heuristics that a software team may adopt, so they can base their development and design decisions on those rules, or heuristics. Examples include Nielsen's Heuristics [42, 43], Shneiderman's 8 Golden Rules [49], Norman's 7 Principles [44], or Tog's Principles [22].

On the other hand, when interfaces are already built and we intend to evaluate how good an interface is, we use evaluative techniques. Some examples of evaluative techniques are user testing, Goals, operators, methods and selection rules (GOMS) to model user behavior and hierarchically decompose the problem in sub-objectives, cognitive walkthrough, Heuristic evaluation [43], formative evaluation, controlled experiments, statistical testing and questionnaires.

Regarding these two types of techniques that share the goal of developing good user interfaces, the GenderMag method can be used both in a constructive and an evaluative way. The GenderMag method can be used throughout the conception of interfaces, as a constructive technique, to build good interfaces, and it can be used as an evaluative technique to evaluate partially or totally developed interfaces.

In this study, we will employ the GenderMag method to assess a software development tool, details of which will be elaborated in Section 2.3. Our aim is to identify and address inclusivity concerns within the chosen software development tool. By leveraging the GenderMag method, we intend to uncover areas where the tool falls short in terms of gender inclusiveness and user-friendliness, and propose comprehensive solutions that cater to a broader range of users. For this purpose, it is expected that the GenderMag method will be employed along with some other evaluative techniques such as controlled experiments.

Controlled experiments are a method of evaluation where specific hypotheses are tested under a controlled environment. These experiments aim to understand the impact of various independent variables on dependent variables through systematic manipulation and observation. The essential components of controlled experiments are [6]:

- **Formulation of Hypothesis:** The experiment starts with a hypothesis, which is a prediction about the outcome. For example, hypothesizing that "Interface X is faster than Interface Y".
- **Manipulation of Independent Variables:** These are the variables that are intentionally varied or manipulated to observe their effect. For instance, changing interface designs, task types, or the number of menu items in a software application.
- **Measurement of Dependent Variables:** These variables are affected by the independent variables. They are measured to observe the impact of the changes in

independent variables. Examples include task completion time, the number of errors made, or user preferences.

- **Use of Statistical Techniques:** Statistical methods are applied to analyze how changes in the independent variables affect the dependent variables. This analysis helps in accepting or rejecting the hypothesis based on empirical data.
- **Control of Unknown Variables:** Efforts are made to eliminate or control unknown variables that might affect the experiment's outcome. This ensures that the observed effects are solely due to the manipulation of the independent variables.
- **User Participation:** Users involved in these experiments should be representative of the target population. The sample size can vary based on resources and the nature of the hypothesis being tested.

Controlled experiments are particularly useful in Human-Computer Interaction (HCI) studies to test specific aspects of user interfaces. They provide quantitative data that can be analyzed to make informed decisions about interface design and functionality.

2.2 Relevance of the question

Software products play a pivotal role in reaching and influencing a multitude of people across diverse markets and populations. Their impact extends beyond mere functionality, affecting the way individuals interact, businesses operate, and societies function. The appropriateness or inadequacy of these products can ripple through various sectors, shaping user and developer experiences, influencing market dynamics, and even contributing to societal shifts. Recognizing the broad reach and potential consequences of software products underscores the importance of their thoughtful development, ensuring they align with the needs and expectations of the wide array of users and contexts they touch.

Creating software that is non-inclusive is not only unethical but it also has a significant economic impact generating loss of market share since this product is not fit to answer to a diverse public and its needs, becoming useful to fewer costumers [28, 60].

Following the same train of thought mentioned in the Section 2.1.1.4 about InclusiveMag, while in this dissertation, the focus will be gender-inclusive interfaces and applicable methods that allow the detection and correction of issues standing in the way of statistical gender problem-solving styles, females are not the only group to be under-served. It is relevant to note that people with disabilities more often than not find themselves out of the software development target user characteristics [48].

When it comes to lack of representation in the development team, it is relevant to consider that non-inclusive software marginalizes and creates groups of people whose characteristics differ from the developers, and because of this, they end up “not fitting” and become under-served by software development teams [28]. While a more diverse

development team does not necessarily mean more inclusive software, believing this is the case may prove to be an obstacle to the implementation of the GenderMag method [25], since teams that follow this way of thinking might be less prone to considering other solutions. However, more diverse teams, with respect to the four factors of age, gender, experience and education, are more efficient and successful in their software development projects [2]. Diversity in software development teams inherently leaves less room for the introduction of gender bias bugs into the software, since people are naturally more alert to problems that affect them directly than to problems that affect other people. This makes a more diverse team better at identifying issues that would affect the different facets. Additionally, increased diversity in gender and in tenure are associated with greater productivity [54].

Research also shows that creating and designing software to be gender-inclusive can end up benefiting not only the marginalized group at hand but all users, with various facet values [25, 28, 55].

In summary, the significance of this study lies in its aim to bridge the current division and promote the inclusion of gender diversity within software development teams. By welcoming diversity, the likelihood of gender bias is significantly reduced. A team diverse in gender and experiences is better equipped at identifying and addressing a broad spectrum of problems, thus enhancing their problem-solving skills.

2.3 Focus

Given the prior sections, it seems clear that in the dynamic field of software development, the relationship between people's problem-solving styles and software design decisions can be seen as a critical challenge. With evidence pointing to problem-solving styles as gender-related, software developed by humans who unconsciously embed these personal choices and preferences in software is prone to accommodate the presence of gender biases, leading to gender-inclusiveness issues.

Addressing this challenge in the industry, the GenderMag method can be seen as a solution, comprising all the steps and systematic techniques mentioned above. This dissertation will focus on leveraging the GenderMag Method to systematically identify gender-inclusiveness issues, propose fixes for them, and assess the impact of those fixes in the overall usability of the piStar tool [46].

This brings us to the focal point of this study, centered on harnessing the GenderMag method in conjunction with the concept of developer experience.

The user perspective prioritizes efficiency and usability, considering the experience of interacting with a specific software product. In contrast, the developer perspective encompasses encounters with various artifacts and activities throughout a developer's engagement in software development. These could be divided into experiences regarding [21]:



Figure 2.6: Illustrative image for the developer experience concept [21].

- **Development infrastructure:** This includes tools for development and management, programming languages, libraries, frameworks, platforms, as well as the processes and methodologies used.
- **Feelings about work:** This encompasses feelings of respect, attachment, and a sense of belonging within the workplace.
- **The value of one's contribution:** This relates to how one's personal goals align with the project's objectives, including plans, intentions, and the level of commitment.

In **Figure 2.6** there is a representation of developer experience as a result of cognitive, affective, and conative elements interacting. This concept encompasses several complex sub-elements across its dimensions. The cognitive dimension involves intellectual interactions with development infrastructure, including the use of tools and software processes, where positive perceptions enhance developer experience. The affective dimension addresses emotional responses to work, including feelings of respect, belonging, and security, as well as attachment to people, teams, or work practices. The conative dimension relates to developers' perception of their contributions' value. It encompasses the intention behind actions, planning, and alignment of personal goals with collective objectives, enhancing the sense of purpose, motivation, and commitment. This alignment significantly influences developer experience positively.

Favorable emotions resulting from the use of a software development tool are likely to significantly enhance the sense of purpose, motivation, and commitment, thereby positively influencing the overall developer experience [21].

By analyzing a specific software tool, in this case the piStar tool, through the Gender-Mag's lens we expect to identify gender-inclusiveness issues, working towards a more positive developer experience.

BACKGROUND

This chapter reviews existing research relevant to this study accompanied by the GenderMag Design Catalog [23], allowing for the full understanding of the field of application of the GenderMag method to problem-solving software interfaces, along with some of the evaluative techniques mentioned before.

The structure of this section is organized around the GenderMag Design Catalog [23], where each design change category will be described in detail and for the empirically validated solutions, detailed examples will be given (examples based on [23]). Through the research and analysis of this catalog, we noticed that many of the fixes and solutions proposed by the GenderMag Design Catalog lack publicly available empirical validation through user studies. We will revisit this idea in Chapter 4.

3.1 Categories of design changes

The following subsections will present the design changes available in the GenderMag Design Catalog [23]. These design changes arose from the desire to reduce gender bias bugs, making interface design more inclusive, using the GenderMag method.

3.1.1 Ambiguous option

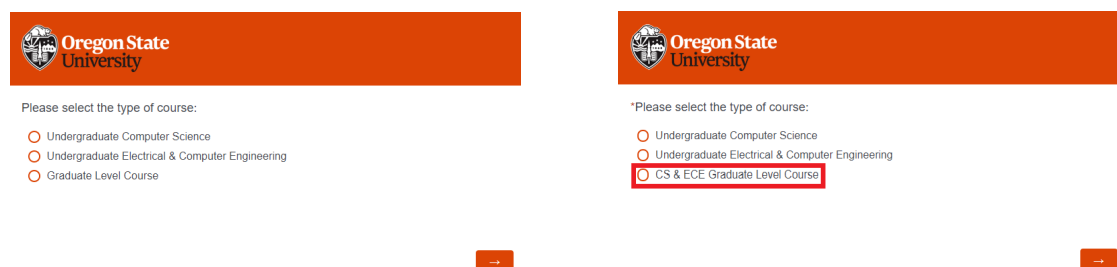


Figure 3.1: Interfaces representing the before and after the “ambiguous option” design change, respectively, regarding scenario one.

This design change should occur in two different scenarios. The first scenario, as depicted in **Figure 3.1** on the left image, is when the user has to choose from a list of

options and at first sight, there may not be an obvious option that fulfils the user's needs. In this case, depicted in the right image, the interface should display more details, specifying exactly what each option includes. This way people with low computer self-efficacy, comprehensive information processing style and high risk aversion are more likely to feel at ease when filling out the survey since they now feel they can follow the expected track.

Figure 3.2: Interfaces representing the before and after the “ambiguous option” design change, respectively, in regards to scenario one.

The second scenario, represented in **Figure 3.2** is when the user has to fill in the survey's text fields but the fields are marked only with titles and do not display examples and clarifications of the expectations for input, as seen in the image on the left, hurting especially users with low computer self-efficacy. Adding clarifications and examples, like in the picture on the right, helps users with low computer self-efficacy know what the interface expects from them, and those users with high computer self-efficacy since they will encounter fewer errors, as they know the expected input.

3.1.2 Big wall of text

Figure 3.3: Interfaces representing the before and after the “big wall of text” design change, respectively.

This design change should occur if there is a big wall of text, like the one represented in the image on the left of **Figure 3.3**, often an indicator of unnecessary information. A risk-averse user will not take the time to read the entire page because digesting unnecessary information can be overwhelming. This interface situation also affects users with a comprehensive information processing style and task-oriented motivations, since they can quickly find and gather information on a specific tool to accomplish a specific goal.

As represented in the image on the right of **Figure 3.3**, restructuring this wall of text by adding a "Learn more..." hyperlink allows users to locate topics easily throughout the page and decide on what topic they would like more information about. Specifically for risk-averse users, due to this design change, they no longer need to waste time reading the entire page. This change also aids the above-mentioned users with comprehensive information processing style and task-oriented motivations since information is now structured to allow for organized browsing.

3.1.3 Difficult first task

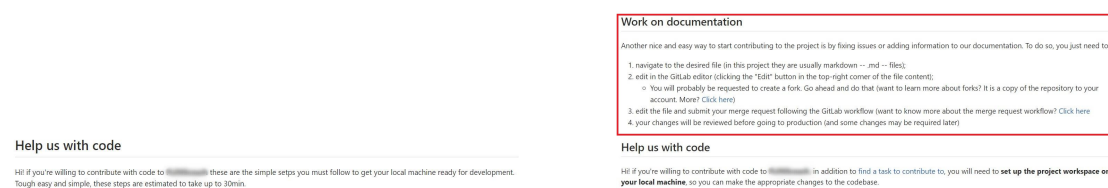


Figure 3.4: Interfaces representing the before and after the “difficult first task” design change, respectively.

This design change is intended to facilitate a smoother transition for users encountering challenging initial tasks. To illustrate, consider **Figure 3.4** and the image on the left, a scenario where a user aims to contribute to an OSS project, which can be an intimidating task. In this case, the interface solely offers guidance on code contributions, neglecting to provide any instructions or hints that contributions to documentation are also possible. This design affects process-oriented learners who would thrive on a step-by-step instructions approach and risk-averse and low computer self-efficacy users since tinkering with the project and the coding setup would not only take more work but also make room for errors.

A solution to this issue, as represented in the image on the right, would be to add a section for documentation contributions with step-by-step instructions. Process-oriented users can now lean on step-by-step instructions to complete their task of documentation contribution. This design change does not affect higher computer self-efficacy and risk-tolerant users since the solution only adds a section on an alternative way to complete the OSS project contribution while still allowing users to proceed with any other way they would prefer (code contribution, in this case). The step-by-step instructions also aid users with higher risk aversion and low computer self-efficacy.

3.1.4 Incomplete instructions

This design change should occur when instructions to continue following a specific set of steps are not organized or specific enough that user might run into a scenario that is not supported by the instructions. Like the one represented on **Figure 3.5** in the image on the left, where instructions for each Operating System (OS) were in different locations.

3.1. CATEGORIES OF DESIGN CHANGES

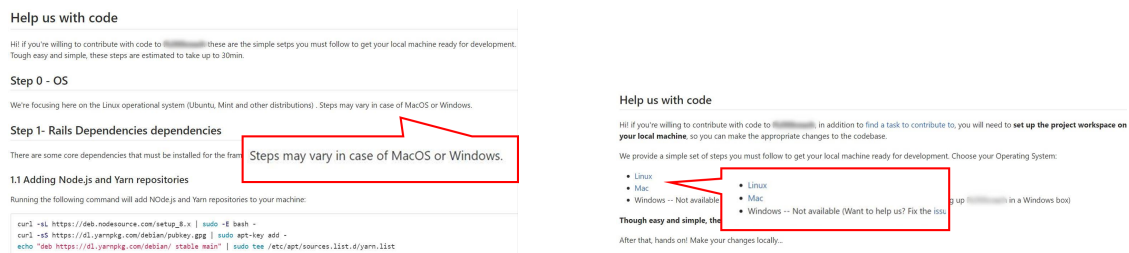


Figure 3.5: Interfaces representing the before and after the “incomplete instructions” design change, respectively.

This design can deter a task-oriented user from completing the task since they would not explore around and gather scattered instructions. This design also affects users with low computer self-efficacy, high risk-aversion and process-oriented learning style.

Reorganizing the instructions to consolidate them into a single, neatly arranged location will cater to different user needs effectively, as represented in **Figure 3.5** in the image on the right. Users who are driven by specific tasks will be able to quickly locate precisely what they require, while those who are motivated by technology can explore as much as they desire. Furthermore, this enhanced organization will be particularly beneficial for users who may have low confidence in their computer skills, those who prefer to avoid risks, and those with a process-oriented learning style.

3.1.5 Login wall

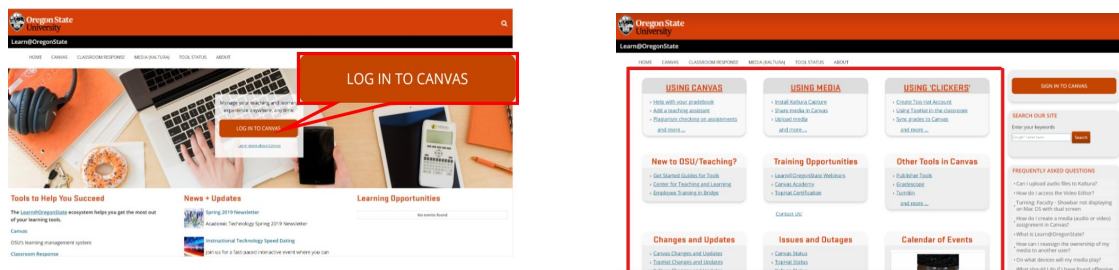


Figure 3.6: Interfaces representing the before and after the “login wall” design change, respectively.

This design change should be implemented in interfaces where access to general information is gated behind a sign-in requirement, just as represented in the left image of **Figure 3.6**. This design can discourage task-oriented users, since they would have to take a detour on their task to perform extra steps including personal information. This would feel like an obstruction and would likely cause users to abandon the task.

Redesigning the home page to prominently display general information enables users to navigate and find what they need effortlessly, eliminating the requirement for additional or superfluous steps. An example for this, is represented in the image on the right of **Figure 3.6**. Removing unfamiliar barriers can be particularly advantageous for users who are risk-averse, favor a process-oriented learning style, or have low computer self-efficacy.

3.1.6 Long list

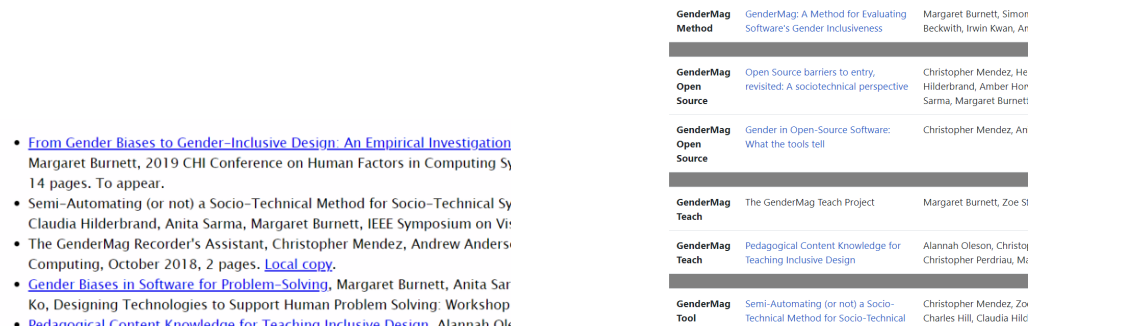


Figure 3.7: Interfaces representing the “long list” design change, respectively.

This design modification focuses on addressing the issues with lengthy and unorganized lists. Such lists can be challenging to comprehend and sift through. An example of such a list can be found in **Figure 3.7**, in the image on the left. Users who are risk-averse would benefit from being able to quickly find what they need without spending excessive time navigating through extensive lists. Additionally, this change is relevant for users who have a detailed information processing style and those who are motivated by specific tasks.

This structured approach to a list, like in the image on the right of **Figure 3.7**, enables a risk-averse user to quickly identify the relevant category, thus avoiding the need to peruse the entire page. Such an arrangement does not inconvenience a risk-tolerant user, as they would not face an organized list as a problem. Additionally, categorizing information aids users who process information comprehensively and those with task-oriented motivations.

3.1.7 Many options to choose from

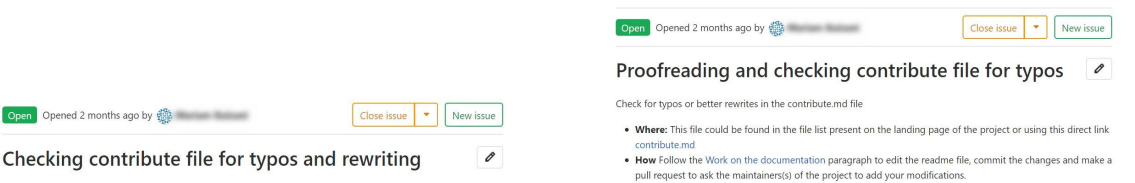


Figure 3.8: Interfaces representing the before and after the “many options to choose from” design change, respectively.

This design change should take place in two different scenarios. The first scenario is when the user, in the midst of completing a task, is faced with a page where the title is too vague and there is a poor description of the task at hand, like the image on the left of **Figure 3.8**. This can heavily affect users with a comprehensive information processing style since they need a more a detailed description of the issue in question, in order to make an informed selection according to their skill level. Moreover, this vagueness can demotivate users, especially those with low confidence in their computer skills, from completing their intended task.

Addressing this issue involves enhancing both the title and description to be more explicit, thereby equipping users with the confidence needed to proceed with their task, just as exemplified in the image on the right of **Figure 3.8**.

By offering more comprehensive details about the issue, users with a comprehensive information processing style are more likely to engage, as they can better assess whether the task matches their skill level. This approach can also accommodate users with a selective information processing style, since the key information will be highlighted in a list, seeing that selective information processing style users prefer this to dense text blocks.



Figure 3.9: Interfaces representing the before and after the “many options to choose from” design change, respectively.

In another instance, the design alteration becomes pertinent when users are presented with a list of tasks or issues from which they must choose one that aligns with their skill level, like the one represented in the image on the left of **Figure 3.9**. The challenge arises when this list fails to provide sufficient details about the skills required for each task. Users who are cautious about taking risks are reluctant to invest time in a task that may not suit their skill set. This situation also poses difficulties for users who have a preference for detailed information processing, those driven by task-specific motivations, and individuals with low confidence in their computer skills.

To tackle this problem, implementing color-coded tags to denote the skills needed for each task can be effective, like the list represented in the image on the right of **Figure 3.9**. This approach allows risk-averse users to quickly discern which issues align with their abilities, enabling them to proceed without unnecessary delay. This strategy does not impede risk-tolerant users, as they are typically not affected by this particular issue.

Color categorization also aids users with comprehensive information processing style, task-oriented motivation and low computer self-efficacy, as it simplifies the process of identifying suitable tasks.

3.1.8 Multi-step task



Figure 3.10: Interfaces representing the before and after the “multi-step task” design change, respectively.

When a user is tasked with completing a multi-step activity, the absence of a visible progress indicator, like in the interface shown in the image on the left of **Figure 3.10**, can

easily lead to confusion or a sense of being lost in the process. This is especially true for risk-averse users, who often have limited time and wish to gauge how much of the task remains before deciding to continue. This situation also affects users with a process-oriented learning style. To remedy this, incorporating a progress bar at the top of the user interface for the entire duration of the task can be beneficial, just as represented in the image on the right of **Figure 3.10**. This enables users to monitor their advancement through the task and understand exactly how much remains. While this feature is particularly helpful for risk-averse users, it also does not negatively impact risk-tolerant users.

Additionally, this method of displaying progression is advantageous for users with a process-oriented learning style, as it provides a clear visualization of their journey through the task.

3.1.9 Paragraph of instructions

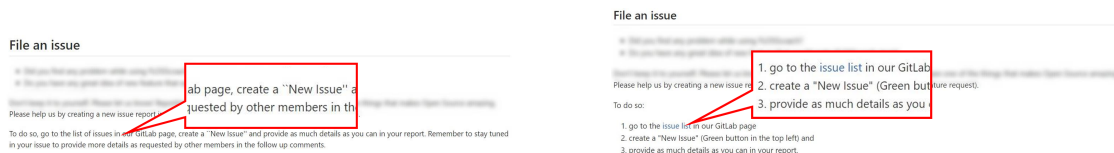


Figure 3.11: Interfaces representing the before and after the “paragraph of instructions” design change, respectively.

This design change is needed when instructions are provided in a paragraph, and the user is directed to visit a specific page, but no direct link is offered, like depicted in the image on the left of **Figure 3.11**. Users with selective information processing desire to quickly and easily find the necessary information without being bombarded by superfluous text. This issue also impacts users driven by task-oriented motivations and those with low computer self-efficacy. Resolving this problem involves setting up a clearly numbered, step-by-step instruction section, along with a direct hyperlink to the specific page the user needs to access, like shown in the image on the right of **Figure 3.11**. This approach guides users with selective information processing towards an organized list of instructions rather than a dense paragraph. The instructions are hyperlinked, eliminating the need for extra navigation. Moreover, this setup does not hinder tinkerers from exploring. Additionally, using numbered instructions and direct hyperlinks benefits users with task-oriented motivations and those with low computer self-efficacy.

3.1.10 Reason to follow link unclear

This design alteration is necessary in three distinct scenarios. The first is when a user encounters a list of links on the interface without any details about their destinations or contents, as depicted in the image on the left of **Figure 3.12**. Users with a comprehensive information processing style require more context to make informed decisions. This situation also affects users with task-oriented motivations and those who are risk-averse.

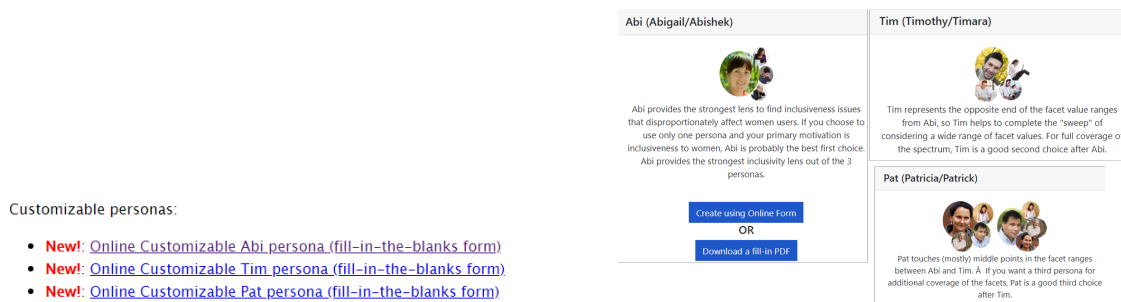


Figure 3.12: Interfaces representing the before and after the “reason to follow link unclear” design change, respectively, regarding scenario one.

To address this, a solution is to create separate sections for each link, accompanied by contextual and, if possible, visual information about the content of each link, as represented in the image on the right of **Figure 3.11**. This provides users with a comprehensive information processing style with sufficient data to make educated choices. Additionally, this change does not adversely affect selective information processors, as they have the option to directly select one of the sections or links without reading the preceding information.

Moreover, offering more contextual information is also beneficial for users with task-oriented motivations and those inclined to avoid risks, as it aids in making more informed decisions.

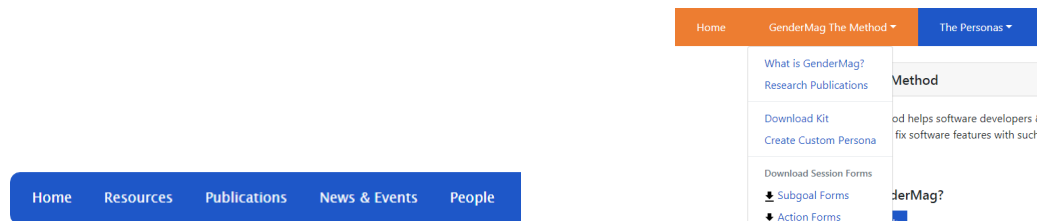


Figure 3.13: Interfaces representing the before and after the “reason to follow link unclear” design change, respectively, regarding scenario two.

In the second scenario, users encounter tabs in the header of an interface that fail to reveal any information about the page’s contents, just as shown in the image on the left of **Figure 3.13**, compelling them to visit the page to discover its contents. This is particularly inconvenient for risk-averse users, who prefer to directly locate the tab containing the information they need, rather than spending time navigating through multiple tabs. This issue also impacts users with task-oriented motivations and those with a comprehensive information processing style. A viable solution is to implement hover drop-down menus, as depicted in the image on the right of **Figure 3.11**, which reveal all the sub-topics available on a particular page when a user hovers over a tab. This feature enables risk-averse users to quickly ascertain the information each tab contains, thereby saving time and avoiding unnecessary navigation. Moreover, this solution doesn’t negatively affect risk-tolerant users, as they are not typically bothered by this issue.

Additionally, hover drop-down menus are beneficial for users with task-oriented motivations and those with a comprehensive information processing style, as they provide immediate insight into the content of each page without the need for clicking and exploring.



Figure 3.14: Interfaces representing the before and after the “reason to follow link unclear” design change, respectively, in regards to scenario three.

The third scenario is when the user, in the midst of completing a task, encounters a page where there is a vague sentence telling them to read a certain page, with a direct link to it, as shown in the image on the left of **Figure 3.14**. If the sentence is too vague and provides little to no information about what the user should expect from clicking on the link, a task-oriented user will face this as exploring outside of the task and they might want to abandon the task, due to the lack of orientation. This also affects users with a comprehensive information processing style and high risk aversion since the reason to click on this link is not fully explicit.

In this case, a solution would be to allow the sentence to contribute with more information about what the user should expect when clicking the link provided, like depicted in the image on the right of **Figure 3.14**. Task-oriented users would benefit from this since they would know what to expect and will not face clicking the link as exploring while technology-motivated users can still explore as much as they want.

Additionally, this solution also makes task completion easier for users with a comprehensive information processing style and high risk aversion since it eliminates the feeling of stepping into the unknown by clicking the link.

3.1.11 Unexplained information

If the interface displays unexplained information, there is a high risk of confusing the user since he is exposed to potentially unnecessary information.

This issue category is heavily interconnected with the user’s computer self-efficacy, motivations, attitude toward risk and information processing learning style. In accordance with this, removing contradicting information helps significantly to increase user confidence. A detailed example of this design change is in the Subsection 3.2.1.

3.1.12 Help users collect, compare, and integrate information

This scenario occurs when a user needs help with a given task, either to understand its functioning or its key words better. In this case, the design change should take place in a scenario where the user would have to leave their context and visit a help page or a dictionary, as depicted in the image on the left of **Figure 3.15**, or where the user can open a tooltip with information but it would disappear once they click something else.

3.2. AN EXAMPLE OF AN EMPIRICALLY VALIDATED DESIGN CHANGE



Figure 3.15: Interfaces representing the before and after the “help users collect, compare, and integrate information” design change, respectively.

This design can pose a challenge for comparing and analyzing information alongside the page where the help was first needed. Additionally, two pieces of information can not be compared side by side since the tooltips do not support selecting other items. This design presents two extremes: users are either allowed a small piece of information at a time (tooltip) or by abandoning the task context, they get a huge amount of information (dictionary). These extremes are not ideal for comprehensive information processors who want all of the information they need and only the information they need, alienating the presence of unnecessary information and the unmet desires for the needed information.

One way of addressing this non-inclusive design is to allow tooltips to stay open and to be moved through the screen, allowing side-by-side comparisons and organization, as shown in the image on the left of **Figure 3.11** Comprehensive information processing style users now have the flexibility to resize, move and pin tooltips which allows to compare information, having the opportunity to revisit concepts and ideas throughout the course of the task.

3.2 An example of an empirically validated design change

To fully grasp the application and validation of these design changes through user studies, it's crucial to delve into relevant literature. The analysis will focus on studying the validation of design changes related to the category of 'Unexplained Information', as outlined in Vorvoreanu et al.'s 2019 study on gender [56].

3.2.1 Example of Unexplained information

An instance of this category can be found in [56], where unexplained information seems to steer users away from certain functionalities. This paper aims to verify if applying the GenderMag method leads directly to interfaces with fewer gender biases. It describes how two HCI researchers analyzed a software product using GenderMag and derived design changes. Then, an empirical study of participants using the original software versus the new version took place, and the results showed that using the GenderMag method

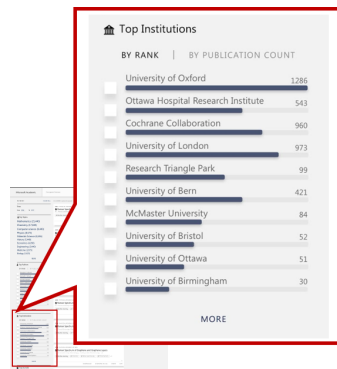


Figure 3.16: Screenshot of the original table. It seems contradictory since the "Rank" filter is selected but the numbers on the right side do not seem to be in order. However, the numbers on the right are actually the number of publications for each university and doesn't affect the rank. [23]

improved not only the women's success when using the new version but also the men's, making the gender gap disappear [56].

In this case, where the personas of Abby and Tim were chosen to carry out the tasks, the design issue impacts both personas and their facets. This scenario asked that the users look for the top-ranked institution in a specific field, and cite the most cited paper published within 15 years. This can be divided into two subgoals: 1) find papers about the specific field published in the top institution and 2) cite the paper. From subgoal 1), the researchers derived two issues. They decided that even forming this subgoal could be challenging for Abby and Tim (Issue 1) and then to actually perform the action, the personas would have to locate the filter (Issue 2). These issues would influence all of Abby's five facets and two of Tim's [56]:.

- **Motivation:** since the Abby persona is task-oriented and learning new features might be seen as a detour, the filter seen in **Figure 3.16** might be faced as a feature Abby is not familiar with, along with the fact that this filter looks more like a visualization, can steer Abby from using this feature.
- **Information processing styles:** These issues impacts both Abby and Tim when it comes to this facet. As Abby tends to process information comprehensively, she would analyze the page and find the filter, but she would feel more confident about using this feature if there was more information about it, what it does and what the numbers correspond to. However, Tim would try to follow the first promising information (connected to tinkering) which would likely lead Tim to perform the wrong action before even noticing this filter.
- **Computer self-efficacy:** This facet of Abby's would also be impacted by these issues since Abby's low computer self-efficacy will not thrive when combined with the filter's lack of feedback, leaving room for Abby to feel like technological issues or no success message is a signal for that a mistake was made by the user.

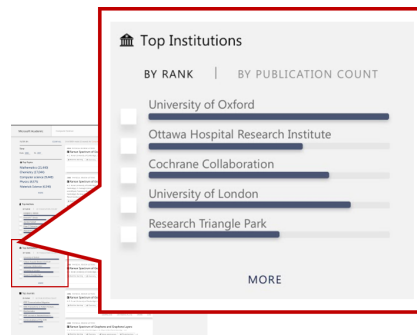


Figure 3.17: Screenshot of the new version [23]

- **Risk aversion:** Abby's risk aversion and lack of interest for tools lacking explanation for their effects, would add to Abby's already existent desire for more information about this filter.
- **Tinkering:** Abby's and Tim's tinkering, directly connected to their learning style, can surely be impacted by these issues. As mentioned before, Tim would tinker and probably perform the wrong action before noticing this filter. Abby, however, who prefers to learn processes first and gather information, would be severely impacted by the interface's lack of information about its functionalities.

In alignment with the goal of the paper, fixes and solutions for these issues were proposed, as shown in **Figure 3.17**, and later, tested. The fixes had in mind all of the facets affected as shown below:

- **Motivation and risk aversion:** These two facets were addressed by making the filter look more like a filter. This was achieved by removing the number of publications, which in turn emphasizes the filter-action checkboxes.
- **Information processing styles:** To aid Tim in regards to this facet, the list was shortened, now showing the top 5 institutions.
- **Computer self-efficacy:** To help Abby, a list of filters that had been applied was added to the top of the filtered search results. This way the users had confirmation as to if they had accomplished their goals with this tool.
- **Tinkering:** Since Tim's issue with tinkering was magnified due to his information processing style, the fix to aid Tim's information processing style was determined to be enough to also help Tim with this facet.

INCLUSIVITY BUGS' DETECTION AND IMPLEMENTATION

This chapter is structured into two principal sections: the inclusivity bugs' detection section, which documents the process of applying the GenderMag method to the piStar tool interface [46], and the implementation section, which provides a description of the code implementation of the fixes to the previously found bugs, with the guidance of the the GenderMag Design Catalog [23].

4.1 Inclusivity Bugs' Detection

As mentioned before in the previous chapters, this research aims to delve into gender-inclusive software design, with a spotlight on software development tools. Leveraging the GenderMag method, a structured approach developed to identify and rectify gender biases in software, the study intends to scrutinize a software development tool to unearth gender inclusiveness issues. By applying the GenderMag framework, the research will evaluate the tool's features to spot gender biases that could hinder inclusivity.

The proposal is anchored on the premise that enhancing gender inclusivity in software tools promotes equality and broadens the developer base, making software more accessible and developer-friendly for a diverse audience. The study underscores that software tools, inherently shaped by their developers' unconscious biases, may exhibit gender biases that can be systematically identified and amended using the GenderMag method.

The objective is to identify problems and suggest solutions using the GenderMag Design Catalog's solutions [23]. The tool selected, as mentioned and further detailed in Section 2.1.2, was the i* 2.0 modeling tool, piStar [46].

4.1.1 Application of the GenderMag Method to piStar

The goal of using the GenderMag method to evaluate the tool's features to identify and rectify gender biases that could hinder inclusivity, requires a preparation of this method.

Abi (Abigail/Abishek) GenderMag Persona



- 28 years old
- Employed as a Software Engineer
- Lives in Cardiff, Wales

Abi has always liked music. While traveling to work in the morning, Abi listens to music from a wide variety of styles. Some nights Abi exercises or stretches, and sometimes likes to play computer puzzle games like Sudoku. Abi likes scanning all her emails first to get an overall picture before answering any of them.

Background and Skills

The technologies at Abi's new employer are new to her. Abi likes math and working with logic. She considers herself a numbers person. Example: Abi works as an Software Engineer and is comfortable with the technologies Abi uses regularly. Abi just moved to this employer 1 week ago, and their software systems are new to Abi writes and edits spreadsheet formulas for work. During free time, Abi also enjoys working with numbers and logic. Abi especially likes working out puzzles and puzzle games, either on paper or on the computer.

Motivations and Attitudes

Motivations: Abi uses technologies to accomplish her tasks. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.

Computer Self-Efficacy: Abi has lower self confidence than her peers about doing unfamiliar computing tasks. If problems arise with her technology, she often blames herself for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.

Attitude toward Risk: Abi's life is a little complicated and she rarely has spare time. So she is risk averse about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they're more predictable about what she will get from them and how much time they will take.

Attitude to Technology

Information Processing Style: Abi tends towards a comprehensive information processing style when she needs to gather more information. So, instead of acting upon the first option that seems promising, she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, her style is "burst-y"; first she reads a lot, then she acts on it in a batch of activity.

Learning: by Process vs. by Tinkering: When learning new technology, Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

Figure 4.1: Abi persona card with custom information.

Following the detailed description of the method's process in Section 2.1.1.3, we must follow a few steps:

1. Select one or more personas from the persona set and tailor the customizable portions of the persona if deemed as needed.
2. Preparatory Phase - define the list of tasks to be completed as well as the ideal sequence of each specific task.
3. Analysis Phase - evaluate the software product through a gender-specialized cognitive walkthrough with each selected persona.

In the case of this software product let's consider the Abi and Tim personas since they represent the two ends of the GenderMag persona spectrum. When it comes to the personas' customization, there is some value in making both Abi and Tim Software Engineers making them more susceptible to encounter the particular tool in evaluation. To generate the persona cards represented in **Figure 4.1 and 4.2** we used the GenderMag's online form¹, that allows the method's users to easily generate persona cards with the desired custom information while also ensuring that only the customizable information is changed.

Having selected our persona set, it's time to transition to the Preparatory Phase. During this stage, it's crucial to outline a set of tasks aimed at evaluating user performance with

¹https://gendermag.org/custom_persona.php

Tim (Timothy/Timara) GenderMag Persona



- 28 years old
- Employed as an Software Engineer
- Lives in Cardiff, Wales

Tim loves public transportation. Tim knows several routes to get to work from home and always exploring ways to optimize trips into the office. Some nights Tim plays computer games with some online friends. Tim starts work with answering emails one by one. Sometimes this backfires, if there is a second related message he hasn't read yet, but he doesn't mind sending a follow-up email.

Background and Skills

The technologies at Tim's new employer are new to him. Tim likes math and working with logic. He considers himself a numbers person. Example: Tim works as an Software Engineer and is comfortable with the technologies Tim uses regularly. Tim just moved to this employer 1 week ago, and their software systems are new to Tim writes and edits spreadsheet formulas for work. During free time, Tim also enjoys working with numbers and logic. Tim especially likes working out puzzles and puzzle games, either on paper or on the computer.

Motivations and Attitudes

Motivations: Tim *likes learning all the available functionality on all of his devices* and computer systems he uses, even when it may not be necessary to help his achieve his tasks. he sometimes finds himself exploring functions of one of his gadgets for so long that he loses sight of what he wanted to do with it to begin with.

Computer Self-Efficacy: Tim has *high confidence in his abilities with technology*, and thinks he's better than the average person at learning about new features. *If he can't fix the problem, he blames it on the software vendor.* It's not his fault if he can't get it to work.

Attitude toward Risk: Tim *doesn't mind talking risks using features of technology*, that haven't been proven to work. When he is presented with challenges because he has tried a new way that doesn't work, it doesn't changes his attitudes toward technology.

Attitude to Technology

Information Processing Style: Tim leans towards a selective information processing style or "depth first" approach. That is, he usually *delves into the first promising option, pursues it, and if it doesn't work out he backs out* and gathers a bit more information until he sees *another option to try*. Thus, his style is very incremental.

Learning: by Process vs. by Tinkering: Whenever Tim uses new technology, he tries to construct his own understanding of how the software works internally. He *likes tinkering and exploring* the menu items and functions of the software in order to build that understanding. Sometimes he plays with features too much, losing focus on what he set out to do originally, but this helps him gain better understanding of the software.

Figure 4.2: Tim persona card with custom information.

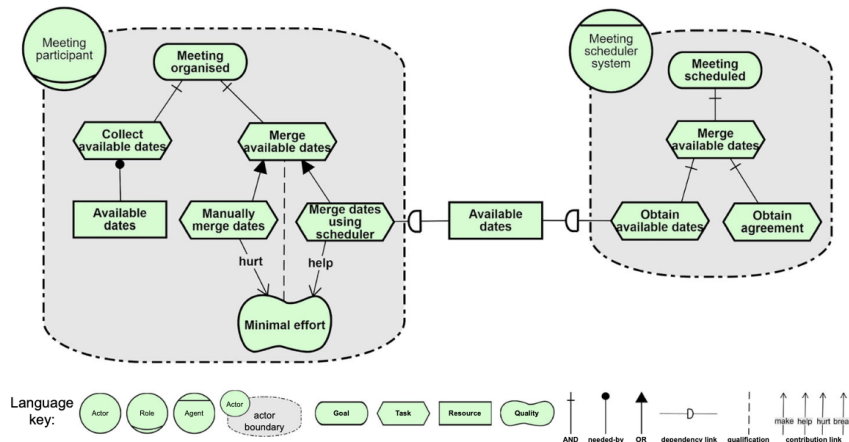


Figure 4.3: Illustration depicting the envisioned model to attain through the goals and subgoals [27].

the interface's features. The goals and subgoals should align towards a common goal, a specific model, in order to maintain the cognitive walkthrough closely resembling a real-world first-time user experience. The tasks within the GenderMag sessions will work towards building the model represented in **Figure 4.3**, as it will become clear now that we enter the Analysis Phase.

Circling back to Chapter 2, more precisely to **Figure 2.2**, on Section 2.1.1, we need to develop a detailed plan where each subgoal and action is described.

1. Pick a persona

2. Pick a use case/scenario in our tool
3. Walk through scenario via “intended” subgoals and actions

An important information to understand the list above is to know that the “intended” subgoals and actions are the developer’s intended subgoals and actions for the user, so it corresponds to what the developer wanted the user to want to do in this scenario and not what the user, considering all their facets, might really want to do. Furthermore, for a clearer differentiation between a subgoal and an action, both intended by the developers, we can employ this analogy: a subgoal resides in the mind, while an action manifests in physical movement. With this perspective, transitioning from the question “Developers, what subgoal would Abi/Tim need to formulate to accomplish this scenario?” to “Developers, what actions do Abi/Tim need to undertake to achieve this subgoal?” becomes seamless.

Following this, the integration of the developer’s perception and their involvement in the application of the GenderMag method is crucial, or else the method becomes a chain of assumptions and could very well lead the evaluation astray. The involvement of the piStar tool’s author, João Pimentel was requested and accepted, so the assumptions made throughout the GenderMag sessions that resulted in “maybe” results were discussed and verified by the tool’s developer, which enriches the application validity. These sessions will be later discussed and described in detail in subsection 4.1.2.

To summarize, we will generate a card similar to the one situated in the bottom left corner of **Figure 2.2** for every scenario under examination. It’s crucial to scrutinize the sequence of steps outlined in this specific card to elucidate the transitions between subgoals and actions. To assess the likelihood of the persona’s actual subgoal aligning with the intended one, we must evaluate it from the persona’s perspective, asking ourselves: “Would Abi have formulated this subgoal...?” Nonetheless, to proceed with the analysis, even if our result is “no” or “maybe”, for the purpose of evaluating the intended action segment, we must presume that even if Abi’s subgoal didn’t originally match the intended one, it does so now. Only after realizing that this was indeed Abi’s subgoal or forcing this to be Abi’s subgoal, can we proceed with the evaluation.

If this evaluation unearths some “no” and/or “maybe” answers the interface might potentially have some inclusivity bugs. In a real world analysis, these potential bugs would be ordered by priority and inserted into a bug tracking system, in our case they will be ranked and ordered as well and put into a list.

The goal will be to have various scenarios to be analyzed through the lens of our two chosen personas, Abi and Tim. Our initial testing scenario involves clearing all sample material from the piStar tool’s working board to construct our own model. Let’s analyze this scenario with both of our personas.

GenderMag’ing with Abi: “Clear piStar working board”

1. **Subgoal 1:** Find one item to be erased: Will Abi have formed this subgoal...?

- a) Yes/no/**maybe**. Why? Abi's learning style and comprehensive information processing style.
- 2. **Action 1:** Click on one item to be erased.
 - a) Will Abi know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
- 3. **Subgoal 2:** Delete the item: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe
- 4. **Action 2a:** Click on keyboard "delete" button.
- 5. **Action 2b:** Click on the "delete" button present on the left toolbar.
 - a) Will Abi know what to do?
 - i. Yes/no/**maybe**. Why? Abi's learning style.
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
- 6. **Subgoal 3:** Confirm item deletion: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe
- 7. **Action 3:** Click on "OK" button in the pop-up alert.
 - a) Will Abi know what to do?
 - i. Yes/no/**maybe**. Why? Abi's risk-aversion.
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe

In summary, the first challenge we faced with Abi in this scenario would arise from her realization that to clear all items from the working board, they would need to individually select and delete each item iteratively, one at a time, as for actions 2a and 2b. Once Abi becomes aware of this, the only potential issue that might arise is whether they correctly associate the option to use the keyboard's delete button or even notices the small "delete" button present on the left toolbar that displays the selected item's properties. As for action 3, the issue stems from Abi's high risk aversion, since confirming to the deletion of an item can, at this point and per the pop-up alert message, feel irreversible. Therefore, the responses to these are "maybe". Additionally, there is another way to clear the working board, which relies on an interface function called "New Model". This feature is available

in the top toolbar when the option “File” is selected and allows the user to swap the current model for a new one. While effective, this option might not be immediately intuitive in the sequence of selecting an item to eliminate, as deleting an item doesn't inherently suggest the notion of swapping out the entire model. However, if the intended subgoal 1 was “Swap model for a new one”, the method would follow a different path.

GenderMag'ing with Abi: “Clear piStar working board”

1. **Subgoal 1:** Swap model for a new one: Will Abi have formed this subgoal...?
 - a) Yes/no/**maybe**. Why? Abi's learning style and comprehensive information processing style.
2. **Action 1:** Click on option “File”.
 - a) Will Abi know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
3. **Subgoal 2:** Generate new model: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe
4. **Action 2:** Click on “New Model”.
 - a) Will Abi know what to do?
 - i. **Yes**/no/maybe.
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
5. **Subgoal 3:** Confirm the model swap: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe
6. **Action 3:** Click on “OK” button in the pop-up alert.
 - a) Will Abi know what to do?
 - i. Yes/no/**maybe**. Why? Abi's risk-aversion.
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe

In either of these scenarios, which vary in their initial subgoals, the pop-up alert might potentially deter Abi from confirming both the deletion of the item and the switch to a new model. Abi may fear losing the model example they initially encountered on the

page and being unable to revisit it. This pop-up fails to provide any guidance on how to access this model example again. Nevertheless, we'll exclude the "maybe" result stemming from this latest method application, as the leap in logic between the overall session goal and subgoal 1 is rather significant. For Tim, however, the process would take a slightly different route.

GenderMag'ing with Tim: "Clear piStar working board"

1. **Subgoal 1:** Find one item to be erased: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
2. **Action 1:** Click on one item to be erased.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Delete the item: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
4. **Action 2:** Click on keyboard "delete" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
5. **Subgoal 3:** Confirm item deletion: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
6. **Action 3:** Click on "OK" button in the pop-up alert.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

Taking into account all aspects of Tim's facets, it's evident that he feels at ease in situations involving risk and is likely to navigate through the interface by pursuing the most promising leads. This approach enables Tim to efficiently maneuver through the interface and attempt to delete the items without any aversion to risk. And the same would be true for Tim if subgoal 1 was "Swap model for a new one".

GenderMag'ing with Tim: "Clear piStar working board"

1. **Subgoal 1:** Swap model for a new one: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
2. **Action 1:** Click on option "File".
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Generate new model: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
4. **Action 2:** Click on "New Model".
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe.**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
5. **Subgoal 3:** Confirm the model swap: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
6. **Action 3:** Click on "OK" button in the pop-up alert.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

Let's analyze a second and third scenario with both of our personas, that will take us closer to the intended model in **Figure 4.3**. In the second scenario Abi and Tim will have to create a new item role, however in the third one they will have to create an agent. Since both of the scenarios are similar in sequence, both were applied in the same text block.

GenderMag'ing with Abi: "Create a role item"

1. **Subgoal 1:** Create a new role/agent item: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on actor drop down menu.
 - a) Will Abi know what to do?

- i. **Yes/no/maybe**
- b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Choose the actor type "role/agent": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click on "role/agent" option.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 5. **Subgoal 3:** Display the role/agent item on the working board: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 6. **Action 3:** Click on working board on an empty space.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: "Create a role/agent item"

- 1. **Subgoal 1:** Create a new role/agent item: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
- 2. **Action 1:** Click on actor drop down menu.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Choose the actor type "role/agent": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click on "role/agent" option.

- a) Will Tim know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes**/no/maybe
5. **Subgoal 3:** Display the role/agent item on the working board: Will Tim have formed this subgoal...?
- a) **Yes**/no/maybe
6. **Action 3:** Click on working board on an empty space.
- a) Will Tim know what to do?
 - i. Yes/no/maybe.
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes**/no/maybe

For the forth scenario we will change the names of the created items. There are two methods for renaming an item: one involves modifying the name directly within the dedicated column showing the item's properties on the left, while the other entails double-clicking on the item, triggering a text box where the new name can be entered and the action confirmed. Let's assume that the developers' intended approach aligns more closely with the first method, given that the property column was specifically included by the developer for editing purposes. Therefore, the scenario unfolds as follows:

GenderMag'ing with Abi: "Change name of role item"

- 1. **Subgoal 1:** Access the dedicated column displaying the properties of the item on the left and find property "name": Will Abi have formed this subgoal...?
 - a) Yes/no/maybe
- 2. **Action 1:** Click on default name "agent".
 - a) Will Abi know what to do?
 - i. **Yes**/no/**maybe**. Why? Abi's information processing style and learning style.
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
- 3. **Subgoal 2:** Change the item's name to "Meeting participant": Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe

4. **Action 2a:** Write "Meeting participant" in the text box.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
5. **Subgoal 3:** Confirm the action: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
6. **Action 3:** Click on the check button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

The "maybe" result in this session arises because the name editing field is not sufficiently apparent; it resembles a link rather than a field. Risk-averse users might find this discouraging and perceive a potential risk if clicking this "link" could lead them away from their current environment. However, for the Tim persona the session goes smoother.

GenderMag'ing with Tim: "Change name of role item"

1. **Subgoal 1:** Access the dedicated column displaying the properties of the item on the left and find property "name": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
2. **Action 1:** Click on default name "agent".
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Change the item's name to "Meeting participant": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
4. **Action 2a:** Write "Meeting participant" in the text box.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**

- b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 5. **Subgoal 3:** Confirm the action: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
- 6. **Action 3:** Click on the check button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

The fifth scenario closely mirrors the fourth, as the objective remains to assess the agent item's name as "Meeting scheduler system." To maintain brevity, further elaboration on this scenario will be omitted. The next steps in this session take us into the realm of the SR inside the role and agent items we have created and the dependency between them. In order to maintain organization, let's branch the session in two, first we will tackle the role named "Meeting participant" and finish its SR and then we will do the same for the agent named "Meeting scheduler system". We will start by adding a goal item on the role item named "Meeting participant", as follows:

GenderMag'ing with Abi: "Add a goal to the role's SR"

- 1. **Subgoal 1:** Create a new goal item: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 2. **Action 1:** Click on "Goal" button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Put goal item inside "Meeting Participant": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click inside the "Meeting Participant" boundary.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**

b) If action... will Abi see progress to subgoal?

i. **Yes/no/maybe**

GenderMag'ing with Tim: "Add a goal to the role's SR"

1. **Subgoal 1:** Create a new goal item: Will Tim have formed this subgoal...?

a) **Yes/no/maybe.**

2. **Action 1:** Click on "Goal" button.

a) Will Tim know what to do?

i. **Yes/no/maybe**

b) If action... will Tim see progress to subgoal?

i. **Yes/no/maybe**

3. **Subgoal 2:** Put goal item inside "Meeting Participant": Will Tim have formed this subgoal...?

a) **Yes/no/maybe**

4. **Action 2:** Click inside the "Meeting Participant" boundary.

a) Will Tim know what to do?

i. **Yes/no/maybe**

b) If action... will Tim see progress to subgoal?

i. **Yes/no/maybe**

The next steps in this branch would logically be to change the goal's name. However, the genderMag application of this task would be very similar to the "GenderMag'ing with Abi/Tim: "Change name of role item".", therefore let's consider the goal's name changes to "Meeting organized", with the same results as the GenderMag'ing section mentioned before. Next, we want to create two tasks that are going to be children to the "Meeting organized" goal, both with an AND relationship. Let us create the first task.

GenderMag'ing with Abi: "Add a task to the role's SR"

1. **Subgoal 1:** Create a new task item: Will Abi have formed this subgoal...?

a) **Yes/no/maybe.**

2. **Action 1:** Click on "Task" button.

a) Will Abi know what to do?

i. **Yes/no/maybe**

b) If action... will Abi see progress to subgoal?

- i. **Yes/no/maybe**
- 3. **Subgoal 2:** Put task item inside “Meeting Participant”: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click inside the “Meeting Participant” boundary.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: “Add a task to the role’s SR”

- 1. **Subgoal 1:** Create a new task item: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 2. **Action 1:** Click on “Task” button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Put task item inside “Meeting Participant”: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click inside the “Meeting Participant” boundary.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

As before, the task’s name would be changed to “collect available dates” following the same processes mentioned above for name editing. To finalize this three item tree, we need to create one more task called “Merge available dates”. To simplify, consider the same application above for creating a task and editing its name. Now, it has come the time to create the AND relationship between the two tasks and the parent item, the “Meeting organized” goal.

GenderMag'ing with Abi: “Add an AND relationship between child and parent”

1. **Subgoal 1:** Create an AND relationship between "Collect available dates" and "Meeting organized": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on "And" button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Connect "Collect available dates" and "Meeting Participant": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
4. **Action 2:** Click on "Collect available dates" first and then on "Meeting Participant".
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: "Add an AND relationship between child and parent"

1. **Subgoal 1:** Create an AND relationship between "Collect available dates" and "Meeting organized": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on "And" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Connect "Collect available dates" and "Meeting Participant": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
4. **Action 2:** Click on "Collect available dates" first and then on "Meeting Participant".
 - a) Will Tim know what to do?

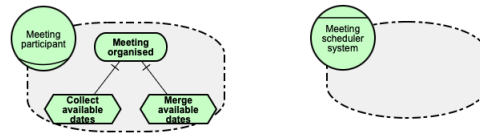


Figure 4.4: Illustration depicting the envisioned model at the time of creating the AND relationship between the children “Collect available dates” and “Merge available dates” to the parent “Meeting Participant”.

- i. **Yes/no/maybe**
- b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

This connectivity process would be repeated for the child “Merge available dates”. At this point, our working board should look exactly like the one shown in Figure 4.4.

Following Figure 4.3, the next steps follow the the same exact steps as before. There are two more tasks to be created named “Manually merge dates” and “Merge dates using scheduler”. Lets consider the items created and their respective names edited, similar as before. Another item to be created is a resource named “Available dates”. This process is very similar to the ones described before, however, let’s exemplify this.

GenderMag’ing with Abi: “Add a resource to the role’s SR”

1. **Subgoal 1:** Create a new resource item: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on “Resource” button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Put resource item inside “Meeting Participant”: Will Abi have formed this subgoal...?

a) **Yes**/no/maybe

4. **Action 2:** Click inside the "Meeting Participant" boundary.

a) Will Abi know what to do?

i. **Yes**/no/maybe

b) If action... will Abi see progress to subgoal?

i. **Yes**/no/maybe

GenderMag'ing with Tim: "Add a resource to the role's SR"

1. **Subgoal 1:** Create a new resource item: Will Tim have formed this subgoal...?

a) **Yes**/no/maybe.

2. **Action 1:** Click on "Resource" button.

a) Will Tim know what to do?

i. **Yes**/no/maybe

b) If action... will Tim see progress to subgoal?

i. **Yes**/no/maybe

3. **Subgoal 2:** Put resource item inside "Meeting Participant": Will Tim have formed this subgoal...?

a) **Yes**/no/maybe

4. **Action 2:** Click inside the "Meeting Participant" boundary.

a) Will Tim know what to do?

i. **Yes**/no/maybe

b) If action... will Tim see progress to subgoal?

i. **Yes**/no/maybe

Having these three new items created, two tasks and one resource, the next thing to do would be to connect them with other items, with the right relationships. The two tasks "Manually merge dates" and "Merge dates using scheduler" are both connected to their parent "Merge available dates" through an OR relationship. Let's create the process of connecting "Manually merge dates" and its parent "Merge available dates", which allows us to omit the process for "Merge dates using scheduler" since it would go the exact same way.

GenderMag'ing with Abi: "Add an OR relationship between child and parent"

1. **Subgoal 1:** Create an OR relationship between "Manually merge dates" and "Merge available dates": Will Abi have formed this subgoal...?

- a) **Yes/no/maybe.**
- 2. **Action 1:** Click on "Or" button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Connect "Manually merge dates" and "Merge available dates": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click on "Manually merge dates" first and then on "Merge available dates".
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: "Add an OR relationship between child and parent"

- 1. **Subgoal 1:** Create an OR relationship between "Manually merge dates" and "Merge available dates": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 2. **Action 1:** Click on "Or" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Connect "Manually merge dates" and "Merge available dates": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
- 4. **Action 2:** Click on "Manually merge dates" first and then on "Merge available dates".
 - a) Will Tim know what to do?

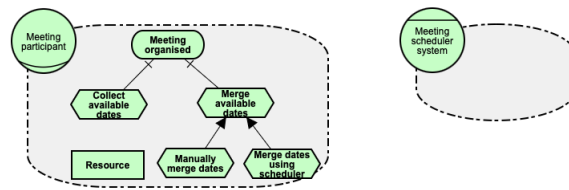


Figure 4.5: Illustration depicting the envisioned model at the time of creating the OR relationship between the children “Manually merge dates” and “Merge dates using scheduler” to their parent “Merge available dates”.

- i. **Yes**/no/maybe
- b) If action... will Tim see progress to subgoal?
 - i. **Yes**/no/maybe

At this point our working board should be looking similar to the one displayed in **Figure 4.5**.

To finish this level of the role’s SR we only need to create a “needed-by” relationship between the resource “Available dates” and the task “Collect available dates”.

GenderMag’ing with Abi: “Add a needed-by relationship between child and parent”

1. **Subgoal 1:** Create a needed-by relationship between “Manually merge dates” and “Merge available dates”: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe.
2. **Action 1:** Click on “Needed-by” button.
 - a) Will Abi know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
3. **Subgoal 2:** Connect “Available dates” and “Collect available dates”: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe

4. **Action 2:** Click on "Available dates" first and then on "Collect available dates".

- a) Will Abi know what to do?
 - i. Yes/no/maybe
- b) If action... will Abi see progress to subgoal?
 - i. Yes/no/maybe

GenderMag'ing with Tim: "Add a needed-by relationship between child and parent"

1. **Subgoal 1:** Create a needed-by relationship between "Manually merge dates" and "Merge available dates": Will Tim have formed this subgoal...?

- a) Yes/no/maybe.

2. **Action 1:** Click on "Needed-by" button.

- a) Will Tim know what to do?
 - i. Yes/no/maybe
- b) If action... will Tim see progress to subgoal?
 - i. Yes/no/maybe

3. **Subgoal 2:** Connect "Available dates" and "Collect available dates": Will Tim have formed this subgoal...?

- a) Yes/no/maybe

4. **Action 2:** Click on "Available dates" first and then on "Collect available dates".

- a) Will Tim know what to do?
 - i. Yes/no/maybe
- b) If action... will Tim see progress to subgoal?
 - i. Yes/no/maybe

Finally, we need to add quality named "Minimal effort" do the role's SR, that later will be linked to the task of "Merge available dates."

GenderMag'ing with Abi: "Add a quality to the role's SR"

1. **Subgoal 1:** Create a new quality item: Will Abi have formed this subgoal...?

- a) Yes/no/maybe.

2. **Action 1:** Click on "Quality" button.

- a) Will Abi know what to do?
 - i. Yes/no/maybe

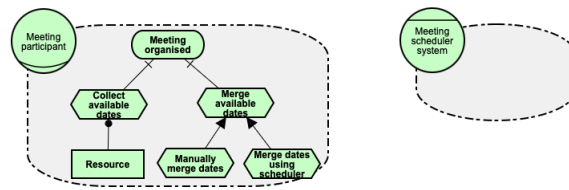


Figure 4.6: Illustration depicting the envisioned model at the time of creating the needed-by relationship between the children “Available dates” to its parent “Collect available dates”.

- b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe
- 3. **Subgoal 2:** Put quality item inside “Meeting Participant”: Will Abi have formed this subgoal...?
 - a) **Yes**/no/maybe
- 4. **Action 2:** Click inside the “Meeting Participant” boundary.
 - a) Will Abi know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes**/no/maybe

GenderMag’ing with Tim: “Add a quality to the role’s SR”

- 1. **Subgoal 1:** Create a new quality item: Will Tim have formed this subgoal...?
 - a) **Yes**/no/maybe.
- 2. **Action 1:** Click on “Quality” button.
 - a) Will Tim know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes**/no/maybe

3. **Subgoal 2:** Put quality item inside "Meeting Participant": Will Tim have formed this subgoal...?

a) **Yes/no/maybe**

4. **Action 2:** Click inside the "Meeting Participant" boundary.

a) Will Tim know what to do?

i. **Yes/no/maybe**

b) If action... will Tim see progress to subgoal?

i. **Yes/no/maybe**

And as we have been doing, let's consider its named changed to "Minimal effort" through the same name editing process as before. This quality will logically be influenced by both child tasks stemming from this parent task, as one will have a negative impact while the other will have a positive one. To reach this point, firstly we need to create a qualification relationship between the task "Merge available dates" and the quality we just created.

GenderMag'ing with Abi: "Add a qualification relationship between a task and a quality"

1. **Subgoal 1:** Create a qualification relationship between "Merge available dates" and "Minimal effort": Will Abi have formed this subgoal...?

a) **Yes/no/maybe.**

2. **Action 1:** Click on "Qualification" button.

a) Will Abi know what to do?

i. **Yes/no/maybe**

b) If action... will Abi see progress to subgoal?

i. **Yes/no/maybe**

3. **Subgoal 2:** Connect "Merge available dates" and "Minimal effort": Will Abi have formed this subgoal...?

a) **Yes/no/maybe**

4. **Action 2:** Click on "Minimal effort" and then on "Merge available dates".

a) Will Abi know what to do?

i. **Yes/no/maybe**

b) If action... will Abi see progress to subgoal?

i. **Yes/no/maybe**

GenderMag'ing with Tim: "Add a qualification relationship between a task and a quality"

1. **Subgoal 1:** Create a qualification relationship between "Merge available dates" and "Minimal effort": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on "Qualification" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Connect "Merge available dates" and "Minimal effort": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
4. **Action 2:** Click on "Minimal effort" and then on "Merge available dates".
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

After this, the only thing left to do is to create the respective contribution links between the quality and its contributor. Let's exemplify for the "help" link between "Merge dates using scheduler" and "Minimal effort" and consider the same exact pattern for the "hurt" link between "Manually merge dates" and "Minimal effort", with the only difference being the button clicked.

GenderMag'ing with Abi: "Add a "help" contribution link between "Merge dates using scheduler" and "Minimal effort"

1. **Subgoal 1:** Create a contribution link between "Merge dates using scheduler" and "Minimal effort": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on "Contribution" button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?

- i. **Yes/no/maybe**
- 3. **Subgoal 2:** Create a “help” contribution link between “Merge dates using scheduler” and “Minimal effort”: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 4. **Action 2:** Click on “help (+)” button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 5. **Subgoal 3:** Connect “Merge dates using scheduler” and “Minimal effort”: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 6. **Action 3:** Click on “Merge dates using scheduler” and “Minimal effort”.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: “Add a “help” contribution link between “Merge dates using scheduler” and “Minimal effort””

- 1. **Subgoal 1:** Create a contribution link between “Merge dates using scheduler” and “Minimal effort”: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 2. **Action 1:** Click on “Contribution” button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Create a “help” contribution link between “Merge dates using scheduler” and “Minimal effort”: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**

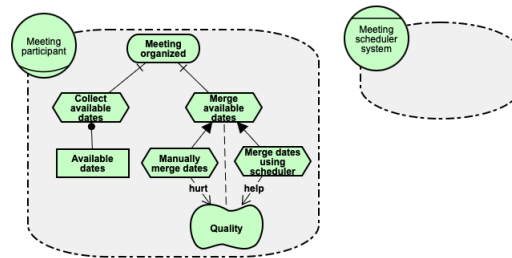


Figure 4.7: Illustration depicting the envisioned model at the time of creating the contribution links between “Merge dates using scheduler” and “Manually merge dates” with the quality “Minimal effort”.

4. **Action 2:** Click on “help (+)” button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
5. **Subgoal 3:** Connect “Merge dates using scheduler” and “Minimal effort”: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
6. **Action 3:** Click on “Merge dates using scheduler” and “Minimal effort”.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

Having done all these steps, our model should be similar to the one displayed on **Figure 4.3**. We are now ready to jump to the other branch we created earlier.

The operations required to build this right side of the model are similar to the ones already described. By processes like the ones above, we would create a goal named “Meeting scheduled”, and three tasks named “Merge available dates”, “Obtain available dates” and “Obtain agreement”. Then, we need to establish their relationship among one

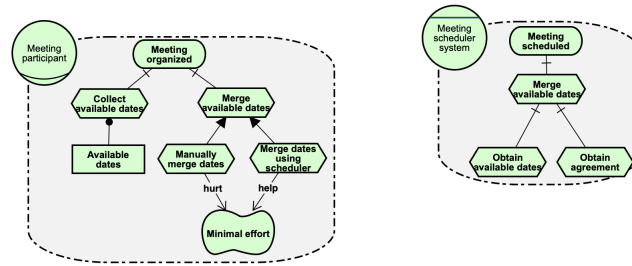


Figure 4.8: Illustration depicting the envisioned model at the time of creating the right side of the model.

another. Three “And” relationships would have to be created, one from “Merge available dates” to Meeting scheduled, while the other two would be from “Obtain available dates” and from “Obtain agreement” to “Merge available dates”, making this one the parent item of the two mentioned before. At this point, our model should look similar to **Figure 4.8**

Finally, to complete this model we would need to link the role and agent we created through a dependency connecting to items inside their SR. In order to do this, we need to create a resource dependency between the task “Obtain available dates” and “Merge dates using scheduler”, to indicate that the task “Obtain ”

GenderMag’ing with Abi: “Add a resource dependency between “Obtain available dates” and “Merge dates using scheduler””

1. **Subgoal 1:** Create a dependency between “Obtain available dates” and “Merge dates using scheduler”: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on “Depedency” button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Create a resource dependency between “Obtain available dates” and “Merge dates using scheduler”: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**

4. **Action 2:** Click on "Resource dependency" button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
5. **Subgoal 3:** Connect "Obtain available dates" and "Merge dates using scheduler": Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
6. **Action 3:** Click on "Obtain available dates" and then "Merge dates using scheduler".
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: "Add a resource dependency between "Obtain available dates" and "Merge dates using scheduler""

1. **Subgoal 1:** Create a dependency between "Obtain available dates" and "Merge dates using scheduler": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
2. **Action 1:** Click on "Depedency" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
3. **Subgoal 2:** Create a resource dependency between "Obtain available dates" and "Merge dates using scheduler": Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
4. **Action 2:** Click on "Resource dependency" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?

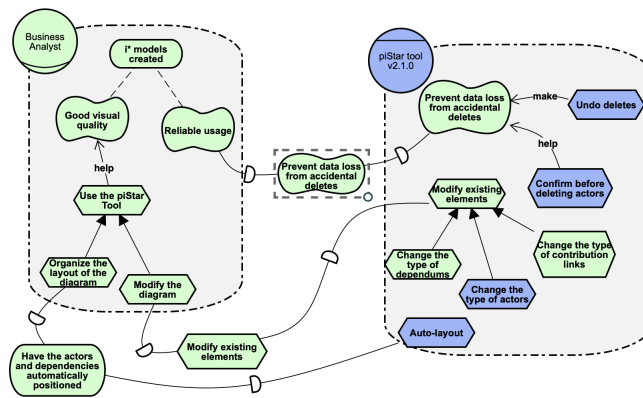


Figure 4.9: Illustration of the model to be uploaded into the tool to test the process of comparing more than two item's attributes.

i. **Yes/no/maybe**

5. **Subgoal 3:** Connect “Obtain available dates” and “Merge dates using scheduler”:
Will Tim have formed this subgoal...?

a) **Yes/no/maybe**

6. **Action 3:** Click on “Obtain available dates” and then “Merge dates using scheduler”.

a) Will Tim know what to do?

i. **Yes/no/maybe**

b) If action... will Tim see progress to subgoal?

i. **Yes/no/maybe**

An interesting scenario to test would be the one of uploading a model and having the personas compare attributes about two or more items. For this, let's consider the model represented in **Figure 4.9**. We'll test a scenario where a user uploads this model and examines the attribute information of three items. This testing stems from the previously identified absence of multi-selection support for actions like deletion, dragging and comparing. Since comparing attributes of just two items might be too straightforward for this scenario, we'll focus on comparing information from three items. We'll begin by testing the upload process and then move on to the comparison phase.

GenderMag'ing with Abi: “Upload a model”

1. **Subgoal 1:** View file management options: Will Abi have formed this subgoal...?

a) **Yes/no/maybe**.

2. **Action 1:** Click on “File” button.

- a) Will Abi know what to do?
 - i. **Yes/no/maybe**
- b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Choose file to be uploaded: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 4. **Action 2:** Click on "Choose File" button.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 5. **Subgoal 3:** Confirm file upload: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe**
- 6. **Action 3:** Click on "Load model".
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**

GenderMag'ing with Tim: "Upload a model"

- 1. **Subgoal 1:** View file management options: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 2. **Action 1:** Click on "File" button.
 - a) Will Tim know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Choose file to be uploaded: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 4. **Action 2:** Click on "Choose File" button.

- a) Will Tim know what to do?
 - i. **Yes/no/maybe**
- b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**
- 5. **Subgoal 3:** Confirm file upload: Will Tim have formed this subgoal...?
 - a) **Yes/no/maybe**
- 6. **Action 3:** Click on "Load model".

- a) Will Tim know what to do?
 - i. **Yes/no/maybe**
- b) If action... will Tim see progress to subgoal?
 - i. **Yes/no/maybe**

Having uploaded the selected file and model resembling the one in **Figure 4.9**, the only thing left to test is the comparing scenario mentioned before.

GenderMag'ing with Abi: "Analyze "Auto-layout" attributes and style"

- 1. **Subgoal 1:** Analyze "Auto-layout" task: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 2. **Action 1:** Click on "Auto-layout" task.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 3. **Subgoal 2:** Analyze "Auto-layout"'s attributes: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**
- 4. **Action 2:** Read through the item's properties, name and description.
 - a) Will Abi know what to do?
 - i. **Yes/no/maybe**
 - b) If action... will Abi see progress to subgoal?
 - i. **Yes/no/maybe**
- 5. **Subgoal 3:** Analyze "Auto-layout"'s style: Will Abi have formed this subgoal...?
 - a) **Yes/no/maybe.**

6. **Action 3a:** Click on the "Style" tab.

- a) Will Abi know what to do?
 - i. Yes/no/maybe
- b) If action... will Abi see progress to subgoal?
 - i. Yes/no/maybe

7. **Action 3b:** Analyze the item's style.

- a) Will Abi know what to do?
 - i. Yes/no/maybe
- b) If action... will Abi see progress to subgoal?
 - i. Yes/no/maybe

GenderMag'ing with Tim: "Analyze "Auto-layout" attributes and style"

1. **Subgoal 1:** Analyze "Auto-layout" task: Will Tim have formed this subgoal...?

- a) Yes/no/maybe.

2. **Action 1:** Click on "Auto-layout" task.

- a) Will Tim know what to do?
 - i. Yes/no/maybe
- b) If action... will Tim see progress to subgoal?
 - i. Yes/no/maybe

3. **Subgoal 2:** Analyze "Auto-layout"'s attributes: Will Tim have formed this subgoal...?

- a) Yes/no/maybe.

4. **Action 2:** Read through the item's properties, name and description.

- a) Will Tim know what to do?
 - i. Yes/no/maybe
- b) If action... will Tim see progress to subgoal?
 - i. Yes/no/maybe

5. **Subgoal 3:** Analyze "Auto-layout"'s style: Will Tim have formed this subgoal...?

- a) Yes/no/maybe.

6. **Action 3a:** Click on the "Style" tab.

- a) Will Tim know what to do?

- i. **Yes**/no/maybe
- b) If action... will Tim see progress to subgoal?
 - i. **Yes**/no/maybe
- 7. **Action 3b**: Analyze the item's style.
 - a) Will Tim know what to do?
 - i. **Yes**/no/maybe
 - b) If action... will Tim see progress to subgoal?
 - i. **Yes**/no/maybe

Taking this pattern as standard for three items, "Auto-layout", "Change the type of contribution links" and "Change the type of actors", we can imagine that our personas, both Abi and Tim would have no problems whatsoever in analyzing any item's properties and style. However, the lack of a multi-selection tool does not pose a potential problem to a scenario where the users are iteratively analyzing information, but rather when the users have to compare multiple information simultaneously.

On the left side of the interface, as we can see on **Figure 2.4**, there is a dedicated column displaying the properties of the item currently selected with the mouse. This column exclusively shows the properties of the selected item, with the displayed information dynamically changing based on the item's selection and its required property categories. The column displays a single set of information at any given time. When a different item is chosen, the previously viewed information vanishes to accommodate the newly selected item's properties. This functionality mirrors a design change category mentioned in Chapter 3, Section 3.1.12, that should take place if the user can open a tooltip with information, for example, but it would disappear once they click something else. Such a design could complicate the process of comparing and analyzing information where it was initially required. Furthermore, in the case of this example in 3.1.12, it is impossible to compare two or more sets of information simultaneously, as the design does not allow for multiple selections or viewing more than one tooltip at a time. Specifically, the developer using the piStar tool [46], would not be able to easily compare the properties of two or more elements at a time. It is also important to notice that fixing this, allowing the user to benefit from multi-selection for various purposes, would, in part, help solve the issue mentioned in the very first GenderMag'ing session: "Clear piStar working board". The issue, in this instance, was that Abi would have to grasp the idea that she could not select all items to be erased, but rather select the items one at a time and delete them individually.

We are now ready to elaborate a prioritized list of potential problems to be resolved. Through the process of analyzing our potential problems, in this case, our "maybe" options, we observe that they primarily arise from either shaping the intended subgoal or determining the subsequent steps post-goal establishment. Consequently, the tool in question consistently demonstrated no issues regarding users' perception of progress after

executing an action. In total, there were six "maybe" outcomes: two stemmed from establishing the intended subgoals, while four were related to Abi's understanding/following of subsequent actions.

1. Comparison of multiple sets of information simultaneously - lack of multi-selection and multi-display of information.
2. Deletion of multiple or all items to clear working board - lack of multi-selection.
3. Deletion of an item - "Delete" button on property column may go unnoticed.
4. Confirming to a seemingly permanent alteration - pop-up alerts confirming action with no indication as to how to undo the action may demotivate users with high risk aversion.
5. Editing fields not obvious - items look more like a hyperlink than editable fields.

4.1.2 GenderMag application: sessions with João Pimentel

During the application of the GenderMag method, we had the privilege of collaborating with João Pimentel, the developer of the piStar tool [46]. With João's assistance, we not only validated the application of the method but also engaged in brainstorming sessions to explore potential improvements and their effects on the persona facets. These discussions were further enriched by the presence of Professor João Araújo and Professor Miguel Goulão, who contributed both during our meetings with João Pimentel and in subsequent deliberations. This collaborative effort provided the support and experienced insight necessary to make informed decisions and advance the project.

The objectives of both meetings were to showcase the GenderMag application on the piStar tool, discuss the assumptions made during the method's application, review the results, and consider suggestions. The meetings began with an overview of the method application and the preliminary "maybe" results, including their origins, rationale, and potential solutions.

Although involving the developer was crucial for applying the method effectively, the most valuable outcome of the sessions was the brainstorming and the list of ideas discussed and proposed by João Pimentel. This process closely mirrored the "GenderMag moments" described in Chapter 2, subsection 2.1.1.3. GenderMag moments are segments of development sessions where the team deliberates on design decisions in real-time, considering the facet values. These moments create a collaborative environment for discussing potential solutions and ideas, thereby reducing the need for extensive GenderMag sessions and making it easier to integrate the method into existing software development practices.

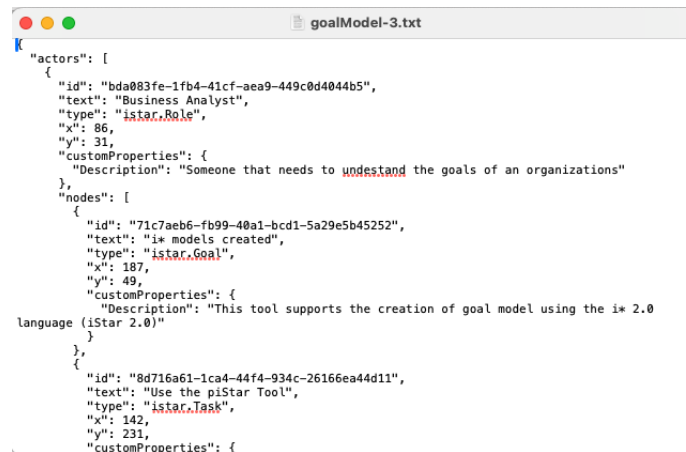
In the sequence of this GenderMag moment, the suggestions made by the tool's developer were all in the realm of either fixing the issues that were brought up or making

the tool even more inclusive. Specifically, in the alert pop-up messages in **Figures 4.13, 4.14, 4.15**, the developer's suggestion was to make these message boxes even more intuitive and user-friendly by creating "OK" and "Cancel" buttons that described the actual action the user would be confirming or canceling. So, for example, as per the **Figure 4.13**, the message would be fixed to "ATTENTION! Are you sure you want to delete this entire actor, along with its content? TIP: You can undo this action using the "Go Back" button.", as it will be described in depth in Section 4.2 about the implementation, the confirming button would now be "OK, delete the entire actor and content." and the canceling button would be "Cancel, do not delete entire actor and content". And the same idea would be applied to the other two alert scenarios. After further discussion, despite this being a potential good solution to implement on this tool in the future, it does not align with the current scope of our project. The buttons "OK" and "Cancel" are commonly used in various systems, such as Office applications, which we know our personas have at least some experience with. Therefore, we have decided not to pursue the implementation of this suggestion at this time.

João Pimentel strongly supported the suggestion to enhance the delete button. While having multiple deletion options is beneficial, having these options does not guarantee that first-time users will easily discover or understand these features. This discussion underscores the importance of providing users with a clear and straightforward path to follow.

The other topic discussed was the lack of multi-selection. The developer agreed that the implementation of multi-selection would in fact make the developer experience better but did not think the multi-selection along with multi-display would be the best way to solve the action of comparing multiple pieces of information, due to the logistics of opening various property boxes in the side-panel which could make the working area polluted and confusing. However, alternative solutions were proposed to deal with this matter. One suggestion was to make the model items, that by default display its name, display the description or other properties, making it possible for the user to switch between the properties and compare them in the model. The other proposed solution was to have a feature where the user could export a kind of a report that would make it possible to compare the information about the items on the model. After further deliberation, we determined that generating a report from the tool would be the most efficient approach. When a model is saved, a text (.txt) file is downloaded, similar to the one shown in **Figure 4.10**. The plan is to generate something similar, filter the necessary information, and present it in a more user-friendly format.

Upon further consideration, we concluded that to fully implement the multi-selection feature, we needed to include both drag-selection and multi-select functionality using the "command" or "ctrl" key, aligning with standard system practices. This dual approach enables users to easily compare items from different areas of the interface, which would be difficult using drag-selection alone. Additionally, the original concept of enabling users to generate a general report, as discussed in previous meetings, was found to be



```
"actors": [
  {
    "id": "bda083fe-1fb4-41cf-aea9-449c0d4044b5",
    "text": "Business Analyst",
    "type": "iStar.Role",
    "x": 86,
    "y": 31,
    "customProperties": {
      "Description": "Someone that needs to understand the goals of an organizations"
    }
  },
  "nodes": [
    {
      "id": "71c7aeb6-fb99-40a1-bcd1-5a29e5b45252",
      "text": "i* models created",
      "type": "iStar.Goal",
      "x": 187,
      "y": 49,
      "customProperties": {
        "Description": "This tool supports the creation of goal model using the i* 2.0"
      }
    },
    {
      "id": "8d716a61-1ca4-44f4-934c-26166ea44d11",
      "text": "Use the piStar Tool",
      "type": "iStar.Task",
      "x": 142,
      "y": 231,
      "customProperties": {
```

Figure 4.10: Snippet of the text file saved when saving a model on the piStar tool [46].

lacking in certain aspects. Specifically, for users who share Abi’s characteristics, the need to compare all and only the relevant information is crucial. However, the general report was designed to present all available information for each item, which would make it challenging for these users to isolate and focus on the specific details they need, as mentioned on subsection 3.1.12. While this feature can also be useful in some scenarios, the best option would be to also allow the user to use this feature but only for selected items.

Lastly, it was suggested that maybe it would be a good idea that users, when opening the interface for the first time, would be greeted with a timed toast with a button to clear the working board. This toast might help, to some degree, the potential confusion associated to clearing the working board completely. And although this might be too much information too soon for Abi, Abi still has the option not to click and follow the steps as presented earlier, but it could potentially make the overall interface experience better even for experienced users. Even so, this last suggestion does not quite fit the overall goal of this work, since we hope to be working towards a more gender inclusive experience of the piStar tool [46].

4.2 Implementation

In order to identify and resolve each potential inclusivity issue, it is necessary to refer to the GenderMag Design Catalog [23] and its proposed design changes, mentioned on Chapter 3, whenever possible. This involves determining the category under which each specific issue falls and then creating a plan to address and rectify it accordingly.

This section describes the implementation of the fixes to the inclusivity issues identified through the application of the GenderMag method. The primary changes were made to the “index.html”, “ui.js”, “propertiesTable.js” and “fileManager.js” files of the piStar tool. These modifications were designed to address the identified problems, such as the ambiguous delete button, the need for multi-selection and multi-display of information,

and the enhancement of confirmation dialogues to be more user-friendly, particularly for users with higher risk aversion.

During the initial exploration and implementation process, it became clear that to efficiently and accurately apply the intended design changes, the tool's overall structure would need modification. While the original structure allowed for some adaptations, enabling multi-selection within the existing code, further restructuring would be required for production-level quality. However, for the purposes of this work, the code was adapted to produce a reliable prototype that closely mirrors the original tool with the proposed fixes, though it lacks the styles feature since it was not compatible with the code adaptations. As a result, users cannot change the color of cells in this prototype, a feature that is not relevant to the planned tests.

The code adaptation did present challenges. While full multi-selection was intended to include both drag-selection and control via the “command” and “ctrl” keys, this was not fully achievable. Nevertheless, the implementation phase successfully produced a prototype² encompassing all the proposed features, except this one, allowing us to proceed to the testing phase.

4.2.1 Deletion of an item

This particular issue arose when we tried GenderMag'ing the task of deleting items such as actors, links, and dependencies. The problem was that the "Delete" button on the side panel wasn't prominently highlighted, causing the evaluation team to question whether they should use the keyboard shortcut "Delete". The concern is that users unfamiliar with this interface might overlook the small, monochromatic button. An ever so slightly more attention-grabbing button could alleviate this uncertainty and improve user experience.

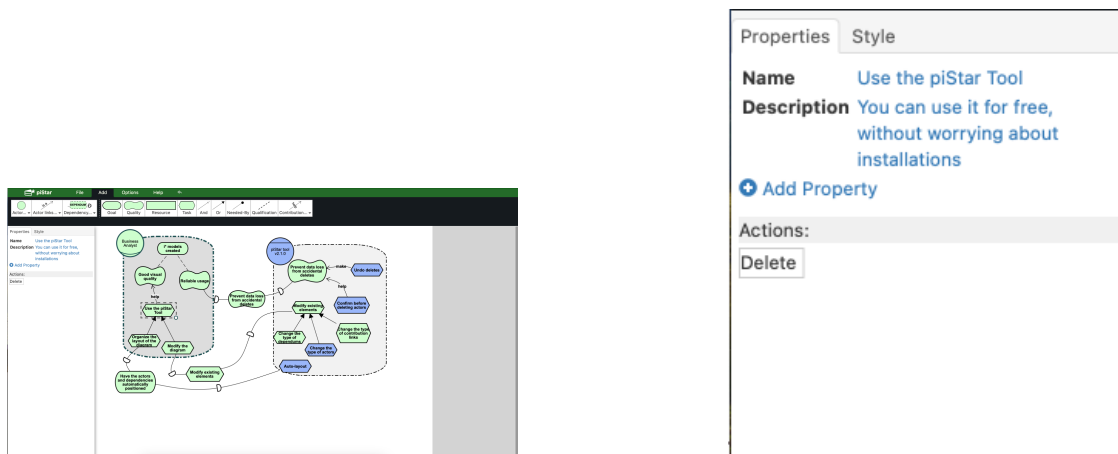


Figure 4.11: Screenshot of the piStar [46] interface, with a highlight on the “Delete” button before the implementation of a fix for the potential inclusivity bug. On the right, we can a zoomed-in version.

²https://github.com/DeboradDias/piStar_GenderMag

On **Figure 4.11** we can see how the small and monochromatic button is barely noticeable amidst the side panel. **Figure 4.12** shows the revised version aimed at drawing more attention. The goal is to make this button stand out, ensuring that first-time users notice it even if only briefly. This brief moment of attention could help users remember the button for future tasks. When specifically thinking about users that align with Abi's comprehensive information processing style, we know they are likely to gather information before proceeding to execute the task. By aiming to make this button more noticeable, the goal is that Abis will quickly access the interface and see the "Delete" button in the side panel, which can mitigate future confusion.

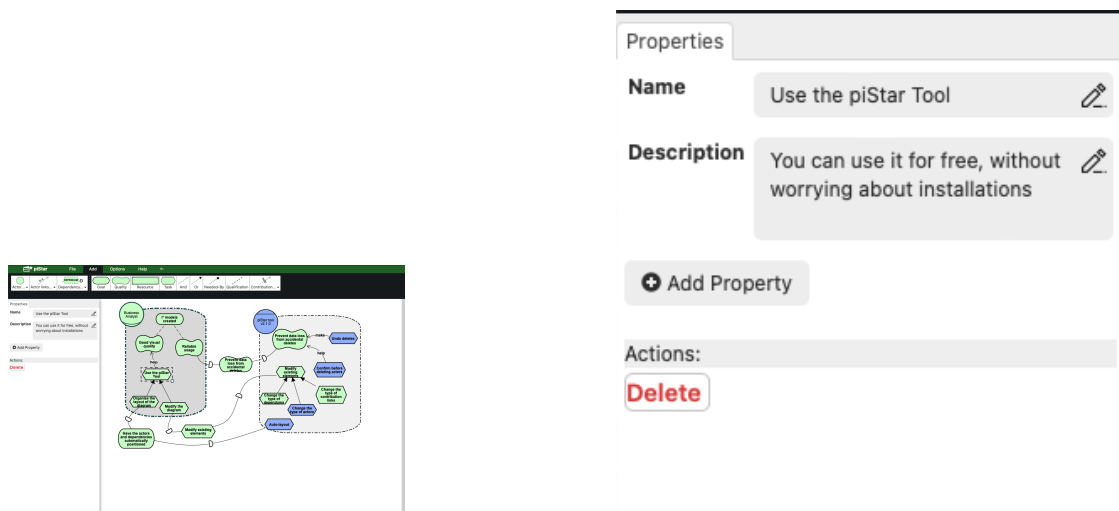


Figure 4.12: Screenshot of the piStar [46] interface, with a highlight on the "Delete" button after the implementation of a fix for the potential inclusivity bug. On the right, we can a zoomed-in version.

When comparing **Figure 4.11** and **Figure 4.12**, we can see the actual differences between the two buttons. On **Figure 4.12** we can see that the lettering is now not only bigger, but red as well, which generally is correlated with removal/deletion actions.

We classify this issue under the "Ambiguous Option" category of the GenderMag Method Catalog's design change categories. Users may be unsure whether to delete an item using the keyboard's delete key, the side-panel delete button, or the backspace key. By making the side-panel delete button more prominent, we aim at providing users with a clear and straightforward option for deleting items.

It is also relevant to mention that to address this, the 'index.html' file was updated to include a more prominent delete button. The changes involved modifying the CSS to increase the button size and changing its color to red, as mentioned above, a universally recognized color for deletion actions. The updated button now stands out clearly on the side panel, aiming to making it easier for users, especially those with a comprehensive information processing style, to locate and use this feature.

4.2.2 Confirming to a seemingly permanent alteration

This issue arose in multiple GenderMag sessions. It is standard practice for software systems to provide user alerts before completing specific actions, particularly those like deletions or replacements. These alerts are crucial, especially when such actions are irreversible or could result in the inadvertent loss of the user's current work. This ensures that users are informed and can make conscious decisions before proceeding, thereby enhancing the overall usability and reliability of the software tool.

However, as discussed in Section 2.1.1, different personas exhibit varying values across the five facets assessed by the GenderMag method. Notably, one of these facets is risk-aversion. Considering this, it is plausible to infer that alert messages solely cautioning about the risk of confirming an action, without providing guidance on preparatory steps, could potentially deter a risk-averse user from completing a task. At the very least, it might prolong the task duration as the user seeks additional information before proceeding with the action.

The overall improvements made to all three alert messages below involve providing users with a "TIP" a practice already utilized extensively throughout the tool. This approach ensures that tips are not limited on the interface but are instead consistently integrated across the tool, almost creating an expectation for users to receive guidance during critical or complex actions.

The first scenario where an alert was shown was when the GenderMag session included deleting an actor with content, meaning more items inside the actor. The message was "ATTENTION! Are you sure you want to delete this entire actor, along with its content?" as we can see on **Figure 4.13**.



Figure 4.13: Screenshot of the piStar [46] interface, with a highlight on the alert message for deleting an actor. On the left, we can see the alert message before the implementation of a fix for the potential inclusivity bug. On the right, we can see after the implementation of the fix.

In Figure 4.13 on the left, there is no indication provided on how to undo the action if the outcome does not align with expectations. Despite this, there exists a method for undoing actions. The revised message, on the right, "ATTENTION! Are you sure you want to delete this entire actor, along with its content? TIP: You can undo this action using the "Go Back" button." aims to reassure users by informing them of the option to undo the action if it does not meet their expectations, thereby enhancing user experience.

The second instance where an alert was displayed occurred during a session involving the opening of a new model at the expense of the currently open one. The message presented, as seen on **Figure 4.14** on the image on the left, "Are you sure you want to create a new model and delete the current model? This action is irreversible." failed to

offer any instructions on how users could safeguard their work before proceeding with the action.

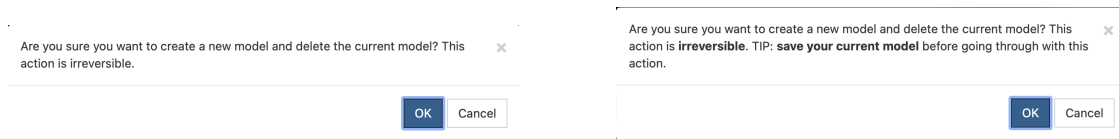


Figure 4.14: Screenshot of the piStar [46] interface, with a highlight on the alert message for creating a new model. On the left, we can see the alert message before the implementation of a fix for the potential inclusivity bug. On the right, we can see after the implementation of the fix.

On Figure 4.14 on the right, we can see a revised version of this message, where the aim is to remind the user that they can safely go through with this action if they save their current model first. The goal is to make the user feel at ease with the interface and its features.

The third scenario where a message like the ones previously shown was displayed was when the session included loading example models. On Figure 4.15 on the left, we can see the message “WARNING! If you click on an example below, your current model will be erased. This action is irreversible.”. This message aims at warning the user that if they choose to load any of the example models, than their current model will be discarded, which is irreversible. However, this message lacks in guidance on how to proceed if the goal is to both continue working on their model and also check a model example.

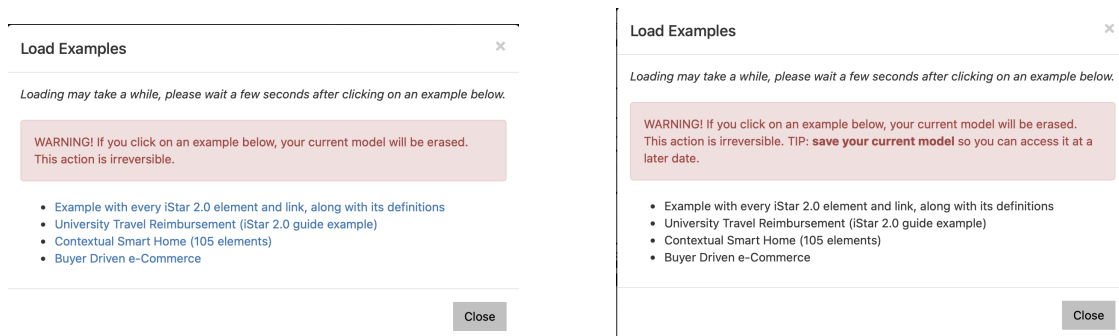


Figure 4.15: Screenshot of the piStar [46] interface, with a highlight on the alert message for loading example models. On the left, we can see the alert message before the implementation of a fix for the potential inclusivity bug. On the right, we can see after the implementation of the fix.

As in the other two scenarios mentioned before, the strategy here is to give the user a clear path to follow in case they do want to go through with the action of loading an example model, but also want to keep working on their model. The new message, displayed on Figure 4.15 on the right, is “WARNING! If you click on an example below, your current model will be erased. This action is irreversible. TIP: save your current model so you can access it at a later date.”.

This potential inactivity bug fix can perhaps fit into the category of “Incomplete

instructions” mentioned in Chapter 3. This design change should occur when instructions to continue following a specific set of steps are not organized or specific enough, which can deter a task-oriented user from completing the task since they would not explore around. This category’s aim is to help users keep going and completing their task by add instructions for all common situations.

To make the design changes mentioned above, the confirmation dialogues in the “ui.js” file were updated to include more informative messages. These messages now include “TIP” sections that provide users with instructions on how to backtrack or save their progress before proceeding. For instance, before opening a new model, users are now prompted with a message suggesting that they save their current work to avoid loss of data.

4.2.3 Editing fields not obvious

This potential inclusivity bug lies in the fact that the name editing field does not look like an editable field but rather like a link that will take the user outside of its task context. Specifically, when the user hovers their mouse on the field, then the current name on the item selected appears on the name editing field underlined by a dashed line, as depicted on Figure 4.16.

Additionally, from the categories listed and described in Chapter 3, we must determine where this particular issue fits. Although this specific scenario does not neatly fit into any category, one could say it fits into the category of “Ambiguous option”. Since the field resembles a link rather than a field and risk-averse users might find this discouraging or confusing, making them wonder where they can edit the item’s properties. Moreover, these same users can perceive a potential risk if clicking this “link” could lead them away from their current environment.

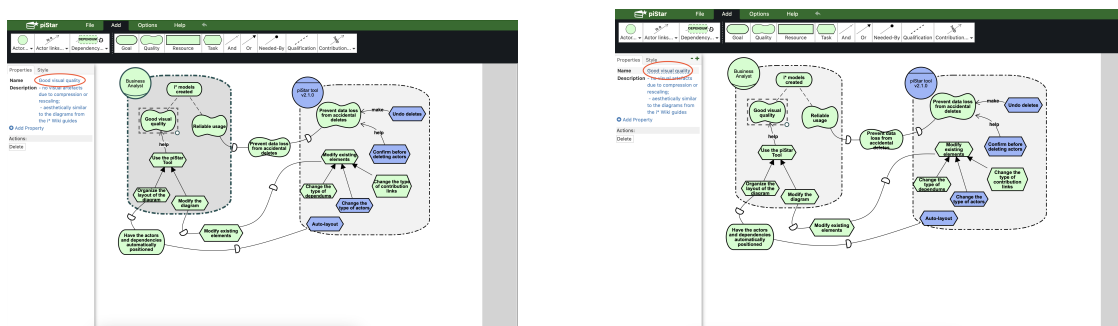


Figure 4.16: Screenshot of the piStar [46] interface, with a highlight on the name editing field. On the right, we can see the underlined field when hovered over.

To address this particular potential inclusivity issue, we can adhere to the objective of the “Ambiguous option” design change category, which aims at enhancing clarity by incorporating specific examples and detailed information. As the objective is to clearly indicate the editable field, the idea is to make the “editing” feature more obvious by using an icon that clearly shows this feature, as shown on Figure 4.17. Additionally, by not using

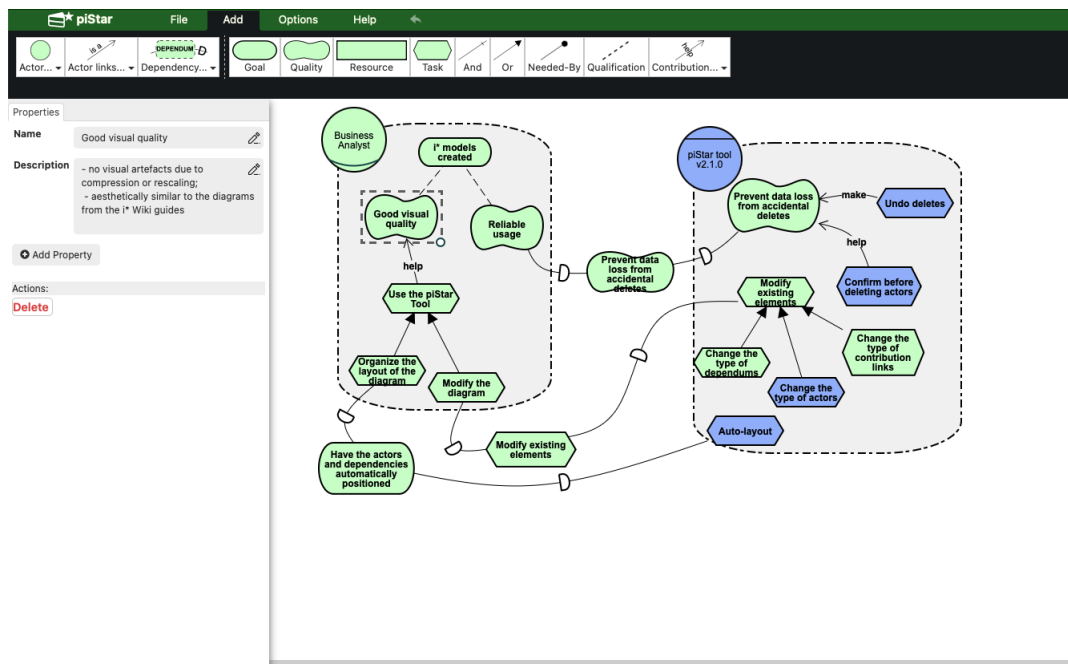


Figure 4.17: Illustration depicting the piStar [46] interface for the properties editing design changes.

blue as the color of the text, that is typically used in links, we get further and further from the link look.

To achieve the mentioned changes, the “index.html” file and its CSS was modified to include visual cues that make these fields stand out as editable elements. An edit icon was added next to these fields, and the text color was changed to a shade that distinguishes it from regular links, which helps users quickly identify editable content. Additionally, a box was added to further make the text look like a field.

4.2.4 Multi-Selection for Multi-delete and Report Generation Feature

The lack of a multi-selection feature in the piStar tool was identified as a significant barrier to comparing multiple sets of information simultaneously. To resolve this, instead of implementing a multi-display within the interface, a report generation feature was introduced. This allows users to select multiple items within the model through drag-selection, as depicted on **Figure 4.18**.

Followed by the generation of a PDF report that displays the properties of all selected items side-by-side for easy comparison, as displayed on **Figure 4.19**, while also allowing for the generation of a report about the general model as shown on **Figure 4.20**.

The multi-selection feature also facilitates the deletion of multiple items. Users can now drag-select multiple items, as on **Figure 4.18**, enabling them to access both a “Generate Report” button and a “Delete” button, on a custom side panel for multi-selection as seen on **Figure 4.21**, as well as using the delete key just as for single selection.

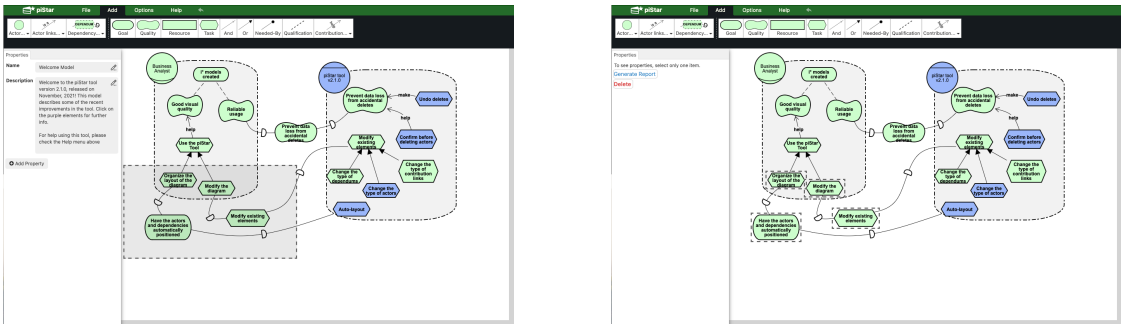


Figure 4.18: Illustration depicting the envisioned piStar [46] interface for multi-selection feature.

Name	Description	Type
Modify existing elements		Task
Have the actors and dependencies automatically positioned		Goal
Modify the diagram		Task
Organize the layout of the diagram		Task

Figure 4.19: Illustration depicting the envisioned piStar [46] interface for report feature.

This was achieved by introducing new JavaScript functions in the “ui.js” and “fileManager.js” files that handle the selection and deletion of multiple items, while also formatting the collected data into a structured PDF report. The index.html file was modified to include a button that triggers the report generation process, both in the menu bar, in the “File” tab, and the side panel, providing users with a convenient way to export and compare item properties outside the tool interface.

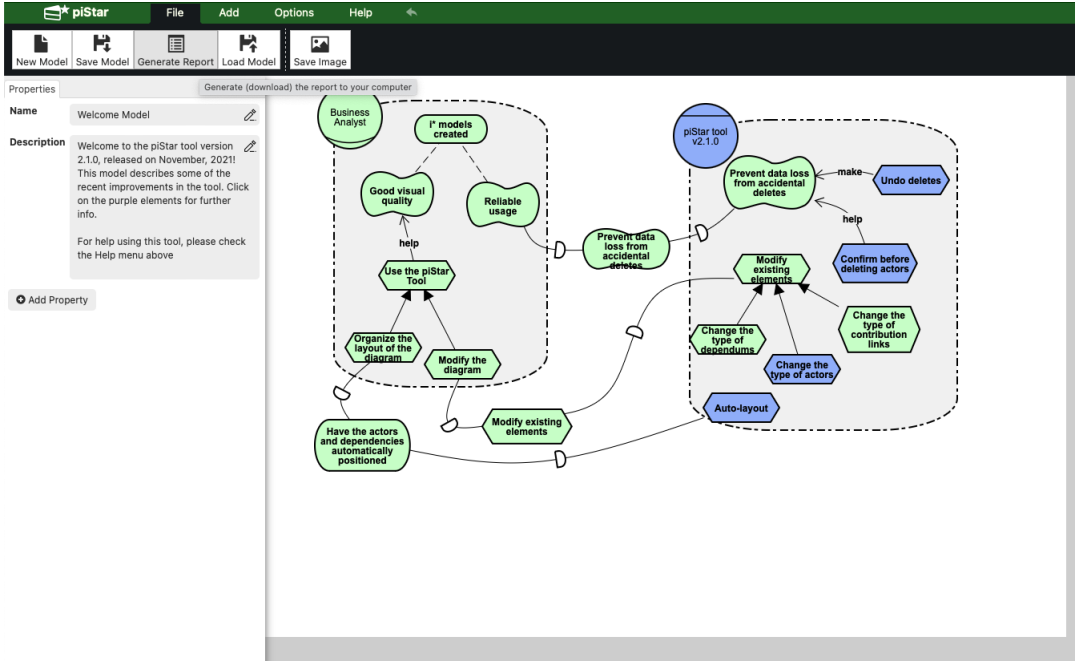


Figure 4.20: Illustration depicting the envisioned piStar [46] interface for general model report feature.

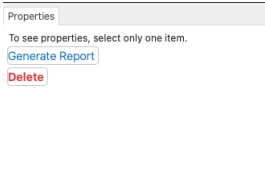


Figure 4.21: Illustration depicting the envisioned piStar [46] interface for general model report feature.

TESTING AND RESULT ANALYSIS

5.1 Experiment Planning

5.1.1 Goals

The original research objective, as outlined in the introduction, sets the foundation for the experimental study. However, to ensure clarity it is essential to refine the research goals in more concrete terms.

We aim to compare the usability and gender-inclusivity of the piStar tool [46] with a modified version of the tool, focusing specifically on their impact on software engineering tasks, particularly in the context of requirements modeling, serving a diverse population. Below is the key research goal that will inform the structure of the experimental study [16]:

- **Goal 1:** Analyze the piStar tool versus its modified version
For the purpose of understanding its effects
With respect to their usability and efficiency, from the perspective of potential diverse users engaged in software engineering tasks, specifically in performing requirements modeling tasks.

This goal is centered on evaluating the overall usability of the original piStar tool and its modified version, particularly focusing on how well they assist diverse users in completing requirements modeling tasks, which are a crucial part of software engineering workflows. We aim to gather insights into user experience, more specifically, developer experience.

By focusing on potential users and their interactions with the tools, the experiment will provide insights into practical implications for software engineering, enhancing the relevance and applicability of the findings.

5.1.2 Experiment Units

In this experiment, participants were divided into two groups, with one group assigned to test the original version of the system and the other to test the modified version of the

piStar tool. The division was carried out with the goal of getting each group to consist of an equal number of participants and an equal representation for both tested personas, Abis and Tims. The objective was to maintain similarity between the groups, so the only significant variable was the version of the tool being tested.

Participants were recruited based on specific criteria to ensure they were representative of the target user population. All selected participants had a connection to software development and held either degrees or courses in the field. As participants were recruited, they were assigned to groups using the GenderMag survey, present in **Annex I** which is a Portuguese adaptation from the original present in **Annex II**. This survey helped guide the decision on which version of the tool they would use and ensured that participants were distributed between the groups based on the goal of maintaining balance. This balance is maintained through a pragmatic approach, where the average of all facet values for each participant is calculated, assuming equal weight for each facet. This average is then compared to the median of the medians for each facet across all participants. This way we promote a balanced division of participants between the personas, Abi and Tim.

5.1.3 Experiment Material

The experiment was conducted in a controlled environment using the piStar tool, which we will refer to as Version 1, and its modified version of the tool, which will be referred to as the Version 2. A presentation about the language i* was done and a guide with steps was provided to all participants to standardize the process and ensure consistency across all experimental sessions. This guide included step-by-step instructions of the tasks to be performed, as in annexes **III** in Portuguese and **IV** in English.

5.1.4 Tasks

Participants were tasked with performing a series of activities designed to test the usability of the tool they were assigned to. The tasks were carefully selected to reflect typical user scenarios and to challenge the specific functionalities that raised potential issues as described in Chapter 4. These tasks included:

1. Step 1 – Clear the workspace: The goal is to have a blank screen so we can start working.
2. Step 2 – Create a role and edit its name to “Meeting participant.”
3. Step 3 – Add a goal (option “Goal”) within the role “Meeting participant.”
4. Step 4 – Edit its name to “Meeting organized”.
5. Step 5 – Create two tasks side by side, below “Meeting organized,” one named “Collect available dates” and the other “Merge Available dates”.

6. Step 6 – Connect these two tasks to the goal “Meeting organized” through “AND” relationships.
7. Step 7 – Create a resource named “Available dates” within the role “Meeting participant”.
8. Step 8 – Connect it to the task “Collect available dates” through a “needed-by” relationship.
9. Step 9 – Create two tasks side by side, below “Merge available dates,” one named “Manually merge dates” and the other “Merge dates using scheduler”.
10. Step 10 – Connect these two tasks to the task “Merge available dates” through “OR” relationships.
11. Step 11 – Create a quality named “Minimal effort” below these last two tasks. Step 12 – Connect the task “Manually merge dates” to the quality “Minimal effort” through a “hurt” contribution link.
12. Step 13 – Connect the task “Merge dates using scheduler” to the quality “Minimal effort” through a “help” contribution link.
13. Step 14 – Connect the task “Merge Available dates” and the quality “Minimal effort” through a qualification relationship.
14. Step 15 – Add the description “Manually merging two sets of dates” to the task “Manually merge dates”.
15. Step 16 – Add the description “Use scheduler to merge dates” to the task “Merge dates using scheduler”.
16. Step 17 – Compare and analyze the information related to three items: “Manually merge dates,” “Merge dates using scheduler,” and “Minimal effort”.
17. Step 18 – Delete four items: the resource “Available dates,” the tasks “Manually merge dates” and “Merge dates using scheduler,” and the quality “Minimal effort”.
18. Step 19 – Let’s discard this model and create a new one.

Each task was crafted to prompt specific interactions with the tool, enabling a thorough assessment of the usability of the key features. Beyond simply evaluating the tool’s basic functionalities, the tasks were designed to observe whether users would instinctively seek out or expect the new features, even in Version 1 where they were not implemented. The goal was to determine if diverse users would identify the absence of these features as limitations, mirroring the gaps that we had previously identified during the Gender-Mag’ing sessions. This approach provided insight into how intuitive and necessary these enhancements were perceived to be by the participants.

5.1.5 Hypotheses, parameters and variables

The primary hypothesis under evaluation is that the proposed solution offers superior/inferior usability compared to the original tool. Usability, in this context, is defined by several parameters, including task completion time, correctness level (which we defined as 1 if the user failed the task, 2 if the user was given a tip to complete the task and 3 if the user completed the task successfully with no help), user satisfaction, and cognitive load. These parameters were chosen because they are directly related to the ease of use and overall user experience.

1. Average Time:

- **Null Hypothesis:** There is no significant difference in the average time completing tasks when using Version 1 and Version 2.
- **Alternative Hypothesis:** Version 2 has a significantly different average time from version 1, meaning that users either complete tasks faster or slower with the modified version.

2. System Usability Scale (SUS):

- **Null Hypothesis:** There is no significant difference in the SUS scores between Version 1 and Version 2.
- **Alternative Hypothesis:** Version 2 has significantly different SUS scores from Version 1, indicating either improved or worsened perceived usability in version 2.

3. NASA Task Load Index (NASA TLX):

- **Null Hypothesis:** There is no significant difference in NASA TLX scores between Version 1 and Version 2.
- **Alternative Hypothesis:** Version 2 has significantly different NASA TLX scores from version 1, meaning users experience more or less cognitive load and task difficulty in version 2.

4. Correctness Level:

- **Null Hypothesis:** There is no significant difference in the average correctness level between Version 1 and Version 2.
- **Alternative Hypothesis:** Version 2 has a significantly different correctness level from version 1, indicating that users perform more accurately or less accurately with the modified version.

These are the following variables:

1. **Independent Variables:** The version of the tool (1 or 2, in this case, the original or proposed versions) used by the participants. Additionally, the personas Abi or Tim attributed to the participants and their facet values, and if the experiment was the user's first interaction with the language i* or with the piStar tool [46].
2. **Dependent Variables:** Metrics such as task completion time, number of errors or level of correctness, user's perception of usability (usability scores), and cognitive load (effort) measurements.

5.1.6 Design

The experiment was designed as an A/B testing study, where participants were assigned to either group A or group B based on necessity of balancing the groups. This design was chosen for its ability to directly compare the usability of the two versions without the influence of learning effects that could occur if participants were exposed to both versions. The decision to limit each participant to a single tool was made to ensure that any observed differences in performance could be attributed solely to the tool version, rather than to prior experience or familiarity.

5.1.7 Procedure

The experimental procedure was conducted remotely via Zoom, allowing for real-time observation and interaction with participants. Each session began with an introduction to the experiment and a brief explanation of the language i*. This introduction was the same for all participants.

Participants were then given a GenderMag survey, as in **Annex I** prior to the session. In the beginning of the session, a brief presentation of the language syntax was made, seen on **V**. The sessions were recorded for further analysis, with particular attention paid to task completion, errors, and participant feedback. After this, the user was given two surveys to answer, the NASA TLX¹ [13] to measure the user's perception of effort and the SUS questionnaire [10, 11] to gather the user's perception of the tool's usability.

5.1.8 Analysis Procedure

The data collected during the experiment was analyzed using a combination of quantitative and qualitative methods. Quantitative analysis involved statistical testing to evaluate the significance of the differences observed between the two groups, such as correctness level, completion time and questionnaire results. Qualitative analysis focused on participant feedback and observations made during the sessions. The analysis was structured to assess whether the data supported the hypotheses.

¹https://hsi.arc.nasa.gov/awards-pubs/news_view.php?news_id=117

5.2 Execution

5.2.1 Preparation

As part of the usability testing process, each user was first introduced to the i* modeling framework and the objectives of the session through a structured presentation, accompanying the slides in Appendix V. This presentation was critical in ensuring that all participants were as equally prepared for the tasks they were about to undertake, as possible.

The session opened with an explanation of the usability test's purpose, which was to evaluate a tool specifically designed for i* modeling. This was followed by a broader overview of the i* language, a goal-oriented modeling framework primarily used in requirements engineering and organizational analysis. The framework's distinctive approach not only identifies what requirements exist but also explores why they are needed by capturing the motivations and relationships between different actors within a system.

The presentation briefly touched on the two main components of the i* framework, the SR Model and the SD Model.

Participants were also introduced to the key concepts of i* modeling, such as actors, goals, tasks, resources, qualities, and various types of relationships like "and," "or," and "needed-by." The relationships between tasks and qualities, including how a task might contribute positively or negatively to a quality, were emphasized to ensure users understood the framework's core mechanics.

Once the framework was explained, participants were briefed on their specific task: to build an i* model step-by-step using the provided guide and reference images. The observer's role during this process was non-intrusive, with their responsibility limited to clarifying tasks or addressing any uncertainties, ensuring that the usability of the platform could be evaluated without bias.

To facilitate the testing, participants were asked to control the presenter's screen via Zoom, where the i* modeling was running. Throughout the session, participants were encouraged to think aloud, sharing their thought processes, challenges, and insights as they navigated the tool. This verbal feedback was crucial for capturing the users' experiences and logic in real time.

The presentation followed a standardized format, ensuring consistency across all testing sessions. This uniform introduction guaranteed that each participant started the task with closely the same level of understanding of both the i* framework and the usability test objectives, creating a controlled environment for gathering comparative data on the tool's performance.

5.2.2 Deviations

A notable deviation occurred during Step 17, which asked participants to "Compare and analyze the information related to three items: 'Manually merge dates,' 'Merge dates

using scheduler,' and 'Minimal effort.'" The expected behavior in Version 1 was for users to manually compare the information and properties of these three items iteratively. In Version 2, however, users had an additional option: if they selected all three items simultaneously, a "generate report" option would appear, automating the comparison process.

While most participants eventually reached the expected action, many were confused initially and paused for around 30 seconds up to 60 seconds, requiring a tip to proceed. This confusion appears to have been caused by the phrasing of the task, which emphasized comparing and analyzing as the end goal. The task was deliberately worded to avoid leading participants to the "generate report" feature in Version 2, but this lack of direction left them uncertain about how to complete the task. After receiving a tip or clarification, participants proceeded as expected.

The issue seems to lie in the cognitive disconnect between the instruction and the task's end goal. When asked simply to compare and analyze, without a more tangible action or problem to solve, participants hesitated, unsure of how to engage with the interface. While they eventually discovered the intended functionality, the tip was necessary to guide them. This suggests that the task was framed too abstractly, and participants may have been more successful without guidance if the task had been presented as solving a specific question or problem. For instance, asking users to provide a specific insight based on their comparison could have directed them to naturally use the comparison or report-generation features as part of the solution process.

To ensure that the analysis was not affected by this deviation, all average time-related statistics were calculated both with and without the inclusion of step 17.

In summary, while most participants did arrive at the expected action, the confusion and hesitation observed indicate that the phrasing of Step 17 did not optimally guide them. The need for a tip highlights the importance of providing clear instructions that not only avoid leading users but also help them intuitively understand the purpose of the task. A more goal-oriented phrasing could have reduced confusion and allowed participants to discover the tool's features more organically.

5.3 Analysis

In this section, we present an analysis of the data collected during the study, utilizing both descriptive statistics and correlation analyses to understand the relationships between various user facets and tool performance. The analysis explores facets such as self-efficacy, motivations, learning style, information processing style, and risk aversion, and examines their relationships with the dependent variables, including average time, NASA TLX, SUS, and correctness level.

5.3.1 Descriptive Statistics

In this section, we begin by providing a detailed overview of the participants' demographics, focusing specifically on their age and prior experience with the i* framework, as presented in Table 5.1. This information is essential to contextualize the study.

User ID	Age	Has previous experience with i*?
1	23	Yes
2	23	Yes
3	22	No
4	22	No
5	23	No
6	22	No
7	22	No
8	22	Yes
9	29	No
10	28	No
11	22	Yes
12	23	Yes
13	24	No
14	23	Yes
15	25	No
16	26	No
17	23	No
18	23	No
19	21	No
20	23	No
21	23	No
22	54	No
23	24	No
24	25	No

Table 5.1: Distribution of participants' ages and experience with i*

One of the key objectives of this study was to balance the two personas, Abi and Tim, across both versions to ensure that each version had an equitable distribution of participants in terms of the GenderMag personas.

As seen in Table 5.2, the mean GenderMag values between the two versions are quite similar, suggesting that, on average, both groups were comparable in their cognitive characteristics. This was an important goal of the study, as we aimed to evaluate both versions on an equal footing, as much as possible. However, despite the similar means, the t-test results in Table 5.3 reveal no statistically significant difference between the two versions, indicating that the variations observed between the groups were minimal and likely not impactful in skewing the results of the evaluation.

However, despite our efforts to balance the groups by pragmatically separating users based on the average of their facet values, the distribution of participants across the two

Table 5.2: Descriptive Statistics for GenderMag values

	Version	N	Mean	Std. Dev	Std. Error Mean
GenderMag	1	12	22.1833	2.72157	0.78565
	2	12	22.2667	2.61650	0.75532

Table 5.3: Independent Samples Test of the distribution of the GenderMag values of participants in version 1 and 2.

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval Lower	95% Confidence Interval Upper
GenderMag	Equal variances assumed	0.202	0.658	-0.076	22	0.940	-0.08333	1.08984	-2.34352	2.17686
	Equal variances not assumed			-0.076	21.966	0.940	-0.08333	1.08984	-2.34373	2.17706

versions is not perfectly balanced for every facet and it should be acknowledged as a potential limitation in interpreting the findings.

The histograms and box plots provide insight into the distribution of participant facets, beginning with self-efficacy in **Figure 5.1**, motivations in **Figure 5.2**, learning style in **Figure 5.3**, information processing style in **Figure 5.4**, and risk aversion in **Figure 5.5**. These visualizations highlight a slight imbalance in facet distribution across participants, which adds important context for interpreting the differences between the two versions. The imbalance is particularly noticeable in the learning style and information processing style distributions, as shown in **Figures 5.3** and **5.4**, where the inconsistencies are more pronounced.

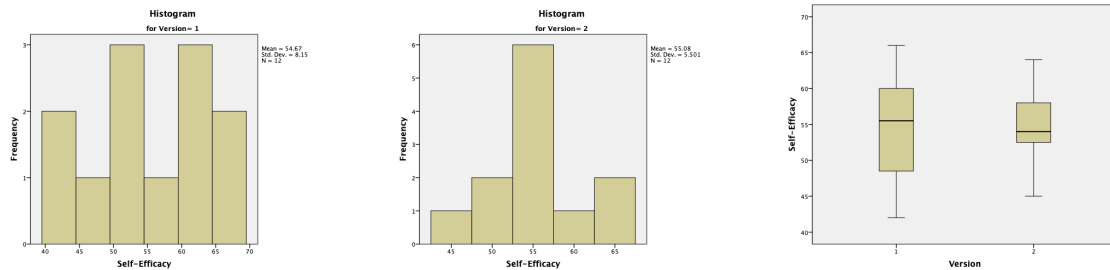


Figure 5.1: Histograms and a box plot to represent the distribution of the self-efficacy facet values through versions.

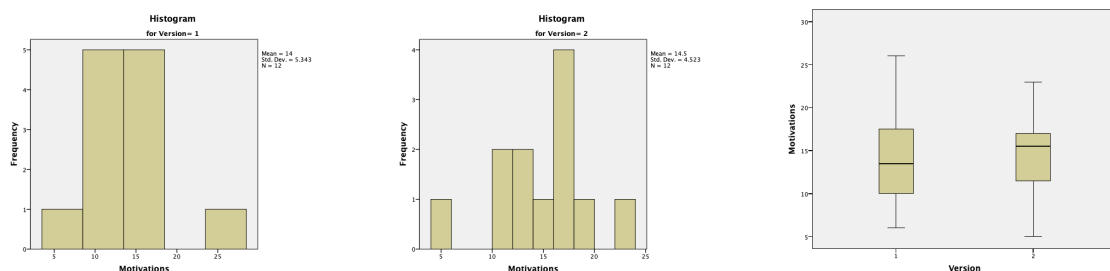


Figure 5.2: Histograms and a box plot to represent the distribution of the motivations facet values through versions.

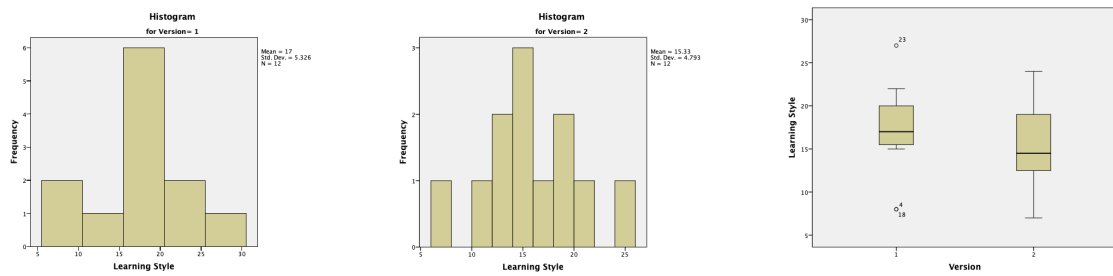


Figure 5.3: Histograms and a box plot to represent the distribution of the learning style facet values through versions.

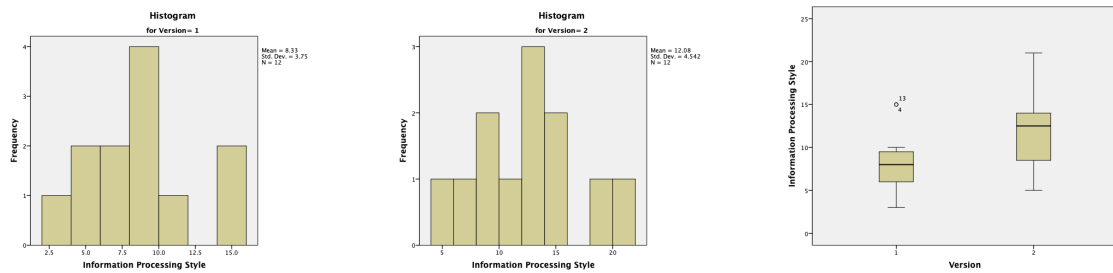


Figure 5.4: Histograms and a box plot to represent the distribution of the information processing style facet values through versions.

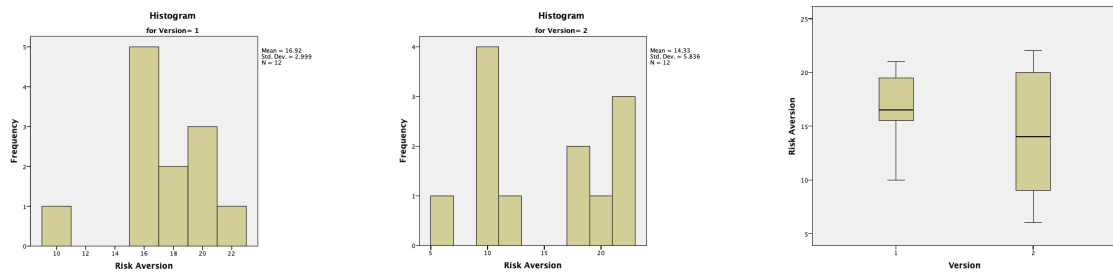


Figure 5.5: Histograms and a box plot to represent the distribution of the risk aversion facet values through versions.

This second part of the analysis involves examining the descriptive statistics for both independent and dependent variables across the two versions of the tool used in the study. Table 5.5 presents the mean, standard deviation, and Shapiro-Wilk significance values for the key variables. The Shapiro-Wilk test is used to assess the normality of the data distribution, providing insights into whether parametric tests such as t-tests are appropriate.

All facets, our dependent variables, appear to follow a normal distribution, as indicated by the Shapiro-Wilk test results, as in Table 5.4. However, this is not the case with some of the independent variables. Specifically, NASA TLX for version 2, SUS for version 1, and the Average Correctness Level in both versions do not seem to follow a normal distribution.

The third part of the analysis focuses on the correlation coefficients between independent variables (self-efficacy, motivations, learning style, information processing style, and risk aversion) and dependent variables (average time, NASA TLX, SUS, and correctness

Table 5.4: Descriptive Statistics for dependent variables with Shapiro-Wilk Significance

Variable	Version	N	Mean	Std. Dev	Std. Error Mean	SW Sig.
Self-Efficacy	1	12	54.67	8.150	2.353	0.385
	2	12	55.08	5.501	1.588	0.173
Motivations	1	12	14.00	5.343	1.542	0.705
	2	12	14.50	4.523	1.306	0.773
Learning Style	1	12	17.00	5.326	1.537	0.285
	2	12	15.33	4.793	1.384	0.996
Information Processing Style	1	12	8.33	3.750	1.082	0.255
	2	12	12.08	4.542	1.311	0.908
Risk Aversion	1	12	16.92	2.999	0.866	0.302
	2	12	14.33	5.836	1.685	0.069

Table 5.5: Descriptive Statistics for independent variables with Shapiro-Wilk Significance

Variable	Version	N	Mean	Std. Dev	Std. Error Mean	SW Sig.
Average Time	1	12	467.6617	81.95527	23.65845	0.219
	2	12	533.8708	155.95930	45.02157	0.162
Average Time (Excluding 17)	1	12	406.5567	67.41629	19.46141	0.611
	2	12	430.2533	112.36202	32.43612	0.221
NASA TLX	1	12	21.7208	16.50197	4.76371	0.127
	2	12	18.2792	11.51128	3.32302	0.029
SUS	1	12	88.5417	10.02601	2.89426	0.048
	2	12	90.0000	6.12372	1.76777	0.311
Average Correctness Level	1	12	2.9533	0.02774	0.00801	0.001
	2	12	2.9442	0.02906	0.00839	0.001

Table 5.6: Independent Samples T-Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	95% Confidence Interval	
									Lower	Upper
Average Time	Equal variances assumed	1.929	0.179	-1.302	22	0.206	-66.20917	50.85926	-171.68481	39.26648
	Equal variances not assumed			-1.302	16.645	0.211	-66.20917	50.85926	-173.68758	41.26925
Average Time (excluding 17)	Equal variances assumed	2.334	0.141	-0.626	22	0.537	-23.69667	37.82655	-102.14414	54.75081
	Equal variances not assumed			-0.626	18.011	0.539	-23.69667	37.82655	-103.16377	55.77044
NASA TLX	Equal variances assumed	2.872	0.104	0.593	22	0.560	3.44167	5.80822	-8.60384	15.48717
	Equal variances not assumed			0.593	19.656	0.560	3.44167	5.80822	-8.68768	15.57101
SUS	Equal variances assumed	1.638	0.214	-0.430	22	0.671	-1.45833	3.39142	-8.49171	5.57504
	Equal variances not assumed			-0.430	18.205	0.672	-1.45833	3.39142	-8.57771	5.66104
Average Correctness Level	Equal variances assumed	0.075	0.786	0.790	22	0.438	0.00917	0.01160	-0.01489	0.03322
	Equal variances not assumed			0.790	21.953	0.438	0.00917	0.01160	-0.01489	0.03322

level). Kendall's Tau-b and Spearman's Rho were used to compute these correlations. Tables 5.7, 5.8, and 5.9 present the correlations across all participants for versions in general and for each version individually.

In Table 5.7, we observe a significant positive correlation between motivations and average correctness level ($r = .466$, $p = .010$) in Kendall's Tau-b and ($r = .526$, $p = .008$) in Spearman's Rho, suggesting that participants with higher intrinsic motivation tend to have a higher correctness level when using the tool.

When breaking down the correlations by version (Tables 5.8 and 5.9), interesting differences emerge. For version 1, motivations show a significant positive correlation with average correctness level ($r = .511$, $p = .044$) in Kendall's Tau-b analysis and ($r = .584$,

Table 5.7: Kendall's Tau-b and Spearman's Rho Correlations for all samples

	Average Time	Average Time (Excluding 17)	NASA TLX	SUS	Average Correctness Level
Kendall's Tau-b					
Self-Efficacy Correlation Coefficient	-.197	-.204	-.052	.024	-.114
Sig. (2-tailed)	.187	.171	.727	.880	.508
Motivations Correlation Coefficient	.074	.022	.004	.110	.466**
Sig. (2-tailed)	.618	.881	.980	.479	.010
Learning Style Correlation Coefficient	.104	.074	.086	-.106	.211
Sig. (2-tailed)	.485	.618	.566	.495	.211
Information Processing Style Correlation Coefficient	.189	.182	.164	-.126	-.228
Sig. (2-tailed)	.204	.222	.273	.419	.221
Risk Aversion Correlation Coefficient	.063	.086	.179	-.232	-.063
Sig. (2-tailed)	.672	.566	.231	.136	.716
N	24	24	24	24	24
Spearman's Rho					
Self-Efficacy Correlation Coefficient	-.255	-.300	-.092	.036	-.151
Sig. (2-tailed)	.230	.154	.669	.866	.481
Motivations Correlation Coefficient	.059	-.010	.010	.172	.526**
Sig. (2-tailed)	.785	.963	.964	.423	.008
Learning Style Correlation Coefficient	.126	.092	.086	-.154	.260
Sig. (2-tailed)	.558	.669	.690	.473	.220
Information Processing Style Correlation Coefficient	.233	.253	.215	-.179	-.284
Sig. (2-tailed)	.272	.232	.314	.403	.179
Risk Aversion Correlation Coefficient	.113	.089	.255	-.326	-.073
Sig. (2-tailed)	.599	.679	.230	.120	.736
N	24	24	24	24	24

Table 5.8: Kendall's Tau-b and Spearman's Rho Correlations for samples from version 1

	Average Time	Average Time (Excluding 17)	NASA TLX	SUS	Average Correctness Level
Kendall's Tau-b					
Self-Efficacy Correlation Coefficient	.000	-.031	-.188	-.048	-.236
Sig. (2-tailed)	1.000	.890	.405	.880	.356
Motivations Correlation Coefficient	.092	.123	-.092	.110	.511*
Sig. (2-tailed)	.679	.582	.679	.627	.044
Learning Style Correlation Coefficient	.240	.176	-.016	-.148	-.097
Sig. (2-tailed)	.292	.440	.944	.525	.709
Information Processing Style Correlation Coefficient	.094	.219	.156	.048	-.189
Sig. (2-tailed)	.678	.332	.489	.834	.461
Risk Aversion Correlation Coefficient	.173	.205	.079	-.339	.095
Sig. (2-tailed)	.445	.366	.728	.142	.712
N	12	12	12	12	12
Spearman's Rho					
Self-Efficacy Correlation Coefficient	.025	-.049	-.229	.034	-.304
Sig. (2-tailed)	.939	.879	.473	.917	.336
Motivations Correlation Coefficient	.109	.144	-.067	.208	.584*
Sig. (2-tailed)	.736	.656	.837	.516	.046
Learning Style Correlation Coefficient	.300	.246	-.018	-.203	-.075
Sig. (2-tailed)	.344	.441	.956	.528	.817
Information Processing Style Correlation Coefficient	.092	.292	.176	.073	-.232
Sig. (2-tailed)	.777	.357	.584	.823	.469
Risk Aversion Correlation Coefficient	.191	.230	.131	-.425	.135
Sig. (2-tailed)	.552	.473	.685	.169	.676
N	12	12	12	12	12

$p = .046$) in Spearman's Rho. This suggests that highly motivated users in version 1 were more accurate. In contrast, this effect was mitigated on version 2, where this correlation is no longer visible.

However, for version 2, a stronger negative correlation is found between self-efficacy and average time ($r = -.627$, $p = .029$) in Spearman's Rho, indicating that users with higher self-efficacy completed tasks significantly faster in this version.

5.3.1.1 Conclusion on Hypotheses

Based on the analysis of the collected data, we evaluated the hypotheses for each of the key dependent variables. The findings for each hypothesis are discussed below:

1. **Average Time:** The results indicate that there is no statistically significant difference

Table 5.9: Kendall's Tau-b and Spearman's Rho Correlations for samples from version 2

	Average Time	Average Time (Excluding 17)	NASA TLX	SUS	Average Correctness Level
Kendall's Tau-b					
Self-Efficacy Correlation Coefficient	-.438	-.375	-.031	.120	.141
Sig. (2-tailed)	.053	.097	.890	.613	.581
Motivations Correlation Coefficient	-.016	-.109	.264	.119	.468
Sig. (2-tailed)	.945	.629	.240	.614	.066
Learning Style Correlation Coefficient	.185	.062	.277	-.067	.209
Sig. (2-tailed)	.408	.783	.215	.774	.409
Information Processing Style Correlation Coefficient	.277	.154	.215	-.303	-.046
Sig. (2-tailed)	.215	.491	.335	.195	.854
Risk Aversion Correlation Coefficient	.031	-.094	.219	-.303	-.046
Sig. (2-tailed)	.890	.677	.332	.469	.406
N	12	12	12	12	12
Spearman's Rho					
Self-Efficacy Correlation Coefficient	-.627*	-.599*	-.035	.201	.195
Sig. (2-tailed)	.029	.040	.913	.531	.544
Motivations Correlation Coefficient	-.105	-.172	.281	.139	.539
Sig. (2-tailed)	.744	.592	.376	.667	.070
Learning Style Correlation Coefficient	.182	.042	.305	-.106	.259
Sig. (2-tailed)	.570	.897	.335	.744	.417
Information Processing Style Correlation Coefficient	.326	.225	.319	-.421	-.065
Sig. (2-tailed)	.301	.483	.312	.173	.824
Risk Aversion Correlation Coefficient	.106	-.088	.332	-.264	-.241
Sig. (2-tailed)	.743	.785	.292	.407	.676
N	12	12	12	12	12

in the average time between Version 1 and Version 2. Therefore, we accept the null hypothesis since we cannot conclude that Version 2 leads to significantly faster or slower task completion times compared to Version 1.

2. **SUS:** The SUS scores showed a difference between the two versions, with Version 2 slightly outperforming Version 1. However, the analysis shows no statistically significant difference. Therefore, we accept the null hypothesis since we cannot conclude that Version 2 has a higher or lower usability score than Version 1.
3. **NASA TLX:** For the NASA TLX scores, the analysis reveals a difference between Version 1 and Version 2, with Version 2 showing lower cognitive load. However, once again, this result indicates that there is no statistically significant difference. As a result, we accept the null hypothesis since we cannot conclude that Version 2 has higher or lower cognitive load than Version 1.
4. **Correctness Level:** The results indicate that there is no statistically significant difference in the correctness level between Version 1 and Version 2. As a result, we accept the null hypothesis since we cannot conclude that Version 2 results in improved or worsened accuracy compared to Version 1.

In conclusion, the analysis of the hypotheses across the four key variables — Average Time, SUS scores, NASA TLX scores, and Correctness Level — consistently revealed no statistically significant differences between Version 1 and Version 2. Despite some observable trends, such as Version 2 showing marginal improvements in usability and cognitive load, these differences were not strong enough to reject the null hypotheses. Therefore, we conclude that Version 2 does not offer substantial improvements over Version 1 in terms of task completion time, usability, cognitive load, or accuracy.

5.4 Discussion

5.4.1 Evaluation of Results and Implications

Before beginning this evaluation of the results, it's important to note that the lack of an abundance of statistically significant differences between the versions, combined with the survey feedback from participants, indicates that the tool in its original form does not have severe inclusivity issues. It demonstrates high user satisfaction and low effort requirements, which makes further improvements more challenging.

5.4.1.1 Quantitative Evaluation

As mentioned in subsection 5.3.1, despite our efforts to balance participants across versions with attention to the personas they aligned with, an analysis of the facet values by version reveals some imbalances, particularly in the distribution of information processing styles. This facet's distribution differs significantly between the two versions. Upon reflection, this may be related to the notable differences in the mean values of the average times for each version, as shown in table 5.4.

At first glance, one might expect that with more intuitive features, users would take less time completing the tasks. However, version 2 actually recorded longer times, both when considering all steps and excluding step 17. When examining the participants' information processing styles, we see a significant number of Abis, with lower facet values in this area, using version 1, while Tims, with higher facet values, were more prevalent in version 2.

Initially, we struggled to understand why version 2 showed higher times than version 1, given that only one new feature was added. However, we then considered the characteristics of the participants themselves. Jumping to a tentative conclusion, this may support the idea that Tims (who statistically align more closely with males) tend to engage in counterproductive tinkering [5]. This could relate to the finding that Abis (statistically more aligned with females) tend to pause more while tinkering, and this pausing encourages reflection [5]. This leads us to consider that version 2 may have had more tinkerers who explored more extensively, which ultimately proved counterproductive in terms of average time.

Furthermore, the correlations between certain user facets and performance metrics offer interesting insights into how individual characteristics, such as self-efficacy and motivation, influence tool usage. The significant correlation between self-efficacy and task completion time in Version 2 suggests that users with higher confidence in their abilities were able to take greater advantage of the modified tool's features, completing tasks more quickly. This indicates that the improvements made to the tool may particularly benefit users with higher self-efficacy, potentially reducing cognitive barriers for more confident users.

However, the reduced motivational impact observed in Version 2 suggests that the modifications may have leveled the playing field, allowing less motivated users to achieve the same level of correctness as more motivated users, without making the latter's performance worse. This improvement enhances accessibility across a broader range of user traits. It indicates that refining usability features has made the tool more universally usable, aligning with the broader goal of fostering inclusivity in software by minimizing barriers that diverse users might encounter.

5.4.1.2 Qualitative Evaluation

Although the statistical analysis indicated no significant differences between Version 1 and Version 2 across the evaluated metrics (average completion time, SUS, NASA TLX, and correctness level), several insights can be drawn from the qualitative data gathered during the experiment. Notably, user behavior and feedback revealed critical information about usability challenges and the intuitive design of both tool versions.

A key observation was that a majority of participants (70.83%) across both versions attempted to use the multi-selection feature as their first action. This behavior occurred even among users who had used the tool before, indicating that multi-selection is a natural expectation for users when interacting with graphical modeling tools. The lack of this feature in Version 1 led to noticeable frustration among participants, who expressed confusion when unable to perform this action. Although users adapted by performing actions iteratively, the initial frustration suggests that the multi-selection feature could significantly improve user experience. This frustration would have been even more relevant if the test had incorporated bio-sensors to measure stress or frustration, as well as eye-tracking technology, which can also provide insights into stress levels [32, 61].

Interestingly, even two participants with prior experience with the tool attempted to use the multi-selection feature, as mentioned, further reinforcing the idea that this feature is not only desirable but intuitive. In Version 2, where multi-selection was available, users seamlessly executed the tasks without confusion, highlighting the feature's value in aligning with user expectations.

Despite these frustrations, the overall user satisfaction, as measured through the SUS survey, did not reflect high levels of frustration. This may be due to users adapting to the lack of multi-selection during the course of the tasks, and their feedback, captured after task completion, most times no longer reflected the initial discomfort. For the users with prior experience, their familiarity with the tool likely mitigated frustration, as they adjusted quickly to the absence of multi-selection in Version 1. These factors support the notion that the tool already demonstrated high levels of usability in its original state.

Additionally, one user without prior experience using the tool and assigned to Version 1 remarked on the odd presentation of the name and description fields. Specifically, they found it peculiar that these fields were underlined and blue, resembling hyperlinks, and that the description field showed "Empty" as a placeholder when no content was provided.

This feedback aligns with some of the design changes made in Version 2, where such visual elements were altered to create a more intuitive and less confusing interface.

While this feedback was limited to a single participant, it highlights the potential impact of small design inconsistencies on user perception and experience. In Version 2, these interface adjustments were intended to make the tool more user-friendly and reduce confusion around editable fields. If a larger sample size had been used, it is possible that more prominent results regarding these interface changes could have emerged. This suggests that subtle design enhancements, even if not statistically validated in this study, could still play a relevant role in improving overall user satisfaction and reducing cognitive friction during task completion.

Regarding the changes made to alert messages and the delete button, no notable findings emerged. However, it is worth mentioning that some participants exhibited nervous behavior when confirming these alerts, often seeking reassurance from the observer before proceeding with the action. Despite this, no clear patterns relating to personas or tool versions were observed.

These insights suggest that while the statistical measures did not reveal significant differences, there are meaningful qualitative differences between the two versions. The multi-selection tool, in particular, appears to play a critical role in enhancing user experience, reducing frustration, and aligning with users' natural workflow expectations. These findings underline the importance of integrating user-centric features that cater to intuitive behaviors, even when quantitative metrics do not fully capture the user experience nuances.

5.4.2 Threats to Validity

Several factors may have influenced the validity of the results obtained in this experiment. These limitations are discussed below to provide context for the interpretation of the findings and to acknowledge areas where the study's internal and external validity may be compromised.

One of the most significant threats to validity is the small sample size used in the experiment. Due to resource constraints, fewer participants were recruited than originally planned. This limited the statistical power of the study and may have reduced the generalizability of the results. With a larger sample, the results could potentially yield more robust conclusions and better represent the diverse range of user experiences.

Another concern is the fact that the majority of the participants were either current or former students of NOVA University or Técnico. This homogeneity in the participant pool may have introduced bias into the results. Participants with similar academic backgrounds might share common approaches or expectations when using software tools, potentially skewing the usability feedback. Additionally, their familiarity with similar academic environments may limit the external validity of the results when applied to broader user populations in different contexts.

Furthermore, the participants registered values on the GenderMag spectrum within a narrow range, likely due to their shared academic focus. While they may represent diversity within this interval, they do not represent the population as a whole, which presents a limitation. However, for the purpose of this study, which focuses on developer experience, the participant pool needed to consist of developers, inherently constraining the range of the GenderMag spectrum. This homogeneity may also be tied to the challenge of finding significant variations in the results, as was observed during the study.

Another limitation to consider is the imbalance in the distribution of participants across the two versions of the tool. Despite our efforts to balance the groups by assigning users based on the average of their facet values, a perfect balance was not achieved for all facets. While this imbalance does not appear to have significantly influenced the overall results, it could be tied to some discrepancies in the data, as mentioned in [5.4.1.1](#).

The tasks assigned during the experiment were relatively simple, which may have given participants a misleading sense of the tool's overall usability. Although the tasks were designed to cover essential functionalities, their simplicity may not have fully tested the range of challenges users could face when using the tool in more complex, real-world scenarios. As a result, participants may have perceived the tool as more intuitive or efficient than it would have been in a more demanding setting. This can lead to an overestimation of the tool's ease of use.

The brevity of the i* model created during the session could also limit the insights drawn from the experiment. Since the model was small, participants were not exposed to potential difficulties or limitations that might emerge when using the tool for more extensive or complex models. Longer modeling sessions could reveal usability issues that were not evident in this study, potentially affecting the assessment of the tool's performance.

Additionally, to minimize any performance disparities arising from participants' varying levels of experience, the tasks were designed with high granularity. This meant that users were required to focus less on interpreting what the task required and more on how to perform it using the tool's interface. While this approach was partly intentional, aimed at reducing potential biases introduced by differences in problem-solving abilities, it also represents a limitation. The reliance on step-by-step instructions means the study doesn't capture how users might approach tasks independently, and that differences arising from diverse problem-solving styles were less likely to emerge because the instructions were detailed. One way to address this would be to vary the granularity across tasks, allowing some steps to be more open-ended. However, this would reintroduce the initial issue of uneven experience levels influencing the results.

In summary, the small sample size, the homogeneity of the participant pool, the simplicity of the tasks, the slightly imbalanced distribution of facets by version and the short modeling sessions are all factors that may have affected the results of this study. These limitations should be taken into consideration when interpreting the findings.

5.4.3 Inferences

The analysis of the results highlights several important conclusions regarding the usability of the piStar tool, both in its original and modified versions, and provides insights into the impact of the modifications on developer experience, particularly in relation to diverse user personas. While the statistical results did not reveal significant differences between the two versions of the tool across key variables (such as task completion time, usability scores, cognitive load, and correctness levels), qualitative feedback from participants underscores notable trends that offer valuable lessons.

First, the absence of statistically significant differences in the quantitative data should not be interpreted as evidence that the modifications made to the piStar tool were inconsequential. The qualitative data from user feedback provides evidence that several of the new features, particularly the multi-selection functionality, addressed critical usability issues that users naturally expected to be part of the interface. The fact that users repeatedly tried to use multi-selection in Version 1, where it was not implemented, suggests that this feature aligns closely with common user expectations in graphical modeling environments. This feature's absence caused frustration, while its presence in Version 2 allowed for smoother interactions, indicating a clear improvement in user experience.

Additionally, the study demonstrates the complexity of balancing quantitative and qualitative data when assessing usability. The statistical data, which is often relied upon for robust conclusions, did not capture the full scope of user frustration or satisfaction levels. In contrast, the qualitative insights provide a more nuanced view, revealing that small design adjustments, even if they do not immediately translate into measurable improvements in task completion times or perceived cognitive load, can significantly affect user experience. This underscores the importance of considering user feedback and qualitative measures alongside quantitative data in software usability studies.

In conclusion, the results suggest that while the modified version of the piStar tool did not yield significant statistical improvements across all metrics, it addressed key usability issues and improved user satisfaction, particularly in alignment with user expectations for common features like multi-selection. The study highlights the importance of qualitative feedback in usability testing and suggests that even small design changes can have a meaningful impact on diverse user populations.

5.4.4 Lessons Learned

Throughout the process of planning, executing, and analyzing the study, several valuable lessons emerged that can inform future research efforts and usability testing in the context of software tools. These lessons touch on a variety of practical, methodological, and analytical considerations.

- **The Importance of Balanced Participant Groups:** One of the primary challenges encountered during the study was ensuring an even distribution of participant

personas across the two versions of the tool. Despite efforts to balance the groups, some facets distributions remained slightly imbalanced, which could have influenced the outcomes. This experience highlights the need for more rigorous participant selection and balancing strategies, especially in studies using personas like Abi and Tim. Ensuring equal representation of cognitive traits in both groups is essential to minimize biases and obtain reliable results.

- **The Value of Qualitative Data:** While statistical significance was not found in many of the key metrics, qualitative data provided critical insights into how participants interacted with the tool and how certain features (like multi-selection) impacted user experience. This reinforces the idea that qualitative data is an indispensable complement to quantitative metrics in usability studies. Small nuances, such as user frustration or adaptation, may not always be captured in numerical data but can be pivotal in understanding the overall user experience. Incorporating "think-aloud" protocols, user interviews, or post-task reflections can provide a more realistic picture of how a tool is perceived and used.
- **The Challenges of Remote Experimentation:** Conducting the study remotely, while necessary for practicality, presented some difficulties in ensuring consistency across sessions. Participants' varying familiarity with online tools like Zoom, along with potential technical issues, may have introduced variability in how tasks were performed. Although remote experimentation offers flexibility, our experience suggests that future studies could benefit from in-person testing, with more comprehensive data on users' stress levels. Given the importance of qualitative analysis in this study, the inclusion of bio-sensors or eye tracking technology might provide more relevant and abundant insights.
- **Designing Tasks that Reflect Real-World Use:** The tasks assigned to participants, while useful in evaluating specific tool functionalities, were relatively simple and did not fully capture the complexity of real-world software engineering tasks. This could have led participants to overestimate the tool's ease of use, as they were not required to engage with the tool over extended periods or on more complex problems. Future studies should incorporate more intricate, real-world scenarios to better evaluate a tool's usability under conditions that more closely mimic actual user workflows. Additionally, longer sessions could reveal more subtle usability challenges that only become apparent with extended use, like user fatigue.
- **The Need for Larger Sample Sizes:** A recurring limitation in this study was the small sample size, which reduced the statistical power of the analysis and limited the ability to detect significant differences between the two tool versions. A larger sample size would provide more robust results and better reflect the diversity of user experiences. Future studies should allocate sufficient time and resources to recruit

a larger, more diverse participant pool to improve the validity and generalizability of the findings.

- **Clear Task Instructions and Goal-Oriented Tasks:** The confusion surrounding certain tasks, particularly Step 17, demonstrated the importance of clear and well-structured task instructions. Participants hesitated when the task was too abstract or lacked a clear, actionable goal. In future studies, tasks should be framed with concrete objectives, helping participants intuitively engage with the tool's features. Phrasing tasks in a problem-solving context can guide users more effectively and reduce the need for intervention from observers.

In summary, the making of this study has reinforced the importance of careful planning, balancing, and methodological rigor in usability research. It has highlighted how qualitative insights and real-world tasks are key to understanding and improving software tools for diverse user groups. By applying these lessons in future work, the effectiveness of usability testing and the quality of software tools can be greatly enhanced.

CONCLUSIONS

6.1 Summary

This study aimed to enhance gender inclusiveness in software development tools by applying the GenderMag method to the piStar tool. The GenderMag method was used to identify potential gender biases, followed by targeted modifications to improve inclusivity, usability, and developer experience for a diverse population of developers. A controlled experiment was conducted to evaluate the effectiveness of these modifications, comparing the original version of the piStar tool (Version 1) with the modified version (Version 2).

Key areas of evaluation included task completion time, usability (measured by SUS scores), cognitive load (assessed by NASA TLX scores), and correctness levels. The statistical analysis did not show significant differences between the two versions across these metrics, leading to the acceptance of the null hypotheses for all parameters. However, qualitative insights gathered from participant feedback revealed valuable usability patterns and design challenges that were not fully captured by quantitative metrics.

The findings indicate that while Version 2 did not demonstrate statistically significant improvements, several design changes, particularly the addition of the multi-selection feature, addressed key user frustrations. A significant proportion of participants across both versions intuitively attempted to use the multi-selection functionality, emphasizing its importance for user experience. Additionally, design inconsistencies in Version 1, such as the presentation of the name and description fields, were highlighted by a user as confusing, further reinforcing the necessity of interface refinements in Version 1.

In conclusion, although the statistical analysis did not show clear superiority of Version 2, the qualitative feedback suggests that certain modifications have the potential to meaningfully improve the usability and inclusiveness of the tool. The study demonstrates that small design changes can have significant impacts on user satisfaction and experience, even if these impacts are not always statistically measurable in a limited-sample experiment.

6.2 Impact

The impact of this research extends beyond the specific improvements made to the piStar tool. By applying the GenderMag method, this study contributes to the broader effort of making software development tools more inclusive and equitable. Gender inclusivity in software tools fosters diversity in software engineering, ensuring that developers of all genders can participate fully and productively.

One of the key contributions of this research is the demonstration of how gender-inclusive design principles can be systematically integrated into developer tools. By identifying and addressing usability issues related to cognitive style, motivations, and risk aversion, this study has laid the groundwork for more inclusive software design practices in the broader field of software engineering.

Moreover, this research has highlighted the importance of UCD in addressing hidden biases that may otherwise go unnoticed. The insights gathered from this study emphasize that inclusivity improvements benefit not only underrepresented groups but also improve the overall user experience for everyone.

Finally, the modifications made to the piStar tool and the application of the GenderMag Design Catalog suggestions aimed at contributing to the validation of the catalog, enriching its body of empirical evidence. This can guide future software development efforts aimed at enhancing inclusivity in technical tools.

6.3 Future Work

Several avenues for future research and development emerge from this study. First and foremost, increasing the sample size in future experiments could provide more statistically robust insights into the effectiveness of the inclusivity modifications made to the piStar tool. A larger, more diverse participant pool would also help in assessing the generalizability of the findings to a broader user base.

Additionally, future studies could explore more complex tasks and longer modeling sessions. This would likely expose a wider range of usability challenges and provide a more comprehensive understanding of the tool's performance under different levels of complexity.

Finally, integrating other inclusivity frameworks, such as the Inclusive Design Toolkit¹ and InclusiveMag[38] could enrich future research. Since InclusiveMag is a generalization of GenderMag, it can be adapted to focus on specific user groups with unique needs. This could include, for instance, addressing the requirements of senior developers or neurodivergent individuals. By tailoring inclusivity methods to different user demographics, future tools could better accommodate a wider range of abilities and backgrounds, enhancing the inclusivity of software design overall.

¹<http://www.inclusivedesigntoolkit.com>

BIBLIOGRAPHY

- [1] A. C. G. de Almeida. “Quality Evaluation of Requirements Models: The Case of Goal Models and Scenarios”. PhD thesis. Universidade NOVA de Lisboa (Portugal), 2019 (cit. on p. 15).
- [2] S. Altiner and M. B. Ayhan. “An approach for the determination and correlation of diversity and efficiency of software development teams”. In: *South African Journal of Science* 114.3-4 (2018), pp. 1–9 (cit. on p. 18).
- [3] A. Bandura. “Social foundations of thought and action”. In: *Englewood Cliffs, NJ* 1986.23-28 (1986) (cit. on p. 6).
- [4] 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 p. 6).
- [5] 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. 6, 88).
- [6] A. Blandford, A. L. Cox, and P. Cairns. “Controlled experiments”. In: Cambridge University Press, 2008 (cit. on p. 16).
- [7] A. E. Blandford et al. “Scoping analytical usability evaluation methods: a case study”. In: *Human–Computer Interaction* 23.3 (2008), pp. 278–327 (cit. on pp. xi, 9).
- [8] A. Bourazeri and S. Stumpf. “Co-designing smart home technology with people with dementia or Parkinson’s disease”. In: *Proceedings of the 10th Nordic Conference on Human-Computer Interaction*. 2018, pp. 609–621 (cit. on p. 11).
- [9] 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 p. 8).
- [10] J. Brooke. “SUS: A quick and dirty usability scale”. In: *Usability Evaluation in Industry* (1996) (cit. on p. 79).
- [11] J. Brooke. “SUS: a retrospective.” In: *Journal of usability studies* 8.2 (2013) (cit. on p. 79).

- [12] J. Butler. *Gender trouble: Feminism and the subversion of identity*. routledge, 2011 (cit. on p. xi).
- [13] A. Cao et al. “NASA TLX: Software for assessing subjective mental workload”. In: *Behavior research methods* 41 (2009), pp. 113–117 (cit. on p. 79).
- [14] J. Castro, M. Kolp, and J. Mylopoulos. “A requirements-driven development methodology”. In: *Advanced Information Systems Engineering: 13th International Conference, CAiSE 2001 Interlaken, Switzerland, June 4–8, 2001 Proceedings* 13. Springer. 2001, pp. 108–123 (cit. on p. 14).
- [15] E. F. Churchill. “Sugared puppy-dog tails: gender and design”. In: *Interactions* 17.2 (2010), pp. 52–56 (cit. on p. 9).
- [16] M. Ciolkowski et al. “Empirical investigation of perspective-based reading: A replicated experiment”. In: *Fraunhofer Institute for Experimental Software Engineering, Germany* (1997) (cit. on p. 75).
- [17] F. Dalpiaz, X. Franch, and J. Horkoff. “istar 2.0 language guide”. In: *arXiv preprint arXiv:1605.07767* (2016) (cit. on p. 14).
- [18] A. Dix. *Human-computer interaction*. Pearson Education, 2003 (cit. on pp. 15, 16).
- [19] T. Dohmen et al. “Individual risk attitudes: Measurement, determinants, and behavioral consequences”. In: *Journal of the european economic association* 9.3 (2011), pp. 522–550 (cit. on p. 6).
- [20] J. Engmann. “Evaluating the impact of evolving requirements on wider system goals: using i* methodology integrated with satisfaction arguments to evaluate the impact of changing requirements in HIV / AIDS monitoring systems in the UK”. PhD thesis. MSc Dissertation, School of Informatics, City University, 2009 (cit. on p. 14).
- [21] F. Fagerholm and J. Münch. “Developer experience: Concept and definition”. In: *2012 International Conference on Software and System Process (ICSSP)*. 2012, pp. 73–77. DOI: [10.1109/ICSSP.2012.6225984](https://doi.org/10.1109/ICSSP.2012.6225984) (cit. on pp. xi, xii, 18, 19).
- [22] *First Principles of Interaction Design (Revised & Expanded)*. <https://asktog.com/atc/principles-of-interaction-design/>. Accessed: 2024-01-09 (cit. on p. 16).
- [23] *GenderMag Design Catalogue*. <https://gendermag.org/dc/>. Accessed: 2024-01-16 (cit. on pp. 20, 30–32, 66).
- [24] *GenderMag How it works*. <https://gendermag.org/gendermag.php>. Accessed: 2024-01-03 (cit. on p. 10).

- [25] “GenderMag: A Method for Evaluating Software’s Gender Inclusiveness”. In: *Interacting with Computers* 28.6 (2016-01). This is a pre-copyedited, author-produced version of an article accepted for publication in *Interacting with Computers* following peer review. The version of record Burnett, M., Stumpf, S., Macbeth, J., Makri, S., Beckwith, L., Kwan, I., Peters, A. & Jernigan, W. (2016). GenderMag: A Method for Evaluating Software’s Gender Inclusiveness. *Interacting with Computers*, 28(6), pp. 760-787 is available online at: <https://doi.org/10.1093/iwc/iwv046>, pp. 760-787. DOI: [10.1093/iwc/iwv046](https://doi.org/10.1093/iwc/iwv046). URL: <https://openaccess.city.ac.uk/id/eprint/16621/> (cit. on pp. xi, xii, 1, 2, 4, 5, 7-10, 13, 18).
- [26] E. Gonçalves et al. “A systematic literature review of istar extensions”. In: *Journal of Systems and Software* 137 (2018), pp. 1-33 (cit. on p. 14).
- [27] C. Gralha, M. Goulao, and J. Araujo. “Are there gender differences when interacting with social goal models? A quasi-experiment”. In: *Empirical Software Engineering* 25 (2020), pp. 5416-5453 (cit. on p. 34).
- [28] M. Guizani et al. “Gender Inclusivity as a Quality Requirement: Practices and Pitfalls”. In: *IEEE Software* 37.6 (2020), pp. 7-11. DOI: [10.1109/MS.2020.3019540](https://doi.org/10.1109/MS.2020.3019540) (cit. on pp. 2, 11, 17, 18).
- [29] M. M. Hamid et al. “How to measure diversity actionably in technology”. In: *Equity, Diversity, and Inclusion in Software Engineering: Best Practices and Insights* (2023) (cit. on p. 8).
- [30] M. Hertzum and N. E. Jacobsen. “The evaluator effect during first-time use of the cognitive walkthrough technique.” In: *HCI (1)*. Citeseer. 1999, pp. 1063-1067 (cit. on p. 10).
- [31] J. Horkoff et al. “Taking goal models downstream: A systematic roadmap”. In: *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*. 2014, pp. 1-12. DOI: [10.1109/RCIS.2014.6861036](https://doi.org/10.1109/RCIS.2014.6861036) (cit. on p. 14).
- [32] C. Jyotsna et al. “IntelEye: An intelligent tool for the detection of stressful state based on eye gaze data while watching video”. In: *Procedia Computer Science* 218 (2023), pp. 1270-1279 (cit. on p. 89).
- [33] J. Lockerbie et al. “Exploring the impact of software requirements on system-wide goals: a method using satisfaction arguments and i* goal modelling”. In: *Requirements Engineering* 17 (2012), pp. 227-254 (cit. on p. 14).
- [34] J. M. Lourenço. *The NOVAtesis L^AT_EX Template User’s Manual*. NOVA University Lisbon. 2021. URL: <https://github.com/joaomlourenco/novathesis/raw/main/template.pdf> (cit. on p. i).
- [35] E. Luger. “A design for life: Recognizing the gendered politics affecting product design”. In: *CHI workshop: perspectives on gender and product design*. 2014 (cit. on p. 8).

- [36] T. Mahatody, M. Sagar, and C. Kolski. "State of the art on the cognitive walkthrough method, its variants and evolutions". In: *Intl. Journal of Human-Computer Interaction* 26.8 (2010), pp. 741–785 (cit. on pp. xi, 9, 10).
- [37] J. Margolis and A. Fisher. *Unlocking the clubhouse: Women in computing*. MIT press, 2002 (cit. on p. 5).
- [38] 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. 11, 12, 96).
- [39] J. Meyers-Levy. "Gender differences in information processing: A selectivity interpretation". PhD thesis. Northwestern University, 1986 (cit. on p. 6).
- [40] J. Meyers-Levy and D. Maheswaran. "Exploring differences in males' and females' processing strategies". In: *Journal of consumer research* 18.1 (1991), pp. 63–70 (cit. on p. 6).
- [41] T. Neate et al. "Co-created personas: Engaging and empowering users with diverse needs within the design process". In: *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, pp. 1–12 (cit. on p. 11).
- [42] J. Nielsen. "Enhancing the explanatory power of usability heuristics". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 1994, pp. 152–158 (cit. on p. 16).
- [43] J. Nielsen and R. Molich. "Heuristic evaluation of user interfaces". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1990, pp. 249–256 (cit. on p. 16).
- [44] D. A. Norman. "Design rules based on analyses of human error". In: *Communications of the ACM* 26.4 (1983), pp. 254–258 (cit. on p. 16).
- [45] H. Petrie and N. Bevan. "The evaluation of accessibility, usability, and user experience." In: *The universal access handbook* 1 (2009), pp. 1–16 (cit. on p. 11).
- [46] *piStar tool*. <https://www.cin.ufpe.br/~jhcp/pistar/>. Accessed: 2024-02-06 (cit. on pp. 2, 13, 18, 32, 63, 64, 66–75, 79).
- [47] P. G. Polson and C. H. Lewis. "Theory-based design for easily learned interfaces". In: *Human-Computer Interaction* 5.2-3 (1990), pp. 191–220 (cit. on pp. xi, 9).
- [48] C. Power et al. "Guidelines are only half of the story: accessibility problems encountered by blind users on the web". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2012, pp. 433–442 (cit. on p. 17).
- [49] B. Shneiderman et al. *Designing the user interface: strategies for effective human-computer interaction*. Pearson, 2016 (cit. on p. 16).

-
- [50] R. Spencer. "The streamlined cognitive walkthrough method, working around social constraints encountered in a software development company". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 2000, pp. 353–359 (cit. on p. 9).
 - [51] J. Terrell et al. "Gender bias in open source: Pull request acceptance of women versus men." In: *PeerJ Prepr.* 4 (2016), e1733 (cit. on pp. 1, 4).
 - [52] B. Trinkenreich et al. "Women's Participation in Open Source Software: A Survey of the Literature". In: *ACM Trans. Softw. Eng. Methodol.* 31.4 (2022). ISSN: 1049-331X. DOI: [10.1145/3510460](https://doi.org/10.1145/3510460). URL: <https://doi.org/10.1145/3510460> (cit. on p. 4).
 - [53] P. Turner and S. Turner. "Is stereotyping inevitable when designing with personas?" In: *Design studies* 32.1 (2011), pp. 30–44 (cit. on pp. 6, 8).
 - [54] 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 pp. xii, 2, 18).
 - [55] 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*. CHI '19. Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–14. ISBN: 9781450359702. DOI: [10.1145/3290605.3300283](https://doi.org/10.1145/3290605.3300283). URL: <https://doi.org/10.1145/3290605.3300283> (cit. on pp. 2, 5, 18).
 - [56] 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. 29, 30).
 - [57] C. West and D. H. Zimmerman. "Doing gender". In: *Gender & society* 1.2 (1987), pp. 125–151 (cit. on p. xi).
 - [58] C. Wharton. "The cognitive walkthrough method: A practitioner's guide". In: *Usability inspection methods* (1994) (cit. on p. 9).
 - [59] C. Wharton et al. "Applying cognitive walkthroughs to more complex user interfaces: Experiences, issues, and recommendations". In: *Proceedings of the SIGCHI conference on Human factors in computing systems*. 1992, pp. 381–388 (cit. on p. 10).
 - [60] G. Williams. "Are you sure your software is gender-neutral?" In: *Interactions* 21.1 (2014), pp. 36–39 (cit. on pp. 4, 17).
 - [61] M. S. Yousefi et al. "Stress Detection Using Eye Tracking Data: An Evaluation of Full Parameters". In: *IEEE Access* 10 (2022), pp. 118941–118952 (cit. on p. 89).
 - [62] E. S.-K. Yu. "MODELLING STRATEGIC RELATIONSHIPS FOR PROCESS REENGINEERING". PhD thesis. Citeseer, 1995 (cit. on p. 14).
 - [63] E. S. Yu and J. Mylopoulos. "From ER To "aR"—modelling strategic actor relationships for business process reengineering". In: *International Journal of Cooperative Information Systems* 4.02n03 (1995), pp. 125–144 (cit. on p. 14).

BIBLIOGRAPHY

- [64] A. L. Zeldin and F. Pajares. "Against the odds: Self-efficacy beliefs of women in mathematical, scientific, and technological careers". In: *American educational research journal* 37.1 (2000), pp. 215–246 (cit. on p. [6](#)).

A

ABI (ABIGAIL/ABISHEK) PERSONA FOUNDATIONS



Abi (Abigail/Abishek)

→28 years old

→Employed as an Accountant

→Lives in Cardiff, Wales

Abi has always liked music. On their way to work in the mornings, Abi listens to music that spans a wide variety of styles. But when Abi arrives at work, they turn it off, and begin their day by scanning all their emails first to get an overall picture before answering any of them. (This extra pass takes time but seems worth it.) Some nights Abi exercises or stretches, and sometimes Abi likes to play computer puzzle games like Sudoku.

Background Knowledge and Skills

→Abi works as an accountant. They are comfortable with the technologies they use regularly, but they just moved to this employer 1 week ago, and their software systems are new to Abi.

→Abi says they're a "numbers person", but they have never taken any computer programming or IT systems classes. Abi likes Math and knows how to think with numbers. Abi writes and edits spreadsheet formulas in their work.

→In their free time, Abi also enjoys working with numbers and logic. Abi especially likes working out puzzles and puzzle games, either on paper or on the computer.

Motivations and Strategies

→Motivations: Abi uses technologies to accomplish their tasks. Abi learns new technologies if and when they need to, but prefers to use methods they are already familiar and comfortable with, to keep their focus on the tasks they care about.

→Information Processing Style: Abi tends towards a comprehensive information processing style when they need to get more information. So, instead of acting upon the first option that seems promising, Abi gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, Abi's style is "burst-y"; first they read a lot, then they act on it in a batch of activity.

Attitude to Technology

Abi is generally comfortable using familiar technology, but they do not get a big kick out of obtaining the latest gadgets or learning how to use them. Abi prefers to stay with the technologies for which they have already mastered the peculiarities, because of the following facets:

→Computer Self-Efficacy: Abi has low confidence about doing unfamiliar computing tasks. If problems arise with their technology, Abi often blames themselves for these problems. This affects whether and how Abi will persevere with a task if technology problems have arisen.

→Attitude toward Risk: Abi's life is a little complicated and they rarely have spare time. So Abi is risk averse about using unfamiliar technologies that they might need to spend extra time on, even if the new features might be relevant. Abi instead performs tasks using familiar features, because they're more predictable about what Abi will get from them and how much time they will take.

→Learning: by Process vs. by Tinkering: When learning new technology, Abi leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. Abi doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when Abi does tinker, it has positive effects on their understanding of the software.

PAT (PATRICIA / PATRICK) PERSONA FOUNDATIONS



Pat (Patricia/Patrick)

→43 years old

→Employed as an Accountant

→Lives in Cardiff, Wales

Pat loves public transportation and knows at least three routes to get there from home. When they arrive at work, Pat scans all their emails first to get an overall picture before answering any of them. (This extra pass takes time but seems worth it.) Some evenings Pat plays computer puzzle games like Sudoku before bed.

Background Knowledge and Skills

→Pat works as an accountant in a consulting firm. They prefer to stay with the technologies for which they have already mastered the peculiarities. Pat just moved to this employer 1 week ago, and their software systems are new to Pat.

→Pat describes themselves as a "numbers person", but they are not a professional programmer and have never taken any computer programming or IT systems classes. Pat has a degree in accounting so they know plenty of Math and know how to think in terms of numbers.

→Even though Pat's an accountant and deals with numbers all day at work, they like working with numbers in their free time, too. Pat especially likes Sudoku and other computer games that involve puzzling.

Motivations and Strategies

→Motivations: Pat learns new technologies when they need to, but Pat doesn't spend their free time exploring technology or exploring obscure functionality of programs and devices that they use. Pat tends to use methods they are already familiar and comfortable with to achieve their goals.

→Information Processing Style: Pat leans towards a comprehensive information processing style when they need to gather information to problem-solve. So, instead of acting upon the first option that seems promising, Pat first gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, Pat's style is "burst-y"; first they read a lot, then they act on it in a batch of activity.

Attitude to Technology

Pat is generally comfortable using familiar technology, but they do not get a big kick out of obtaining the latest gadgets or learning how to use them. Pat prefers to stay with the technologies for which they have already mastered the peculiarities, because of the following facets:

→Computer Self-Efficacy: Pat has medium computer self-efficacy about doing unfamiliar computing tasks. If problems arise with their technology, Pat will keep on trying to figure out how to achieve what they have set out to do for quite awhile; Pat doesn't give up right away when computers or technology present a challenge to them.

→Attitude toward Risk: Pat is busy and so they rarely have spare time. So Pat is risk averse and worries that they will spend time on their technology and not get any benefits from doing so. Pat prefers to perform tasks using familiar features, because they're more predictable about what Pat will get from them and how much time they'll take.

→Learning: by Process vs. Tinkering: When Pat sees a need to learn new technology, they do so by trying out new features or commands to see what they do and to understand how the software works. When Pat does this, they do so purposefully; that is, Pat reflects on each bit of feedback they get along the way to understand how the feature might benefit them. Eventually, if Pat doesn't think it will get them closer to what they want to achieve, Pat will revert back to ways that they already know will work.

TIM (TIMOTHY / TIMARA) PERSONA FOUNDATIONS



Tim (Timothy/Timara)

→28 years old

→Employed as an Accountant

→Lives in Cardiff, Wales

Tim loves public transportation. They know several routes to get there from home and they're always exploring ways to optimize their trips into the office. Work starts with email, which Tim answers one at a time, as soon as they read them. (Sometimes this backfires, if there is a second related message Tim hasn't read yet, but Tim doesn't mind sending a follow-up email.) Some nights Tim plays computer games with their online friends.

Background Knowledge and Skills

→Tim works as an accountant. They just moved to this employer 1 week ago, and their software systems are new to Tim. For Tim, technology is a source of fun, and they are always on the lookout for new computer software. Tim likes to make sure they have the latest version of all software with all the new features.

→Tim says they're a "numbers person", but Tim has not taken any computer programming or IT classes. Tim likes Math and knows how to think in terms of numbers. Tim writes and edits spreadsheet formulas for their work.

→Tim plays the latest video games, has the newest smart phone and a hybrid car. They download and install the latest software, and experiment with its settings. Tim is comfortable and confident with technology and they enjoy learning about it and using new technologies.

Motivations and Strategies

→**Motivations:** Tim likes learning all the available functionality on all of their devices and computer systems they use, even when it may not be necessary to help Tim achieve their tasks. Tim sometimes finds themselves exploring functions of one of their gadgets for so long that they lose sight of what they wanted to do with it to begin with.

→**Information Processing Style:** Tim leans towards a selective information processing style or "depth first" approach. That is, they usually delve into the first promising option, pursue it, and if it doesn't work out Tim backs out and gathers a bit more information until they see another option to try. Thus, Tim's style is very incremental.

Attitude to Technology

For Tim, technology is a source of fun, and they are always on the lookout for new computer software. Tim likes to make sure to have the latest version of all software with all the new features, because of the following facets:

→**Computer Self-Efficacy:** Tim has high confidence in their abilities with technology, and thinks they're better than the average person at learning about new features. If they can't fix the problem, Tim blames it on the software vendor; it's not Tim's fault if he can't get it to work.

→**Attitude toward Risk:** Tim doesn't mind taking risks using features of technology that haven't been proven to work. When Tim is presented with challenges because they have tried a new way that doesn't work, it doesn't change their attitudes toward technology.

→**Learning: by Process vs. by Tinkering:** Whenever Tim uses new technology, they try to construct their own understanding of how the software works internally. Tim likes tinkering and exploring the menu items and functions of the software in order to build that understanding. Sometimes Tim plays with features too much, losing focus on what they set out to do originally, but this helps them gain better understanding of the software.

ANNEX GENDERMAG SURVEY IN PORTUGUESE

GenderMag Facets Survey

From the GenderMag Project (<http://gendermag.org>)

Nome:

Idade:

Género:

Já conhecia a linguagem i*? SIM ☐ NÃO ☐

Já usaste a ferramenta 'piStar' antes? SIM ☐ NÃO ☐

	Discordo Completamente			Não Concordo Nem Discordo			Concordo Completamente		
	1	2	3	4	5	6	7	8	9
1. Eu sou capaz de usar ferramentas de desenvolvimento de software quando...									
a. tenho apenas a ajuda integrada da tecnologia como assistência.									
b. já vi alguém a utilizá-la antes de tentar eu próprio.									
c. não tenho ninguém por perto para ajudar se precisar de ajuda.									
d. alguém me ajudou a começar a utilizá-la.									
e. alguém me mostra como o fazer primeiro.									
f. já usei uma tecnologia semelhante antes, para fazer a mesma tarefa.									
g. nunca usei nada parecido antes.									
2. Eu não confio na minha capacidade para usar e aprender ferramentas de desenvolvimento de software. Eu tenho outros pontos fortes.									
3. Eu tiro tempo para explorar ferramentas de desenvolvimento de software que não é crítica para o meu trabalho.									
4. Uma razão pela qual gasto tempo e dinheiro em ferramentas de desenvolvimento de software é porque é uma maneira de fazer com que fique bem visto entre os meus pares.									
5. É divertido usar novas ferramentas de desenvolvimento de software que ainda não estão disponíveis ao público, como por exemplo participar em programas BETA para testar ferramentas de desenvolvimento de software inacabadas.									
6. Eu gosto de encontrar funcionalidades e capacidades menos conhecidas de ferramentas de desenvolvimento de software que eu uso.									

ANNEX GENDERMAG SURVEY IN ENGLISH

GenderMag Facets Survey

From the GenderMag Project (<http://gendermag.org>)

Where available, how to cite: ACM Digital Library. It's in the supplemental materials for this paper:

[Vorvoreanu et al. 2019] Mihaela Vorvoreanu, Lingyi Zhang, Yun-Han Huang, Claudia Hilderbrand, Zoe Steine-Hanson, Margaret Burnett, "From Gender Biases to Gender-Inclusive Design: An Empirical Investigation", 2019 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4-9, 2019, Glasgow, Scotland, UK. ACM, New York, NY, USA. Paper 53. 14 pages. <https://dl.acm.org/doi/10.1145/3290605.3300283>

You can also download it directly from <http://gendermag.org>

	Disagree Completely			Neither Agree Nor Disagree			Agree Completely		
	1	2	3	4	5	6	7	8	9
1. I am able to use <insert type of technology> when...									
a. I have just the built-in help for assistance.									
b. I have seen someone else using it before trying it myself.									
c. no one is around to help if I need it.									
d. someone else has helped me get started.									
e. someone shows me how to do it first.									
f. I have used similar technology before, to do the same task.									
g. I have never used anything like it before.									
2. I am not confident about my ability to use and learn <insert type of technology>. I have other strengths.									
3. I make time to explore <insert type of technology> that is not critical to my job.									
4. One reason I spend time and money on <insert type of technology> is because it's a way for me to look good with peers.									
5. It's fun to try new <insert type of technology> that is not yet available to everyone, such as being a participant in beta programs to test unfinished <insert type of technology>.									
6. I enjoy finding the lesser-known features and capabilities of the <insert type of technology> I									

ANNEX TEST GUIDE IN PORTUGUESE



Passos:

Passo 1 – Limpar a tela de trabalho: o objetivo é ficar com uma tela em branco para que possamos começar a trabalhar.

Passo 2 – Criar um role e editar o seu nome para "Meeting participant".

Passo 3 – Adicionar um objetivo (opção "Goal") dentro do role "Meeting participant".

Passo 4 – Editar o seu nome para "Meeting organised".

Passo 5 – Criar duas tarefas lado a lado, por baixo de "Meeting organised", uma com o nome "Collect available dates" e outra "Merge Available dates".

Passo 6 – Ligar estas duas tarefas ao objetivo "Meeting organised" através de relações "AND".

Passo 7 – Criar um recurso com o nome "Available dates" dentro do role "Meeting participant".

Passo 8 – Ligá-lo através de uma relação "needed-by" à tarefa "Collect available dates".

Passo 9 – Criar duas tarefas lado a lado, por baixo de "Merge available dates", uma com o nome "Manually merge dates" e outra "Merge dates using scheduler".

Passo 10 – Ligar estas duas tarefas à tarefa "Merge available dates" através de relações "OR".

Passo 11 – Criar uma qualidade com o nome "Minimal effort", por baixo destas duas últimas tarefas.

Passo 12 – Ligar a tarefa "Manually merge dates" à qualidade "Minimal effort", através de um link de contribuição "hurt".

Passo 13 – Ligar a tarefa "Merge dates using scheduler" à qualidade "Minimal effort", através de um link de contribuição "help".

Passo 14 – Ligar a tarefa "Merge Available dates" e a qualidade "Minimal effort" através de uma relação de qualificação.

Passo 15 – Adicionar a descrição "Manually merging two sets of dates" à tarefa "Manually merge dates".

Passo 16 – Adicionar a descrição "Use scheduler to merge dates." à tarefa "Merge dates using scheduler".

Passo 17 – Comparar e analisar a informação relativa a três itens: "Manually merge dates", "Merge dates using scheduler" e "Minimal effort".

Passo 18 – Apagar quatro itens: o recurso "Available dates", as tarefas "Manually merge dates" e "Merge dates using scheduler", e a qualidade "Minimal effort".

Passo 19 – Vamos descartar este modelo e criar um novo.

ANNEX TEST GUIDE IN ENGLISH



Steps:

- Step 1** – Clear the workspace: The goal is to have a blank screen so we can start working.
- Step 2** – Create a role and edit its name to "Meeting participant."
- Step 3** – Add a goal (option "Goal") within the role "Meeting participant."
- Step 4** – Edit its name to "Meeting organised."
- Step 5** – Create two tasks side by side, below "Meeting organised," one named "Collect available dates" and the other "Merge Available dates."
- Step 6** – Connect these two tasks to the goal "Meeting organised" through "AND" relationships.
- Step 7** – Create a resource named "Available dates" within the role "Meeting participant."
- Step 8** – Connect it to the task "Collect available dates" through a "needed-by" relationship.
- Step 9** – Create two tasks side by side, below "Merge available dates," one named "Manually merge dates" and the other "Merge dates using scheduler."

Step 10 – Connect these two tasks to the task "Merge available dates" through "OR" relationships.

Step 11 – Create a quality named "Minimal effort" below these last two tasks.

Step 12 – Connect the task "Manually merge dates" to the quality "Minimal effort" through a "hurt" contribution link.

Step 13 – Connect the task "Merge dates using scheduler" to the quality "Minimal effort" through a "help" contribution link.

Step 14 – Connect the task "Merge Available dates" and the quality "Minimal effort" through a qualification relationship.

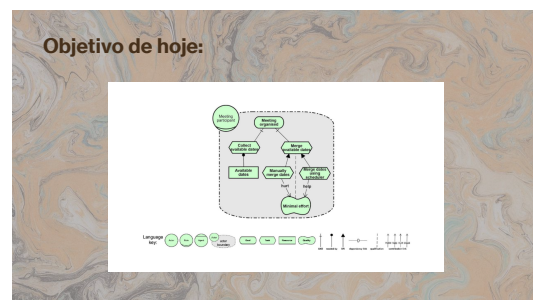
Step 15 – Add the description "Manually merging two sets of dates" to the task "Manually merge dates."

Step 16 – Add the description "Use scheduler to merge dates" to the task "Merge dates using scheduler."

Step 17 – Compare and analyze the information related to three items: "Manually merge dates," "Merge dates using scheduler," and "Minimal effort."

Step 18 – Delete four items: the resource "Available dates," the tasks "Manually merge dates" and "Merge dates using scheduler," and the quality "Minimal effort."

Step 19 – Let's discard this model and create a new one.







2024 Gender-Inclusive Interfaces: The Case of Software Development Tools

Débra Dias

